Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Aprendizaje Automático aplicado al sector hotelero

Machine Learning applied to Hotel Industry

Alien Embarec Riadi

La Laguna, *11* de septiembre de 2020

D. **Pedro Antonio Delgado**, con N.I.F. 45.725.874-B profesor Contratado Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

C E R T I F I C A

Que la presente memoria titulada:

“Aprendizaje Automático Aplicado al Sector Hotelero”

ha sido realizada bajo su dirección por D. **Alien Embarec Riadi**,

con N.I.F. 46.264.322-Y.

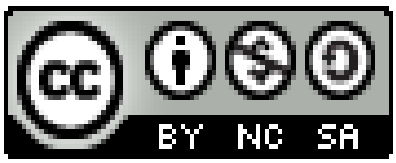
Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 11 de septiembre de 2020

Agradecimientos

En primer lugar, a mis padres y seres queridos por el continuo apoyo y motivación que me han transmitido. Sin ellos no hubiera sido posible llegar hasta donde he llegado.

A mi tutor de TFG, Don Pedro Antonio Toledo Delgado, por la buena orientación y planificación que ha dado al desarrollo de este trabajo.

Licencia



Resumen

El objetivo de este trabajo ha sido estudiar el fenómeno de la ocupación en la industria hotelera. Concretamente, disponemos de dos fuentes de datos [1], representativas de la actividad de reservas en dos hoteles ubicados en Portugal. Esta información viene complementada con una serie de variables adicionales que ayudan a entender cómo varía la demanda en los dos hoteles.

La información disponible, pues, cubre información de reservas, cancelaciones, edades de los clientes, tarifa diaria abonada por el alojamiento, categoría de habitaciones contratadas, etc.

El objetivo de este proyecto es, aprovechando las herramientas del Aprendizaje Automático, estudiar varios problemas comunes de la industria, como predecir el precio diario de alojamiento dadas unas variables que influyen como la antelación, el número de noches contratadas, la probabilidad de cancelación de una reserva, entre otras.

Otra de las metas de este proyecto es crear un modelo que prediga las reservas donde es más plausible la cancelación. El repositorio de datos cuenta con una variable *is_canceled* que indica si la reserva ha sido cancelada o no.

Esta variable, cotejada con otros datos de los que depende, como los recargos aplicados a algunas habitaciones, las peticiones especiales de los clientes, el efecto de las reservas hechas con mucha antelación, entre otros factores, puede contribuir a construir un modelo clasificador con un alto grado de fiabilidad, que dado un conjunto de datos de entrada, etiquete los registros más propensos a cambios y anulaciones.

Palabras clave: predicción precio alojamiento, cancelaciones reservas, regresión, clasificación, ciencia de datos, aprendizaje automático, aprendizaje supervisado, tarifa diaria de alojamiento.

Abstract

The aim of this work has been to study the phenomenon of occupation in the hotel industry. Specifically, we have two data sources [1], representative of the activity of reservations in two hotels located in Portugal. This information is complemented by a series of additional variables that help to understand how demand varies in the two hotels.

The available information therefore covers information on reservations, cancellations, customer ages, daily rate paid for accommodation, category of rooms contracted, etc.

The aim of this project is, making use of the tools of Machine Learning, study several common problems of the industry, such as predicting the daily price of accommodation given some variables that influence as the anticipation, the number of nights contracted, the probability of cancellation of a reservation, among others.

Another goal of this project is to create a model that predicts the reserves where cancellation is more plausible. The data repository has a variable is_canceled that indicates whether the reservation has been cancelled or not.

This variable, compared with other data it depends on, such as the surcharges applied to some rooms, special requests from customers, the effect of reservations made well in advance, among other factors, can help build a classification model with a high degree of reliability, which given a set of input data, label the records most likely to change and cancellations.

Keywords: price prediction accommodation, cancellations reservations, regression, classification, data science, machine learning, supervised learning, daily rate accommodation.

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1: Introducción

1.1 Motivación

El sector hotelero es (hasta el comienzo de la reciente crisis sanitaria del COVID-19 [2]) uno de los pilares de la economía de nuestro país [3], generando cifras astronómicas y constituyéndose en una fuente de riqueza imprescindible para el PIB nacional. Todo esto asociado a un crecimiento en la llegada de turistas extranjeros, continuado en los últimos años [4]. Este flujo ha de ser correspondido con una buena oferta turística, variable según el poder adquisitivo del turista.

Al enmarcarse España y Portugal como un destino turístico internacional [5], los establecimientos hoteleros han de contar con herramientas tecnológicas que les ayuden a entender su clientela, y optimizar sus recursos.

En ese sentido, el aprendizaje automático se erige como una de las mejores alternativas para solventar muchos problemas de gestión que hasta ahora habían tenido una solución manual [6].

Las empresas de la industria han de transformarse y conservar su ventaja competitiva ante el auge de plataformas de alquiler de apartamentos como *Airbnb* [7].

Como ejemplos de empleo de esta tecnología a citar, cabe destacar el programa *LightStay* [8]. Este programa emplea Inteligencia Artificial (de aquí en adelante referida como IA) para predecir el consumo de energía eléctrica, calefacción y agua en las habitaciones de los hoteles. Esta tecnología le ha ayudado a reducir costes y el impacto ambiental de sus actividades [9].

Además, el modelo de IA creado notifica cuando una habitación ha quedado desocupada para interrumpir el suministro de calefacción. Con esto la cadena hotelera logró ahorrar mucho dinero en recursos energéticos.

Por otro lado, la cadena *Wynn Las Vegas* ha anunciado recientemente que ha llegado a un acuerdo [10] con la tecnológica Amazon para implementar servicios del asistente *Amazon Echo* en las habitaciones de la cadena hotelera. Se busca automatizar tareas mediante órdenes de los clientes, como bajar la persiana, apagar el televisor, controlar la iluminación.

Es posible encontrar ejemplos en [11] y [12] donde la IA ha sido introducida para mejorar varios servicios del establecimiento.

En conclusión, la era presente demanda integrar esta tecnología en aras de mantener la competitividad y mejorar la atención a los clientes. Por ello, en este

proyecto se pretende aportar una nueva línea de investigación que ayude a entender cómo varía la demanda en los hoteles y el propio servicio interno.

1.2 Objetivos

El sector de la hostelería necesita reforzarse con herramientas digitales para entender mejor el fenómeno turístico del siglo XXI, como medida para contrarrestar el auge de las plataformas de alquiler de pisos turísticos, y la gran popularidad que estas están alcanzando sobre todo entre los turistas más jóvenes [13].

Como primer objetivo se propone estudiar la evolución de la tarifa de alojamiento diaria (*adr* [14]) en función del número de noches contratadas, la antelación con la que se hizo la reserva, el número de adultos presentes en ella, en adición de otras variables que se irán explorando.

La anterior es una cuestión muy abierta, dado que debido al volumen de datos disponible, es posible encontrar nuevas asociaciones entre el *adr* y otros factores. Este es un aspecto en el que se profundizará más adelante.

Como segundo objetivo se propone construir una herramienta que ayude a predecir las cancelaciones de reservas. Es posible encontrar una variable en los datos (*is_canceled*) que explica si una reserva ha sido cancelada o no.

Esta variable, estudiada en combinación con otras variables independientes, pero con una buena correlación con la variable objetivo, permitirá construir un árbol de decisión [15] que ayudará a mejorar la política de reservas, permitiendo saber qué tipo de reservas sufren más cancelaciones, y esto evidentemente, tiene su ventaja en que permite anticiparse y aplicar requisitos especiales o flexibilizar las condiciones de reserva.

1.3 Esquema del trabajo

El presente trabajo está dividido en 5 capítulos, la distribución y organización de los mismos es la siguiente:

- **Capítulo 1: Introducción**

  Es el capítulo inicial del documento, donde se explica la motivación y los objetivos marcados del proyecto por parte del autor.

- **Capítulo 2: Estado del arte**

  Se introducen los conceptos más importantes relacionados con el trabajo, estos tienen que ver con la propia ciencia de datos que ha dado lugar al

Aprendizaje Automático, y los campos científicos relacionados.

- **Capítulo 3: Análisis Exploratorio de Datos**

  Este capítulo trata los métodos utilizados para atenuar el efecto de los datos nulos y valores atípicos [16] en el conjunto de datos total. También se hace un trabajo exhaustivo de análisis de las variables, su significado y el impacto que podrían tener en los modelos de regresión y clasificación.

- **Capítulo 4: Aplicación de Regresores**

  En este apartado se describen los algoritmos de regresión para predecir variables como el precio diario de alojamiento (*adr*). Estos algoritmos se basan en distintas fórmulas para medir la distancia de la variable independiente de la dependiente.

  Principalmente se emplean medidas de correlación que explican qué tan bien las variables independientes determinan la variable objetivo. Se emplea regresión lineal, algoritmos basados en K-vecinos cercanos como el K-Neighbors Regressor [17], el Ridge [18], el LASSO [19], siendo estos últimos algoritmos que tratan de minimizar la suma de errores al cuadrado aplicando previamente coeficientes de ponderación de las variables independientes.

- **Capítulo 5: Conclusiones**

  Se exponen los resultados obtenidos, se contrasta la fiabilidad del modelo de regresión construido, planteado para predecir la tarifa de alojamiento diaria, frente a los datos reales disponibles en el conjunto de datos inicial.

  Por otro lado, se expone la información recabada sobre la cancelación de las reservas, mediante los algoritmos de clasificación empleados.

  Esta información es de especial interés, ya que por un lado, ayuda a los usuarios a minimizar el coste de su estancia en el hotel, sabiendo reservar el número de días adecuado, y por otro, contribuye a mejorar la planificación interna de los propios hoteles, conociendo mejor la demanda recibida.

CAPÍTULO 2: Antecedentes y Estado del arte

2.1 Hostelería e Impulso Digital

La importancia de los hoteles y su oferta de alojamiento radica en que son infraestructuras concebidas específicamente para atender a huéspedes que se encuentran de viaje por cualquier motivo [20].

Por lo tanto, están diseñados para soportar una estancia temporal de los clientes, y normalmente disponen una serie de dependencias como manutención, lavandería, limpieza, ocio, excursiones. Este hecho diferenciador del resto de categorías de alojamiento hace que sean infraestructuras con una gran demanda.

La actualidad del tema radica en que, más aún en las Islas Canarias, hay una necesidad de fidelizar a los turistas que nos visitan. Es objetivo de la economía del archipiélago. Hay un número importante de iniciativas para atraer un turismo de calidad [21], esto es, con un gasto promedio alto.

Este trabajo puede ayudar a organizar mejor las promociones, ya sea concediendo flexibilidad en las reservas de verano, paquetes turísticos más abundantes etc.

En *kaggle.com*, plataforma con repositorios de datos públicos, hay una fuente de datos publicada por dos investigadores portugueses [22], con cerca de 30 variables, y más de 120000 registros de reservas (anonimizadas), disponibles para su análisis por la comunidad.

Por otro lado, Airbnb, plataforma de pisos turísticos, también ha liberado varios conjuntos de datos [23].

La actualidad del tema radica en que, más aún en las Islas Canarias, hay una necesidad de fidelizar a los turistas que visitan las islas [24], en la medida que el turismo es el principal contribuidor a la economía del archipiélago [25].

Por otro lado, la clave de éxito de un hotel [26] reside en la buena gestión de los recursos con los que cuenta dicho hotel. Como es bien sabido, en esto influyen varios factores. Por un lado, es de especial consideración el factor humano, contar con una plantilla de calidad, en cuanto a calidad del servicio y tamaño de recursos humanos destinados a cada área. Que cada servicio del hotel esté provisto de un número de empleados suficiente y que estos tengan las cualidades de buena atención al público.

2.2 Tratamiento de Datos

Llevar a la práctica los conceptos teóricos de la ciencia de datos lleva aparejado mencionar varías técnicas o campos científicos. Dichas técnicas comprenden campos como la Inteligencia Artificial, el Aprendizaje Automático, o la propia matemática estadística. Estos conceptos convergen para dar lugar al tratamiento de datos masivos que se conoce hoy en día.

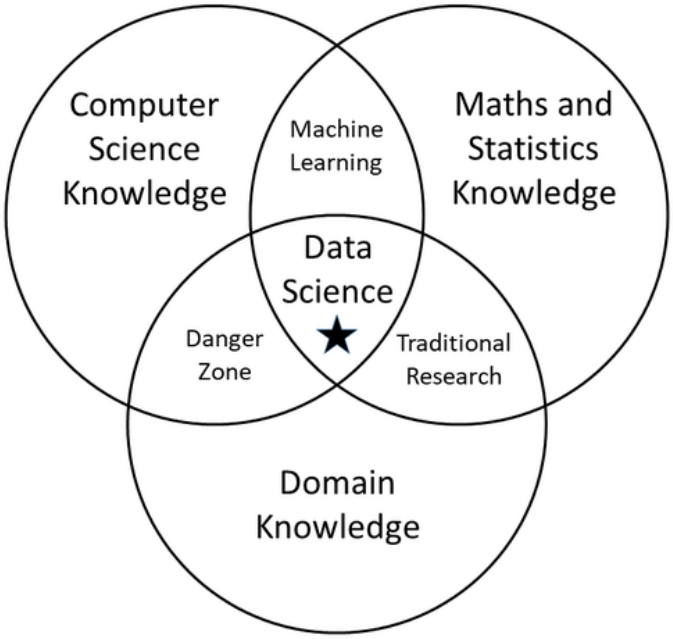


Figura 1. Disciplinas Tratamiento de Datos. Fuente: [27]

Por un lado, como se puede observar en la *Figura 1*, esta disciplina combina las altas capacidades de computación alcanzadas en los últimos años [28], junto con la implementación de diversos algoritmos estadísticos [29], que se encontraban en una fase de estudio muy avanzada a nivel académico.

El tercer factor es el Conocimiento de Dominio al que pertenecen los datos [30]. Este consiste en saber agregar características a los datos que puedan mejorar los resultados de los algoritmos de regresión y de clasificación que se usen. Dichas características no están basadas en datos nuevos, sino en los ya existentes.

Por ejemplo, si en el conjunto de datos se tiene el número de noches de fin de semana que hay en la reserva, una característica adicional que se podría crear sería el porcentaje de noches de fin de semana sobre el total de noches reservadas.

Estas características incrementan el número de variables de las que dispone un algoritmo para predecir, mejorando los resultados de correlación y reduciendo los porcentajes de divergencia entre los datos reales y los predichos por el algoritmo.

2.2.1 Inteligencia Artificial

La Inteligencia Artificial [31], es aquella inteligencia llevada a cabo por las máquinas. A diferencia de la inteligencia natural, característica de los humanos.

La inteligencia de las máquinas se desarrolla mediante distintas disciplinas que se

engloban dentro de las ciencias de la computación.

Algunas disciplinas que cabe mencionar son la Visión por Computador, mediante la capacidad de reconocer objetos. La optimización de procesos en diversas áreas, implementable mediante la automatización de tareas realizadas por humanos. El procesamiento del lenguaje natural, en forma de asistentes de voz conversacionales.

Es posible también citar como ejemplos de IA las redes neuronales, el reconocimiento de patrones, los sistemas autónomos como los aviones no tripulados o los coches autónomos.

Esta ciencia se abre también a aplicaciones médicas [32], con varias líneas de trabajo abiertas, destacando los robots quirúrgicos, análisis de imágenes clínicas etc.

El aprendizaje automático que se pretende estudiar en este trabajo es un subcampo de la inteligencia artificial. Su principal tarea es encontrar patrones en los datos. Se divide a su vez en varias ramas, aprendizaje supervisado, no supervisado, y el aprendizaje por refuerzo.

2.2.2 Ciencia de Datos

La ciencia de datos [33] es un campo multidisciplinar que tiene asociados varios métodos científicos cuyo objetivo es la extracción de conocimiento de los datos.

El problema de los datos es que tienden a crecer masivamente [34], hacer un análisis visual de ellos es una tarea muy compleja, que supera incluso las capacidades de los programas de procesamiento de datos tradicionales.

Por ello, surgen varias técnicas y campos que se dedican a desarrollar procedimientos para automatizar el análisis de los datos, y devolver conocimiento útil en forma de representaciones gráficas y métricas numéricas que ayuden a entender en qué sentido y dimensiones se mueven los datos estudiados.

Las técnicas que constituyen la ciencia de datos suponen una evolución de la estadística convencional. La ciencia de datos se apoya en el aprendizaje automático, la minería de datos, y las funcionalidades de lenguajes de programación para manipular datos.

El tratamiento de datos, más concretamente el tratamiento de datos masivos, suele estructurarse entorno a la perspectiva de las 5 Vs [35]: Volumen, Velocidad, Variedad, Veracidad y Valor

En cuanto al **Volumen**, esta característica del Big Data se refiere al tamaño de las cantidades de datos que se generan actualmente. Antiguamente los datos se generaban de forma manual. Actualmente provienen de máquinas o dispositivos y

se gestan de manera automática, por lo que el volumen a analizar es masivo**.**

El segundo componente es la **Velocidad**, se refiere a la rapidez con la que se generan los datos, en la era actual es muy habitual que se generen grandes volúmenes de datos que hacen que queden desfasados los inmediatamente anteriores.

La **Variedad** indica la heterogeneidad de los datos. Tratándose de un mismo dominio de aplicación, pueden provenir de distintas fuentes. Un ejemplo es el sistema de gestión de reservas de un hotel, suele estar conectado a muchos canales de venta ya que ello garantiza una buena ocupación de la infraestructura.

Estos canales suelen ser la venta en línea, la venta a través de buscadores de hoteles, que pueden ser muchos. También suelen haber canales de venta física, de manera directa o a través de agencias de viajes.

Esta presencia en muchos nichos de mercado no garantiza que los datos estén estructurados y con el mismo formato, por lo que el tratamiento de datos ha de considerar este aspecto a la hora de abordar los problemas aprendizaje automático.

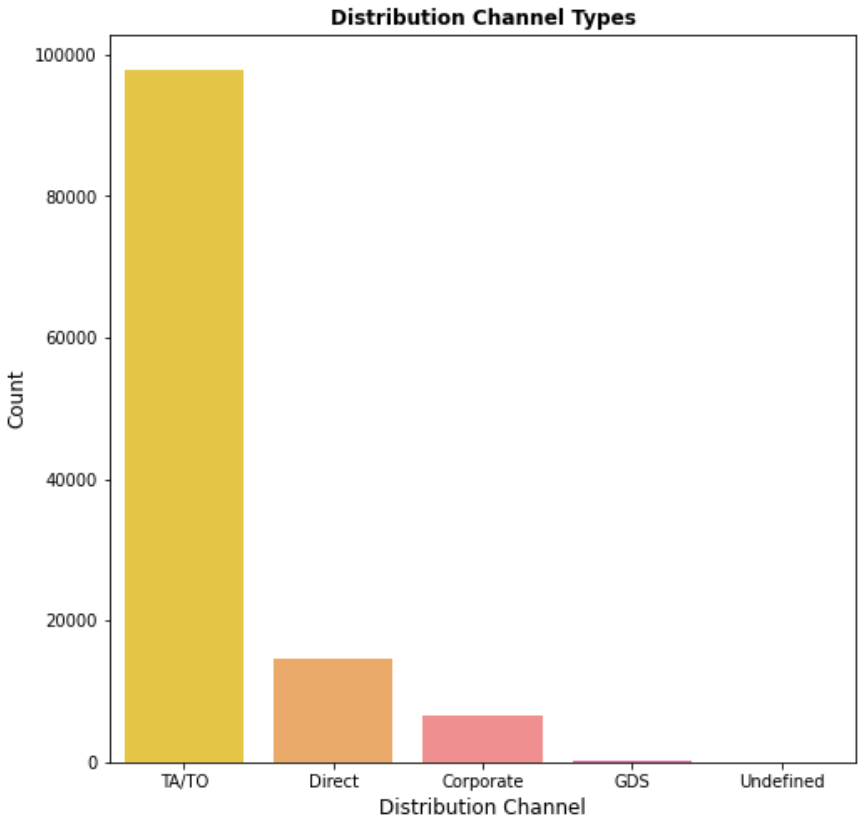


Figura 2. Sistema de Reservas de un Hotel. Fuente: contenido propio.

En la *Figura 1 TA* hace referencia a Agente de Viaje y *TO* a Tour-operador. *GDS* [36] es un sistema de gestión de reservas global que conecta a muchos establecimientos hoteleros.

La **Veracidad** aborda la integridad de los datos. Al provenir estos de varias fuentes

y estar en distintos formatos, pueden presentar información incompleta, nula, o muy distante de su media.

Éste es un problema frecuente en los problemas de Aprendizaje Automático, cómo conseguir el conjunto de variables y de datos que mejor estimen la característica que se quiere predecir. Normalmente una de las primeras tareas en todo proceso de tratamiento de datos suele ser analizar cada una de las variables, ver su media, mediana, varianza, su distribución, número de celdas nulas. Estos indicadores ayudan mucho a detectar información irrelevante o poco frecuente en los datos.

Por otro lado, dado un problema particular, es habitual emplear técnicas de normalización y selección de características. Estos tratamientos van encaminados a obtener el conjunto de variables independientes que mejor predicen la variable objetivo.

En base a esto, el proceso de aprendizaje de los algoritmos mejora, se reduce el tiempo de cómputo, y los resultados presentan menor tasa de error.

Por último, es importante la cualidad del **Valor**. La capacidad de explotar los resultados del aprendizaje automatizado, ya que están destinados a optimizar una tarea particular, y es importante la adaptación del negocio a esta herramienta.

2.2.3 Aprendizaje Automático

El aprendizaje automático [37] o aprendizaje de máquinas (del inglés, *machine learning*) es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las máquinas aprendan.

El objetivo de las investigaciones en este campo es proporcionar a las computadoras la capacidad de aprender, sin ser programadas explícitamente. Desde otro punto de vista [38], el aprendizaje automático se centra en el desarrollo de programas informáticos que pueden cambiar cuando se exponen a datos nuevos.

El aprendizaje automático tiene una amplia gama de aplicaciones, es posible localizar ejemplos en los diagnósticos médicos, en la detección de fraudes en el uso de tarjetas de crédito, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, juegos y robótica.

Dependiendo del tipo de salida que se produzca y de cómo se aborde el tratamiento de los ejemplos, los diferentes algoritmos de Aprendizaje Automático se pueden agrupar en [39]:

- **Aprendizaje supervisado**: Los algoritmos trabajan con datos etiquetados o agrupados en variables conocidas, intentando encontrar que, dadas las variables de entrada, se les asigne la etiqueta de salida adecuada.

- **Aprendizaje no supervisado:** Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan solo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos.

- **Aprendizaje semisupervisado:** Este tipo de algoritmos combinan los dos algoritmos anteriores para poder clasificar de manera adecuada. Se tiene en cuenta los datos marcados y los no marcados.

- **Aprendizaje por refuerzo:** Su información de entrada es el *feedback* o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el sistema aprende a base de ensayo-error.

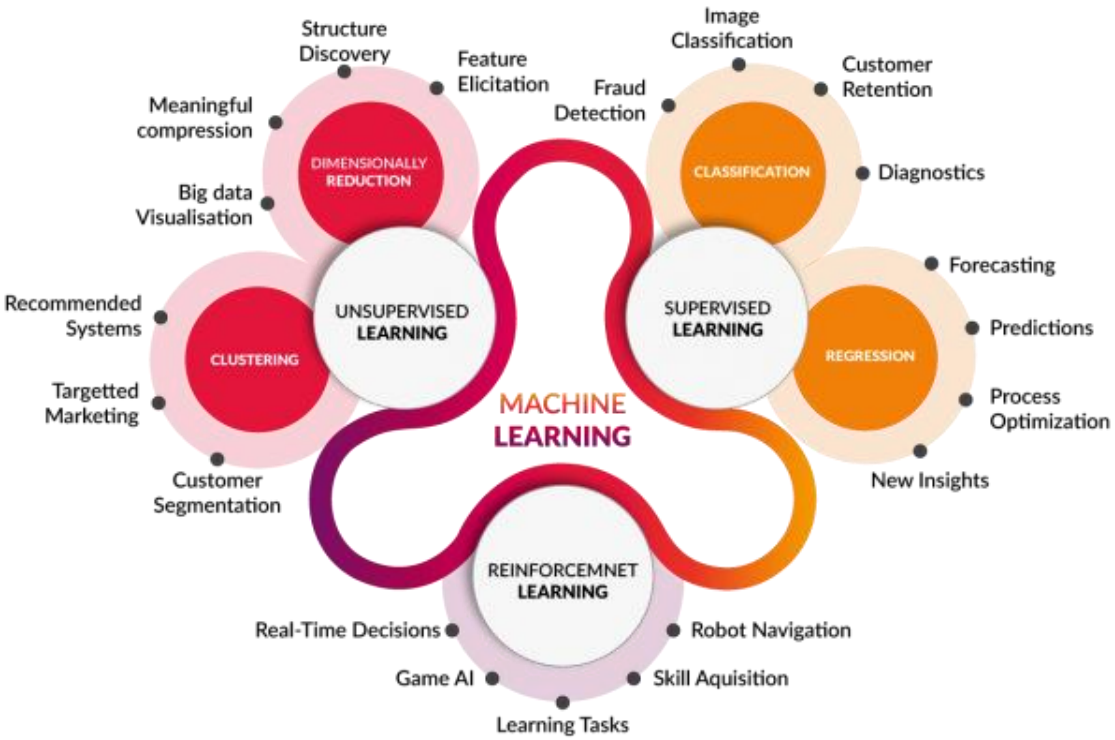


Figura 3. Disciplinas Aprendizaje Automático. Fuente: [40]

2.2.3.1 Aprendizaje Supervisado

El aprendizaje supervisado es cuando se tienen variables de entrada (*x*) y una variable de salida (*Y*) y se utiliza un algoritmo para aprender la función de mapeo de la entrada a la salida.

$$Y = f(X)$$

Figura 4. Fórmula función lineal. Fuente: [41]

Siendo *X* la variable de entrada e *Y* la variable de salida.

El objetivo es aproximar la función de mapeo tan bien que cuando se tengan nuevos datos de entrada (x) se puedan predecir las variables de salida (*Y*) para esos datos.

Se denomina aprendizaje supervisado porque el proceso de aprendizaje de un algoritmo a partir del conjunto de datos de entrenamiento puede considerarse como un profesor que supervisa el proceso de aprendizaje. Conocemos las respuestas correctas, el algoritmo hace predicciones de forma iterativa sobre los datos de entrenamiento y es corregido por el profesor. El aprendizaje se detiene cuando el algoritmo alcanza un nivel de rendimiento aceptable.

Los problemas de aprendizaje supervisado se pueden agrupar en problemas de regresión y clasificación.

- **Clasificación**: Un problema de clasificación es cuando la variable de salida es una categoría, como "rojo" o "azul" o "enfermedad" y "sin enfermedad".

- **Regresión**: Un problema de regresión es cuando la variable de salida es un valor real, como la temperatura o el peso.

Los algoritmos más habituales que aplican para el aprendizaje supervisado son:

- Árboles de decisión
- Clasificación de Naïve Bayes
- Regresión por mínimos cuadrados
- Regresión Logística
- Support Vector Machines (SVM)
- Métodos *Ensemble* (Conjuntos de clasificadores)

2.2.3.2 Aprendizaje no Supervisado

El aprendizaje no supervisado tiene lugar cuando no se dispone de datos etiquetados para el entrenamiento. Sólo se tienen datos de entrada (*X*) y no hay variables de salida correspondientes. Por ello, tienen un carácter exploratorio.

Los problemas de aprendizaje no supervisado pueden agruparse en problemas de agrupación y asociación.

- **Agrupación**: un problema de agrupación es cuando se quiere descubrir las agrupaciones inherentes en los datos, como la agrupación de los clientes por el comportamiento de compra.

- **Asociación**: Un problema de aprendizaje de reglas de asociación es cuando se desea descubrir reglas que describan grandes porciones de los datos, como que las personas que compran X también tienden a comprar Y.

Algunos ejemplos populares de algoritmos de aprendizaje no supervisado son:

K-Means para problemas de agrupación.
Algoritmo *Apriori* para problemas de aprendizaje de reglas de asociación.

CAPÍTULO 3: Análisis Exploratorio de Datos

3.1 Conjunto de Datos

El conjunto de datos ha sido publicado por tres autores portugueses [1], Antonio Nuno, Ana de Almeida y Luis Nunes. Los investigadores se encuentran adscritos a la Universidad de Lisboa [42] [43] [44].

Los datos han sido obtenidos gracias a la colaboración de dos hoteles, el más representativo en los datos está ubicado en la capital portuguesa, y tiene un perfil de hotel de ciudad, con pernoctaciones breves.

El segundo hotel se encuentra en la región del *Algarve*. Se caracteriza por ser un complejo turístico, con una alta ocupación en temporadas de verano.

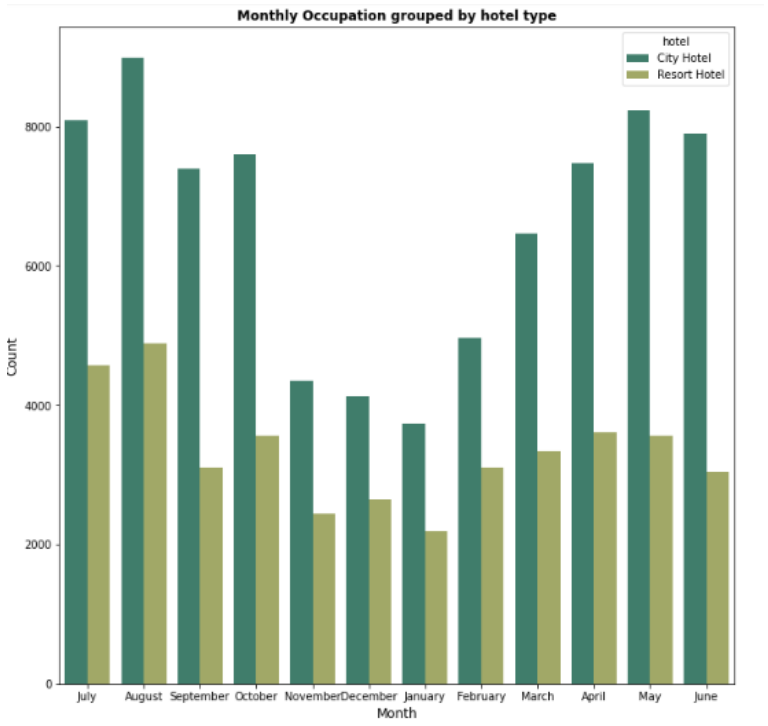


Figura 5.Ocupación Mensual filtrada por tipo de hotel. Fuente: contenido propio

El conjunto de datos inicial dispone de treinta y dos variables. De las cuales 12 columnas son categóricas, y 20 de tipo numérico.

La tabla ocupa 119390 filas. Cada una de ellas corresponde a una reserva efectuada por un cliente.

Los datos inicialmente se encontraban dispersos en diferentes tablas de sistema

gestor de reservas de cada hotel. No obstante, gracias al trabajo de los autores citados, fueron unificados en una única tabla con variables ya existentes y otras calculadas.

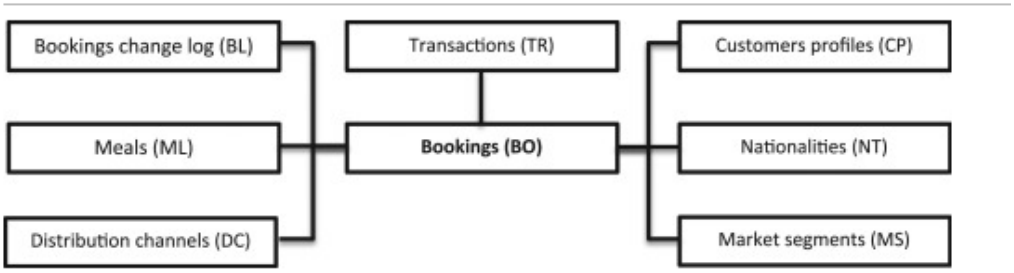


Figura 6. Sistema Gestor de reservas del que se extrajeron los datos. Fuente: [45]

Las variables presentes miden diferentes aspectos de las reservas, como el tipo de hotel, si fue cancelada o no, número de adultos presentes en ella, el número de niños, la tarifa diaria. También es posible localizar información acerca de los recargos aplicados sobre algunas categorías de habitación, peticiones especiales de los clientes, en adición de otras variables que se irán explorando.

```
[13] print(data.shape)

     (119390, 32)
```

Figura 7. Tamaño de los datos. Filas y Columnas. Fuente: contenido propio.

A continuación se presenta una tabla donde se describe el significado de cada una de las columnas.

Nombre Columna	Significado	Valores Posibles
hotel	Hotel (H1 = Resort Hotel o H2 = City Hotel)	City Hotel/Resort Hotel
is_canceled	Valor indicando si la reserva fue cancelada o no	(no cancelado) 0/ (cancelado) 1
lead_time	Número de días que han transcurrido desde la fecha de entrada de la reserva en el *PMS*	Entero que indica el número de días
arrival_date_year	Año de la fecha de llegada del cliente	(2015 / 2016 / 2017)
arrival_date_ month	Mes de la fecha de llegada	Enero - Diciembre
*arrival_date_*	Semana del año en la que se produce la llegada	0-53

week_number		
arrival_date_day_of_month	Día del mes en el que se produce la llegada	1-31
number_of_weekend_nights	Número de noches de fin de semana que el huésped se alojó o reservó para alojarse en el hotel	Entero
stays_in_week_nights	Número de noches de la semana (de lunes a viernes) que el huésped se alojó o reservó para alojarse en el hotel	Entero
adults	Número de adultos registrados en la reserva	Entero
children	Número de niños (entre 2 y 12 años) registrados en la reserva	Entero, NA
babies	Número de bebés registrados en la reserva (< 2 años de edad)	0-1-2-9-10
meal	Tipo de comida reservada. Las categorías se presentan en los paquetes de comidas de hospitalidad estándar: • Indefinido / SC – sin plan de manutención • Bed & Breakfast (BB) – El desayuno está incluido en la tarifa. • Half Board (HB) (desayuno y cena normalmente) • FB – Full Board (desayuno, almuerzo y cena)	BB, FB, HB, SC, Undefined
country	País de origen de la persona que efectúa la reserva. Las categorías son representadas según el estándar ISO 3155–3:2013	Código de tres letras mayúsculas
market_segment	Designación del segmento de mercado. En las categorías, el término "TA" significa "Agentes de Viaje" y "TO" en inglés significa "Operadores Turísticos" (TA/TO/Direct)	Aviation/Complementary/Corporate/Direct/Groups/Offline TA/TO/Online TA/Undefined
distribution_channel	Canal de distribución de reservas. El término "TA" significa "Agentes de Viaje" y "TO" significa "Operadores Turísticos" (TA/TO/Direct)	Corporate/Direct/GDS/TA o TO/Undefined
is_repeated_guest	Valor que indica si el nombre de la reserva era de un huésped repetido (1) o no (0) (TA/TO/Direct)	1(repetido) / 0(no repetido)
previous_cancellations	Número de reservas previas que fueron canceladas por el cliente antes de la reserva actual	Valores numéricosentre 0 y 26
previous_bookings_not_canceled	Número de reservas anteriores no canceladas por el cliente antes de la reserva actual	Valores numéricos entre 0 y 7
reserved_room_type	Código de tipo de habitación reservada. El código se presenta en lugar de la designación por razones de anonimato.	A-B-C-D-E-F-G-H-L-p
assigned_room_type	Código del tipo de habitación asignada a la reserva. A veces la clase de habitación asignada difiere de la reservada.	A-B-C-D-E-F-G-H-I-K-L-p
booking_changes	Número de cambios/enmiendas hechos a la reserva desde el momento en que la reserva fue registrada en el PMS	Entre 0 y 21
deposit_type	Indicación sobre si el cliente hizo un depósito para garantizar la reserva. Esta variable puede asumir tres categorías:	No-Deposit / Non-Refund / Refundable

	Sin depósito - no se hizo ningún depósito; No Reembolso - se hizo un depósito en el valor del costo total de la estancia; Reembolsable - se hizo un depósito con un valor inferior al coste total de la estancia.	
agent	La identificación de la agencia de viajes que hizo la reserva	Valor numérico.
company	Identificación de la compañía/entidad que hizo la reserva o responsable del pago de la misma. La identificación se presenta en lugar de la designación por razones de anonimato	Valor numérico.
days_in_waiting_list	Número de días en la lista de espera	Valor numérico
customer_type	Número de días que la reserva estuvo en la lista de espera antes de ser confirmada al cliente	Contract / Group / Transient / Transient-Party
adr	La tasa media diaria se define dividiendo la suma de todas las transacciones de alojamiento por el número total de noches de estancia	Valor numérico
required_car_parking_spaces	Número de plazas de aparcamiento requeridas por el cliente	Valor numérico
total_of_special_requests	Número de peticiones especiales hechas por el cliente (por ejemplo, cama doble o piso alto)	Valor numérico
reservation_status	El último estado de la reserva, asumiendo una de las tres categorías: **Cancelado** - la reserva fue cancelada por el cliente; **Check-Out** - el cliente se ha registrado pero ya ha salido; **No-Show** - el cliente no se registró ni informó al hotel de la razón por la cual no lo hizo	Canceled / CheckOut / No-Show
reservation_status_date	Fecha en la que se fijó el último estado.	2014-2015-2016-2017

Tabla 1. Descripción Conjunto de Datos. Fuente: contenido propio.

3.2 Herramienta de análisis

La herramienta utilizada en este trabajo fue *Google Collab* [46], que se apoya en el lenguaje Python. Se ha elegido por su gran versatilidad y flexibilidad de uso en el tratamiento de datos. Dispone de importantes librerías para procesamiento de datos como como *Numpy* [47], *SciPy* [48], *Pandas* [49].

También dispone de librerías para realizar aprendizaje supervisado y no supervisado como *Scikit-learn* [50].

En el apartado de visualización de datos también ofrece bastantes opciones con *Matplotlib* [51] y *Seaborn* [52].

Por último, el factor diferenciador del resto de lenguajes y plataformas es la facilidad de uso de *Pyhton* [53] para tratamiento de datos. Viene con una serie de

documentación muy completa y con funciones específicas para numerosas operaciones habituales en procesamiento de datos.

3.3 Valores ausentes

El origen de los datos es muy diverso, pues como se ha visto en los canales de distribución, el hotel está presente en muchas plataformas de venta, esto ocasiona que muchas variables contengan valores nulos por las diferencias de formato y de información que captura cada plataforma. Esto, en muchos sistemas gestores de bases de datos, al fusionar tablas de datos se sustituyen los datos faltantes o no declarados por valores nulos.

El primer paso para detectar estos valores es hacer una visualización de cada variable de manera independiente. Una vez hecho esto obtenemos las variables que contienen datos nulos y se decide el tratamiento a seguir con cada variable para paliar la ausencia de estos valores.

En la figura siguiente se puede observar las variables con valores nulos. La salida ha sido generada con el *data.info()*.

Se deduce que hay variables como *children*, *country*, *company* y *agent* que tienen un porcentaje de valores nulos. Más adelante, la utilización de diagramas de barras y de sectores nos ayudará a encontrar otras variables que presentan valores *undefined* o *NaN*.

Si bien los datos nulos se suelen sustituir por la media, mediana o moda de la variable, se adoptará una solución personalizada a cada una de las variables nulas presentes en este trabajo, a fin de obtener un conjunto de datos cohesionado y encaminado a facilitar la posterior tarea de regresión.

```
agent                           16340
company                        112593
```

Figura 8. Valores nulos en cada columna. Fuente: contenido propio.

Figura 10. Salida generada con el comando *df.isnull().sum()*.

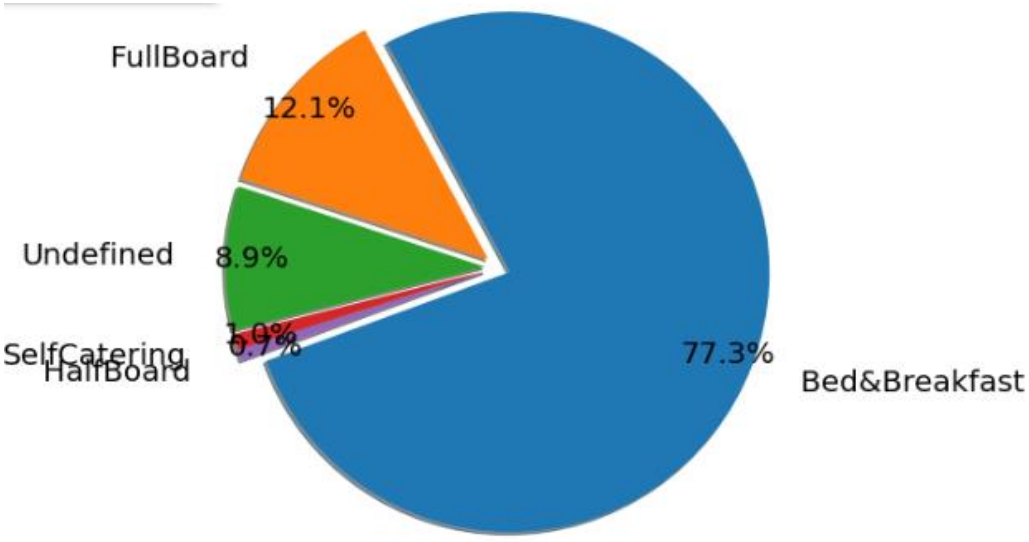


Figura 9. Diagrama de sectores variable meal. Fuente: contenido propio.

En la figura anterior, BB corresponde a *Bed and Breakfast* (desayuno incluido), FB es *Full Board* (pensión completa), HB o *Half Board* (desayuno y cena). SC es *Self-Catering*, y equivale a reserva sin manutención.

En la variable *meal* hay un 0,96 % de valores *undefined*. El tratamiento seguido con los valores ausentes en esta columna ha sido sustituir estas celdas por el valor *SC,* abreviatura de *self-catering*.

Esto se asume porque los valores nulos en la opción de manutención son asociables a que no se ha elegido ninguna opción de alimentación en las instalaciones del hotel, y los valores de self-catering van en ese sentido. Por consiguiente, se estima razonable la conversión a esta última categoría.

```
import pandas as pd


pd.set_option('display.max_rows', 500) #Maximizar información que se imprime de la tabla, a efectos de depuración
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

with open('/content/drive/My Drive/TFG_AlienEmbarecRiadi/hotel_bookings.csv') as csv_file:
    df = pd.read_csv(csv_file, delimiter=',')
    null_meal_1 = "Undefined"

    for i in range(df["meal"].size):
      try:
        if str(df["meal"][i]) == null_meal_1:
          df["meal"][i] = "SC"
      except:
        print("something went wrong, please check that data is of type string")

    df.to_csv("/content/drive/My Drive/TFG_AlienEmbarecRiadi/hotel_bookings_modified.csv", index=False)

    print(df.tail(50))
```

Figura 10. Conversión meal undefined a Self-Catering. Fuente: contenido propio

La segunda variable con valores nulos es *children*, presenta una baja cantidad de filas afectadas por este valor, inferior a 5. En este caso se asumen que son reservas

donde no hay menores de 12 años. Por lo que se transforman a 0 con la línea *data['children'] = data['children'].fillna(0)*.

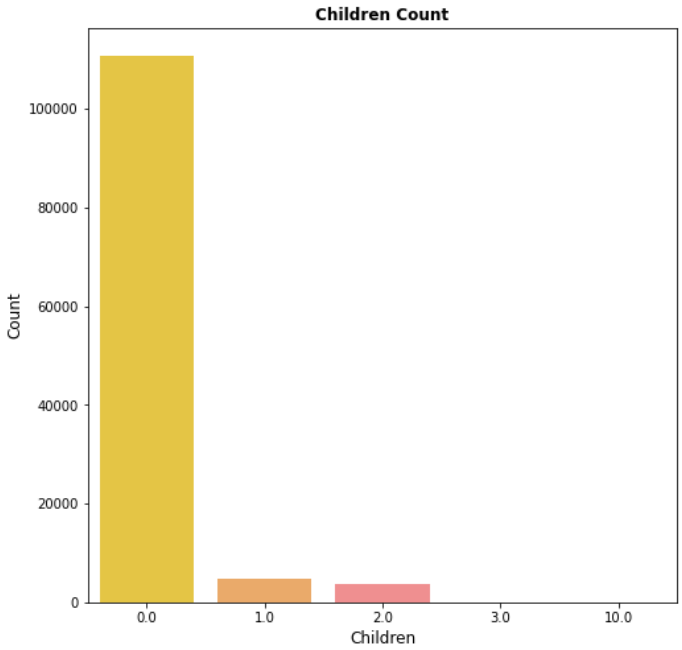


Figura 11. Diagrama de barras variable children. Fuente: contenido propio.

company es una variable que contiene un alto porcentaje de nulos. La decisión tomada para esta variable es su eliminación del conjunto de datos. Se trata de una variable similar a *agent*, por cuanto representa el medio de pago del cliente, y por tanto su información se encuentra cubierta y redundada por otras características.

Por otra parte, cabe citar que de no eliminarse podría afectar los modelos de regresión, ya que estos empeoran sus resultados con los valores nulos, al multiplicarse estos en ecuaciones de distancia cuyo resultado nulo reducirá el grado de verosimilitud de las variables independientes en relación a la característica deseada.

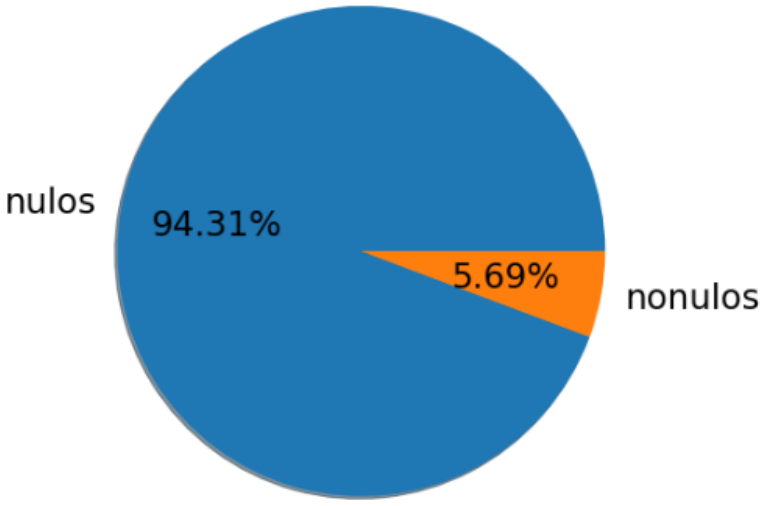


Figura 12. Representación de valores nulos en la variable company. Fuente: contenido propio.

Las variables *market_segment* y *distribution_channel* presentan bajo porcentaje de valores *undefined*. Se ha optado por reemplazar estos valores por la categoría *Unknown*.

```
df['market_segment'] = np.where(df['market_segment'] == 'Undefined', 'Unknown', df['market_segment'])
df['distribution_channel'] = np.where(df['distribution_channel'] == 'Undefined','Unknown', df['distribution_channel'])
```

Figura 13. Transformación numérica market_segment y distribution_channel. Fuente: contenido propio.

Para la columna *agent,* que representa el código de la agencia asociada con el hotel, se ha seguido el mismo procedimiento que con *children*, se asume que los nulos se deben a contrataciones directas de los clientes, sin intermediarios, por lo que se han reemplazado los valores nulos por el valor 0 con *data['agent'].fillna(0, inplace=True)*.

Para la columna *adr*, que indica la tarifa abonada por día de alojamiento, esta presenta cerca de 2000 valores de 0, y un valor negativo. Si bien esto se podría deber a promociones especiales, viajes de empresa u otras razones. Para el propósito de este trabajo conviene prescindir de las filas que contienen estos valores, dado que representan valores atípicos que pueden distorsionar los resultados de regresión.

Para el problema de regresión, también se prescinde de los valores de *adr* superiores de 300 euros. Dado que la mayoría de las reservas oscilan entre 50 y 300 euros por noche de alojamiento.

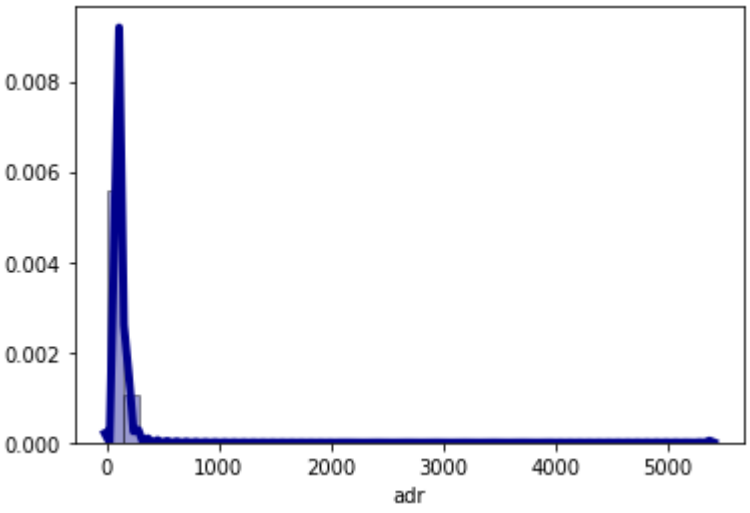


Figura 14. Histograma de la tarifa diaria de alojamiento. Fuente: contenido propio.

Como se puede observar, en el eje Y se encuentra la densidad / frecuencia relativa acumulada de los datos. Se alcanza el 80% de estos en valores de adr inferiores a 300.

Por consiguiente, se eliminan las filas que tienen un *adr* inferior a 0 o superior a 300 euros.

```
# eliminando adr > 300
indexNames = data[ data['adr'] > 300 ].index


print('indexNames ', indexNames)
data.drop(indexNames , inplace=True)
data.dropna(axis=0, inplace=True)

# eliminando adr <= 0
indexNames = data[ data['adr'] <= 0 ].index

print('indexNames ', indexNames)
data.drop(indexNames , inplace=True)
data.dropna(axis=0, inplace=True)
```

Figura 15. Eliminar valores atípicos tarifa diaria de alojamiento. Fuente: contenido propio.

3.4 Agregación de características

El proceso de agregación de características está presente en todas las fases del aprendizaje automático, ya que depende en buena medida del tipo de problema a estudiar. Es muy distinto el conjunto de características que se proporciona a un algoritmo de regresión que a uno de clasificación.

La primera variable agregada fue el número total de noches de alojamiento para cada reserva. Esto iba a dar una visión global de cómo varían las reservas a lo largo del año.

```
with open('/content/drive/My Drive/TFG_AlienEmbarecRiadi/hotel_bookings.csv') as csv_file:
    df = pd.read_csv(csv_file, delimiter=',')
    try:
      df["total_stayed_nights"] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
    except:
      print("Something went wrong, check that operands are of number type")


    for i in range(df["total_stayed_nights"].size):
      try:
        if int(df["total_stayed_nights"][i]) == 0:
          df["weekend_nights_proportion"][i] = 0
        else:
          df["weekend_nights_proportion"][i] = ("%.2f" %((df['stays_in_weekend_nights'][i]/df['total_stayed_nights'][i]) * 100))
      except ValueError:
          print("Something went wrong, check that operands are of number type")

    df.to_csv("/content/drive/My Drive/TFG_AlienEmbarecRiadi/hotel_bookings_modified.csv", index=False)
```

Figura 16. Creación Variable total_stayed_nights y weekend_nights_proportion. Fuente: contenido propio.

La segunda característica creada en la figura anterior es *weekend_nights_proportion*. Es sabido que las noches de fin de semana en la hostelería son más caras que los días entre semana. Más adelante se quiere estudiar la correlación de esta variable con el *adr*.

3.4.1 Conversión numérica de variables categóricas

Para las variables categóricas fue preciso utilizar la técnica conocida como el *Dummy Encoding* [54]. Esta consiste en dividir una columna categórica (que almacena cadenas, objetos, valores no numéricos) en varias columnas numéricas que representan cada una de las categorías de la variable transformada.

Citando un ejemplo, si en un hotel se dispone de habitaciones de tipo A, B, C y D, la transformación numérica resultará en 4 columnas donde para una determinada fila, si se ha seleccionado la habitación A, la columna toma valor uno y el resto de habitaciones que no son de tipo A, adoptan el valor 0.

```
[9]
one_hot = pd.get_dummies(data['reserved_room_type'], prefix='room')
```

```
print(one_hot)
```

```
        room_0  room_1  room_2  room_3  room_4  room_5  room_6  room_7  room_8
2            1       0       0       0       0       0       0       0       0
3            1       0       0       0       0       0       0       0       0
4            1       0       0       0       0       0       0       0       0
5            1       0       0       0       0       0       0       0       0
6            0       0       1       0       0       0       0       0       0
...        ...     ...     ...     ...     ...     ...     ...     ...     ...
119385       1       0       0       0       0       0       0       0       0
119386       0       0       0       0       1       0       0       0       0
119387       0       0       0       1       0       0       0       0       0
119388       1       0       0       0       0       0       0       0       0
119389       1       0       0       0       0       0       0       0       0

[117154 rows x 9 columns]
```

Figura 17. Ejemplo Dummy Encoding. Fuente: contenido propio.

Se ha seguido este procedimiento con todas aquellas variables categóricas que podían tomar más de dos valores posibles.

```
one_hot = pd.get_dummies(data['arrival_date_month'], prefix='month')

data = data.drop('arrival_date_month', 1)

data = data.join(one_hot)

print(data)
```

Figura 18. Ejemplo Dummy Encoding (1). Fuente: contenido propio.

```
one_hot = pd.get_dummies(data['meal'], prefix='meal')

data = data.drop('meal', 1)

data = data.join(one_hot)

print(data)
```

Figura 19. Ejemplo Dummy Encoding (2). Fuente: contenido propio.

3.5 Visualización de datos

A continuación se muestran gráficas de las distribuciones que siguen algunas variables, algo que da cuenta de dónde se sitúan los valores más frecuentes. Ha sido de especial importancia para detectar valores atípicos.

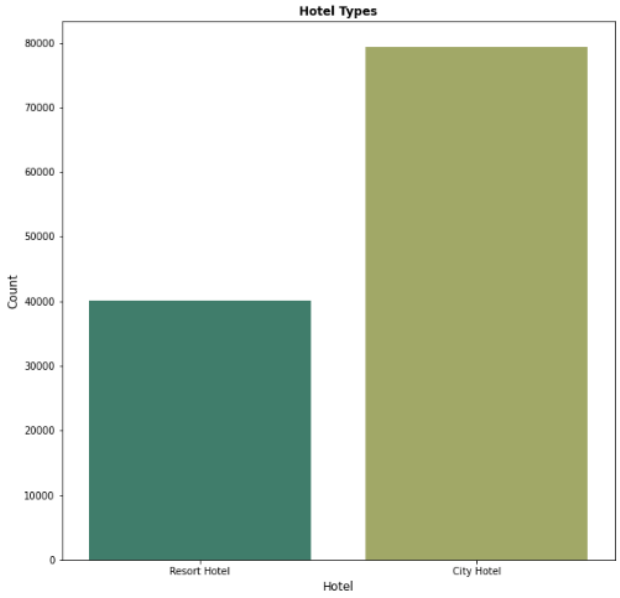


Figura 20. Diagrama de barras variable hotel. Fuente: contenido propio.

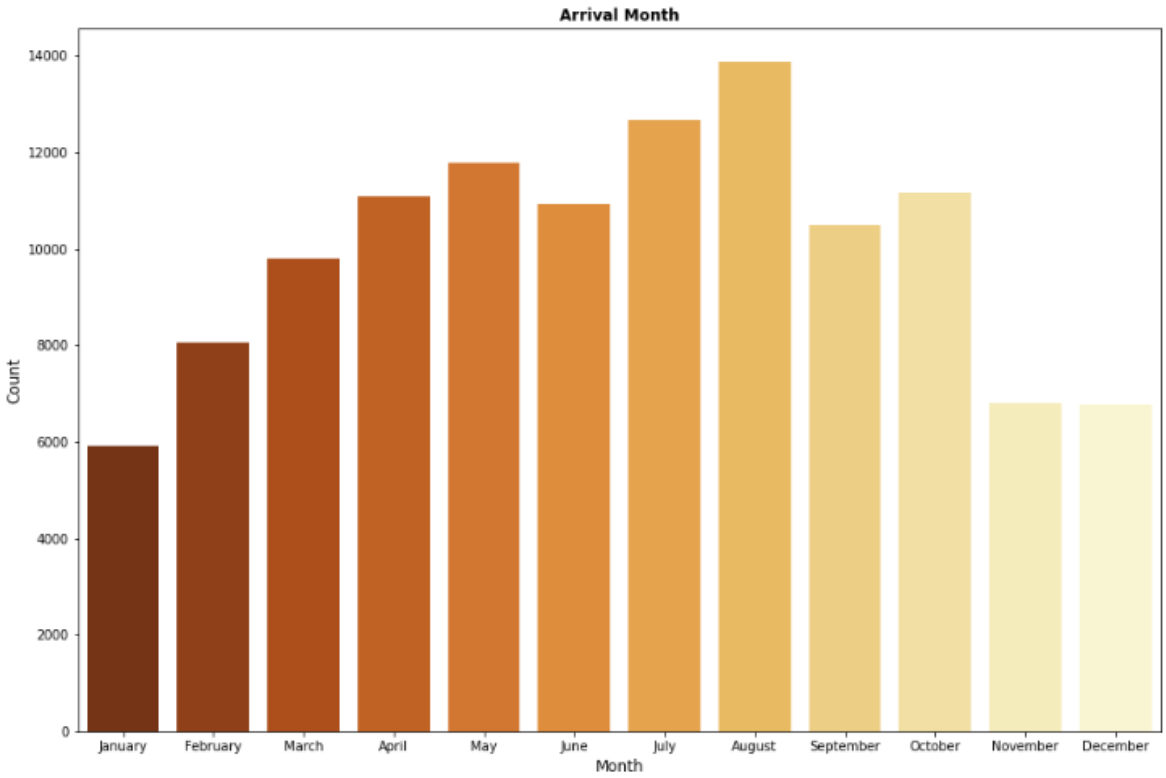


Figura 21. Ocupación Mensual de los hoteles. Fuente: contenido propio.

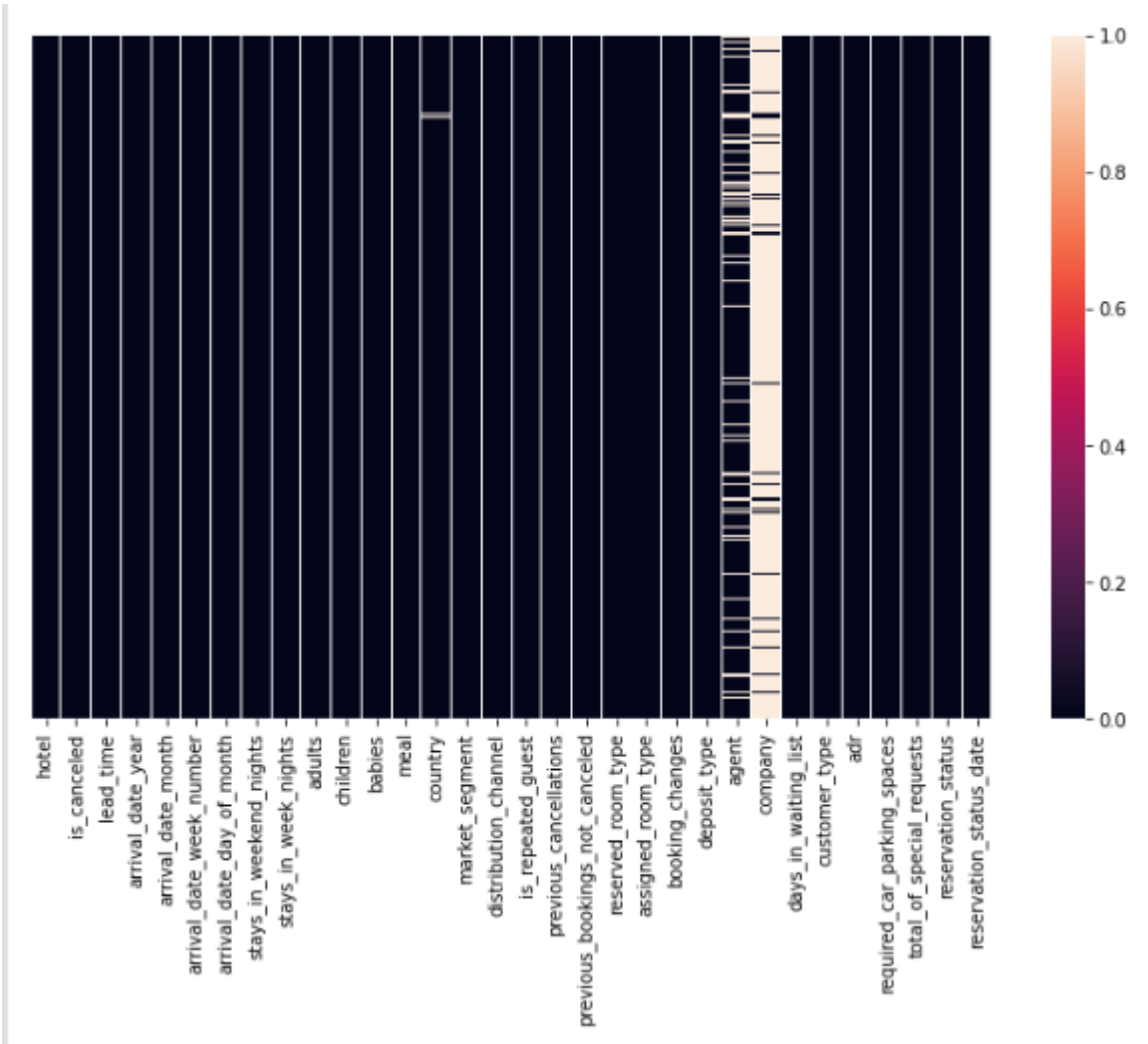


Figura 22. Gráfica valores nulos en el conjunto de datos. Fuente: contenido propio.

Se observa en color blanco las casillas de las columnas que contienen valor nulo. En la columna *company* son el valor más frecuente.

CAPÍTULO 4: Aplicación de regresores

4.1 Problema a estudiar

Se aspira a crear un modelo de regresión para estimar la tasa media diaria a partir de un conjunto de variables independientes que tienen un alto coeficiente de correlación con la variable objetivo. Estas variables son seleccionadas mediante algoritmos de selección de características que se explicarán más adelante.

La métrica de éxito que se va a usar es el coeficiente de determinación R^2.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Figura 23. Fórmula coeficiente de determinación. Fuente: *[55]*

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \text{ and } \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2.$$

Figura 24. Términos Fórmula Coeficiente de determinación. Fuente: [55]

Por otro lado, la métrica de error será el error cuadrático medio. Este se calcula como el sumatorio del cuadrado de la diferencia los datos reales y los datos predichos. Esto a su vez se multiplica por la inversa del número total de datos de las variables predictoras.

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

Figura 25. Fórmula Error Cuadrático Medio. Fuente: *[56]*

El método de aprobación de los resultados es la validación cruzada. Se usa en la fase de entrenamiento y en la de testeo. En cada fase el número de pliegues variará en función del algoritmo.

Generalmente oscila entre un mínimo de 4 y un máximo de 10 particiones. *n – 1* particiones se usan para entrenar, y la *n-ésima* se usa para validar el modelo.

4.2 Algoritmos empleados

4.2.1 Regresión Lineal

El análisis de Regresión es un subcampo del aprendizaje automático supervisado [57]. Su propósito es establecer un modelo para la relación entre un cierto número de características y una variable objetivo contínua.

En los problemas de regresión se persigue obtener una respuesta cuantitativa, como por ejemplo, predicciones sobre precios de inmuebles o la temperatura que hará en los próximos dos días.

Dado un conjunto de puntos, el algoritmo de regresión establecerá un modelo para ajustar la relación de dependencia entre una característica específica independiente (un valor de la variable independiente *x*) y el valor resultado correspondiente (un valor de la variable dependiente *y*).

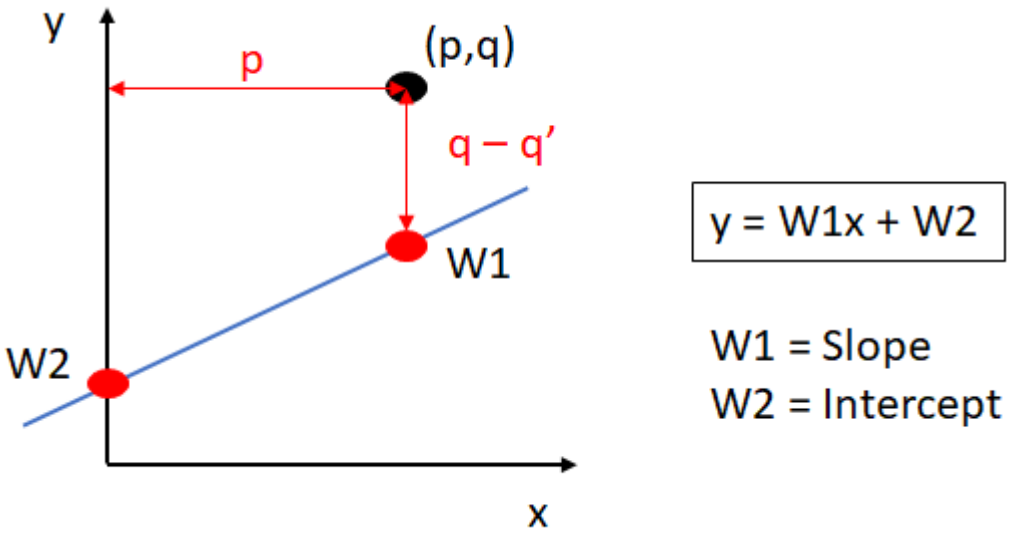


Figura 26. Recta de Regresión. Regresión para el caso de una recta. Fuente: *[57]*

En esta gráfica encontramos los siguientes elementos:

W1: *Slope* o pendiente, es la inclinación de la recta con respecto al eje de abcisas (x).

W2: *Intercept* o intersección, es el punto de corte de la recta con el eje de ordenadas (y).

El objetivo es conseguir que la recta se acerque al punto. Para ello, el algoritmo utilizará un parámetro llamado *coeficiente de regresión*. Esta tasa de aprendizaje es el número por el que se multiplicarán los parámetros de la recta para realizar pequeñas aproximaciones de la recta al punto.

La tasa de aprendizaje determinará la longitud de la distancia cubierta en cada interacción que hará que la recta se acerque al punto. Normalmente se representa por *α*.

$$y = (W_1 + p\alpha)x + W_2 + \alpha$$

Figura 27. Algoritmo regresión lineal. Aproximación absoluta. Fuente: [57]

4.2.2 K-Vecinos cercanos

Es un método cuyo modo de funcionar se basa basa en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de los datos que le rodean.

El número de observaciones próximas a considerar es un parámetro que se le especifica al algoritmo.

La heurística del algoritmo es la siguiente:

1. Dado un punto que queremos predecir su función objetivo. Calculamos la distancia entre este punto y el resto de observaciones. La distancia se calcula mediante las fórmulas de distancia de Manhattan o distancia euclídea.

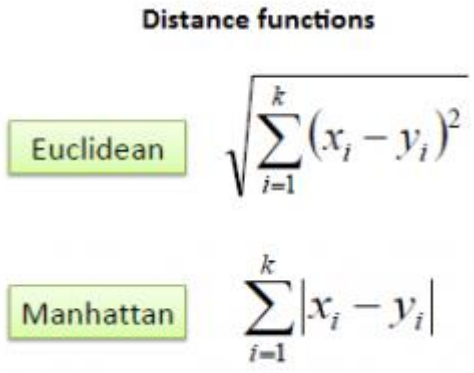


Figura 28. Fórmulas distancia algoritmo K-vecinos más próximos. Fuente: *[58]*

2. Se seleccionan los *n* puntos más cercanos (basados en la distancia calculada anteriormente). *n* es el número de vecinos más cercanos.

3. La estimación de la función objetivo para el punto concreto se calcula como la media de los *n* vecinos más próximos seleccionados en el paso 2.

4. Una función de error es aplicada para calcular la diferencia entre los puntos

estimados en el paso 3 y los datos reales de la variable objetivo.

4.2.3 Regresión polinomial

La regresión polinomial es una forma de regresión lineal en la que la relación entre la variable independiente *x* y la variable dependiente *y* es modelada como un polinomio de grado *n* en *x*.

Consiste en incorporar flexibilidad a un modelo lineal introduciendo nuevos predictores obtenidos al elevar a distintas potencias el predictor original.

Partiendo del modelo lineal:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

Figura 29. Regresión Polinómica. Modelo Lineal. Fuente: *[59]*

Se obtiene un modelo polinómico de grado *d* a partir de la ecuación:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$$

Figura 30. Regresión Polinómica. Modelo no lineal Fuente: *[59]*

4.2.4 Regularización de Tíjonov

La regresión de Tíjonov (también conocida como regresión de Ridge) difiere de la regresión lineal en que añade una restricción a la función de coste que implica multiplicar el parámetro de regresión por un coeficiente lambda que es especificado por el usuario.

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

Figura 31. Función de coste Ridge Regression. Fuente: *[60]*

4.2.5 LASSO

La función de coste del método LASSO (*least absolute shrinkage and selection operator*, por sus siglas en inglés) difiere del método anterior en que no eleva al cuadrado el coeficiente de regresión *w*.

Esto implica que a similitud del Ridge se aproximan a cero los coeficientes muy altos. Adicionalmente, el LASSO elimina del modelo las variables predictoras penalizadas por el Ridge.

$$\sum_{i=1}^{M}(y_i-\hat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{p}w_j\times x_{ij}\right)^2+\lambda\sum_{j=0}^{p}|w_j|$$

Figura 32. Función de coste algoritmo LASSO.Fuente: [60]

4.3 Selección de variables

La selección de variables es una etapa importante en el aprendizaje automático. Supone encontrar el conjunto de características que mejor describen la variable objetivo que se pretende estimar.

Se divide en tres métodos principales:

- *Métodos de filtrado*

  Como su nombre indica, en este método se filtra y se toma sólo el subconjunto de las características relevantes. El modelo se construye después de seleccionar las características. El filtrado aquí se hace usando una matriz de correlación y se hace más comúnmente usando la correlación de *Pearson* [61].

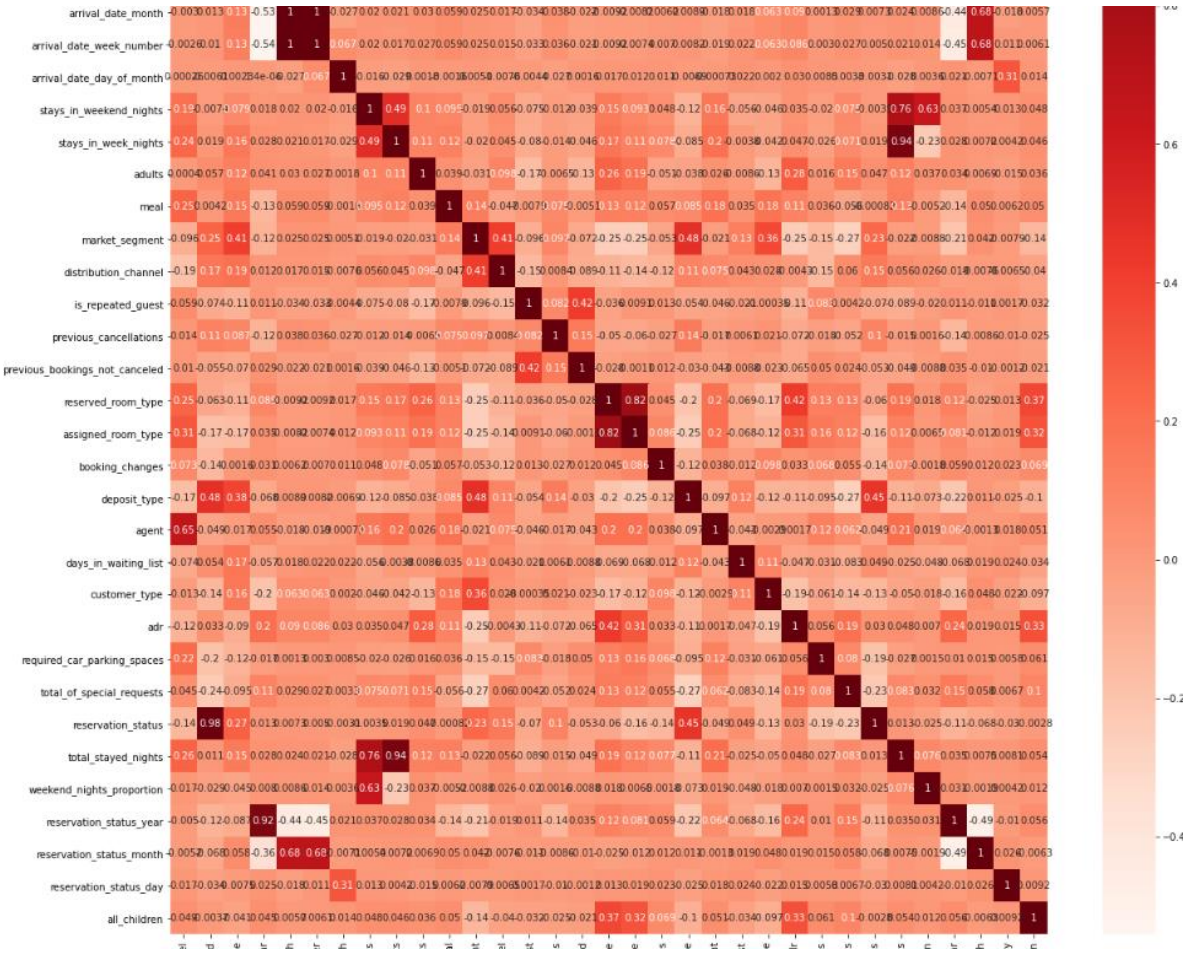


Figura 33. Mapa de correlación conjunto de datos. Fuente: contenido propio

- *Métodos de envoltura*

Un método de envoltura necesita un algoritmo de aprendizaje (habitualmente algoritmos de regresión) y utiliza su rendimiento como criterio de evaluación. Esto significa que se alimentan las características del algoritmo de aprendizaje automático seleccionado y, en función del rendimiento del modelo, se añaden/eliminan las características.

Destacan los algoritmos de Eliminación por atrás y el de eliminación recursiva.

```
cols = list(X.columns)
pmax = 1
while (len(cols)>0):
    p= []
    X_1 = X[cols] #
    X_1 = sm.add_constant(X_1)
    model = sm.OLS(list(y),X_1.astype(float)).fit()
    p = pd.Series(model.pvalues.values[1:],index = cols)
    pmax = max(p)
    feature_with_p_max = p.idxmax()
    if(pmax>0.05):
        cols.remove(feature_with_p_max)
    else:
        break
selected_features_BE = cols
print(selected_features_BE)
```

```
['hotel', 'is_canceled', 'lead_time', 'arrival_date_year', 'arrival_date_month', 'arrival_date_week_number', 'arrival_date_day_of_month',
```

Figura 34. Selección de Variables. Eliminación por Atrás. Fuente: Contenido propio.

- Métodos embebidos

  En esta categoría se emplean algoritmos de regularización, como el *Ridge* y *Lasso*, que como se ha explicado, en su función de coste a medida que se ejecuta el modelo van descartando las variables que peor correlacionan. Por ello, también se emplean en la selección de características.

```
[11] reg = LassoCV()
     reg.fit(X, y)
     print("Best alpha using built-in LassoCV: %f" % reg.alpha_)
     print("Best score using built-in LassoCV: %f" %reg.score(X,y))
     coef = pd.Series(reg.coef_, index = X.columns)

     /usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_coordinate_descent.py:472: ConvergenceWarning: Objective did not converge.
       tol, rng, random, positive)
     Best alpha using built-in LassoCV: 0.435448
     Best score using built-in LassoCV: 0.422050

     print("Lasso picked " + str(sum(coef != 0)) + " variables and eliminated the other " +  str(sum(coef == 0)) + " variables")

     Lasso picked 25 variables and eliminated the other 7 variables

[13] imp_coef = coef.sort_values()
     import matplotlib
     matplotlib.rcParams['figure.figsize'] = (15.0, 15.0)
     imp_coef.plot(kind = "barh")
     plt.title("Feature importance using Lasso Model")
```

Figura 35. Ejecución algoritmo LASSO. Fuente: Contenido propio.

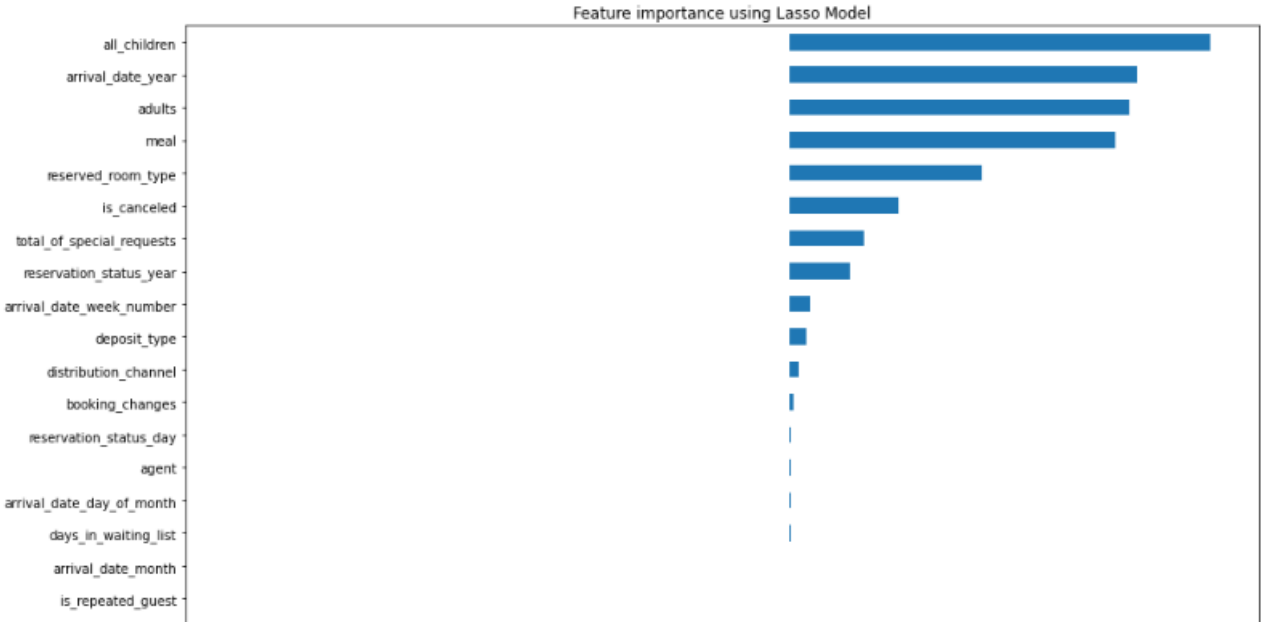


Figura 36. Selección variables LASSO. Fuente: contenido propio

4.4 Resultados Regresión con estandarización

La estandarización de los datos es un proceso que se engloba en lo que se conoce como el escalado de datos.

En muchas ocasiones las medidas y dimensiones de los datos difieren mucho. Por ejemplo, una variable masa puede medir la capacidad de un contenedor de mercancías en 20000 kilogramos, y puede existir una variable longitud en metros que toma el valor 10. Estas diferencias en los datos no son toleradas por los algoritmos de regresión, ya que ponderará mejor la característica de la masa frente a la longitud al tener un rango de valores mucho más alto.

Por consiguiente, los procedimientos de normalización son útiles para convertir el conjunto de datos a una distribución normal. La fórmula de la distribución es la siguiente:

$$Xstand = \frac{x - mean(x)}{standard\ deviation\ (x)}$$

Figura 37. Fórmula estandarización. Fuente: *[62]*

```
[ ] # standarize

    from sklearn.preprocessing import StandardScaler

    scaler = StandardScaler()

    data_transformed = scaler.fit_transform(dataCopy)
    print(data_transformed.shape)
```

Figura 38. Proceso de Estandarización de los datos. Fuente: Contenido propio

Los resultados que se muestran corresponden al coeficiente de determinación R^2. Se trabaja con un número de pliegues variable. Debido a la dispersión de los datos, un efecto ocasionado principalmente por la conversión numérica de las variables categóricas.

Algoritmo	N = 3	N = 4	N = 6	N = 8	N = 10
Regresión Lineal	0,317	0,291	0,246	0,371	0,155
K-vecinos cercanos (K=9)	0,361	0,375	0,245		0,367
Regresión polinomial	0,382	0,409	0,262	0,398	0,155
Ridge	0,416	0,386	0,253	0,401	0,173
LASSO	0,392	0,354	0,273	0,423	0,175

Tabla 2. Tabla Resultados Regresión con Estandarización. Fuente: Contenido propio.

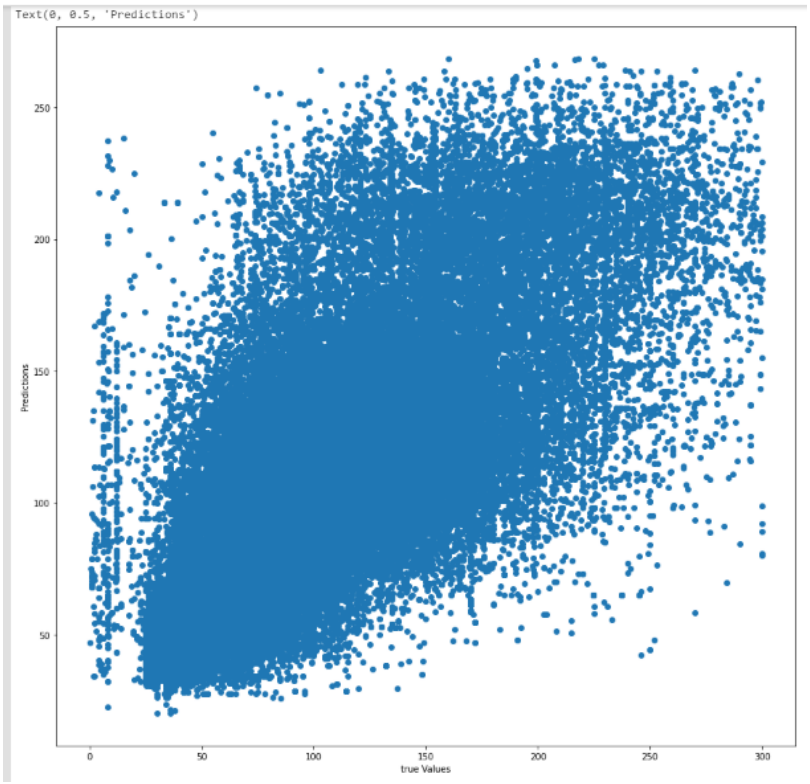


Figura 39. Resultados Regresión K-Vecinos Próximos (con estandarización). Fuente: Contenido propio.

En esta parte el que mejor resultados ha arrojado ha sido el K-Vecinos más próximos, con un coeficiente de determinación de 0,36, lo cual se visualiza en la gráfica, con los precios estimados agrupados en torno a la recta de regresión.

4.5 Resultados Regresión sin estandarización

A continuación, se presentan los resultados cuando se le suministra a los algoritmos un número variable de las *n* mejores características seleccionadas por los algoritmos de filtrado. Las 8 mejores características devueltas por LASSO, por orden decreciente de influencia en la variable objetivo (*adr*), son: *reserved_room_type, arrival_date_month, all_children, meal, adults, arrival_date_year, reservation_status_year, is_canceled.*

Los resultados que se muestran corresponden al coeficiente de determinación R^2.

Algoritmo	N = 3	N = 4	N = 6	N = 8	N = 10
Regresión Lineal	0,198	0,241	0,227	0,266	0,270
K-Vecinos cercanos (K=9)	0,214	0,209	0,293	0,231	0,07
Regresión polinomial	0,181	0,191	0,186	0,223	0,181
Ridge	0,202	0,253	0,242	0,285	0,290
Lasso	0,193	0,234	0,223	0,267	0,269

Tabla 3. Resultados Regresión Sin estandarización. Fuente: contenido propio.

Si bien el coeficiente de determinación es una medida que explica la relación de las variables independientes con la variable objetivo, cabe decir que se trata de un problema muy complejo, dado que cuantificar una cantidad numérica que equivale a un precio depende de muchas más variables que no se encuentran disponibles en los datos facilitados por los autores al momento de finalizar este estudio.

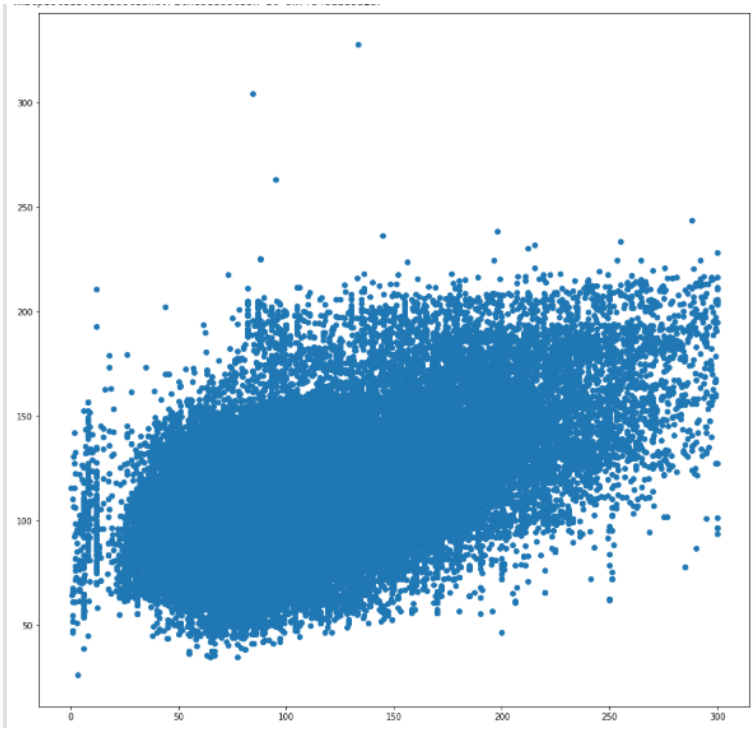


Figura 40. Resultados Regresión Lineal (Sin estandarizar). Fuente: contenido propio

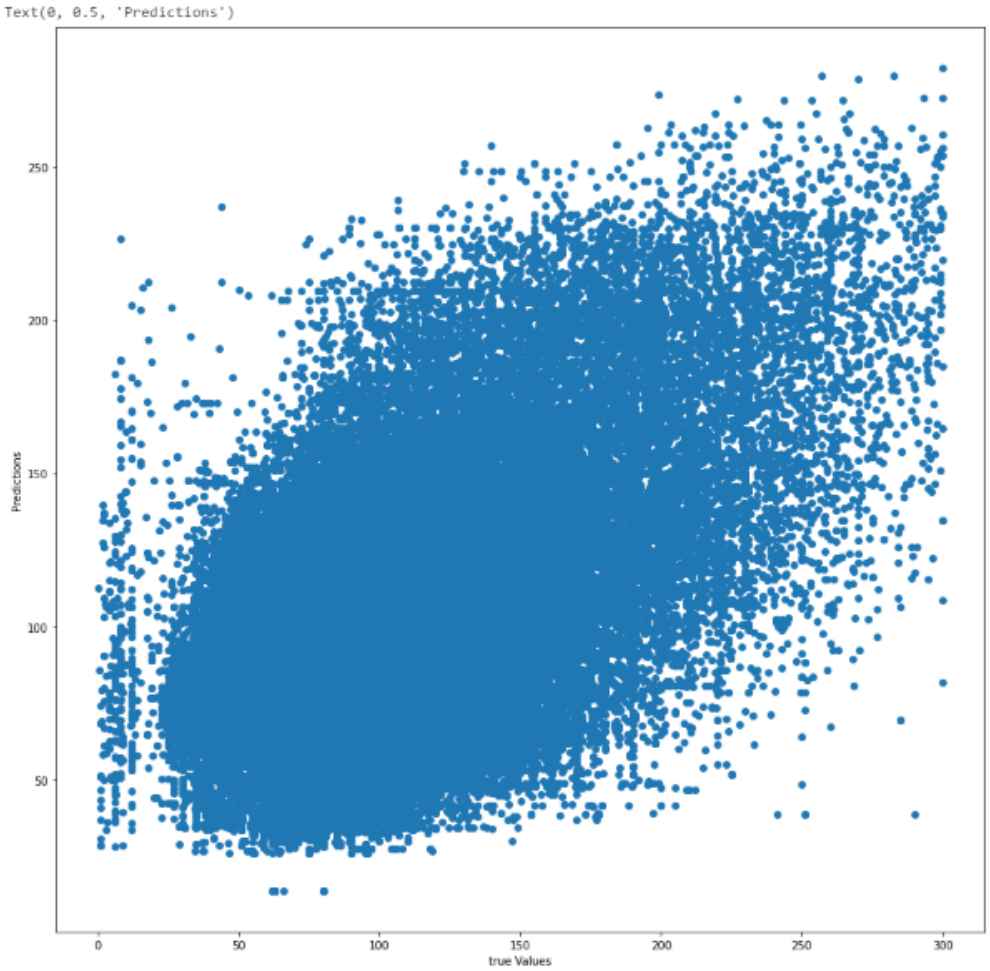


Figura 41 Resultados Regresión k-vecinos próximos (Sin estandarizar). Fuente: contenido propio.

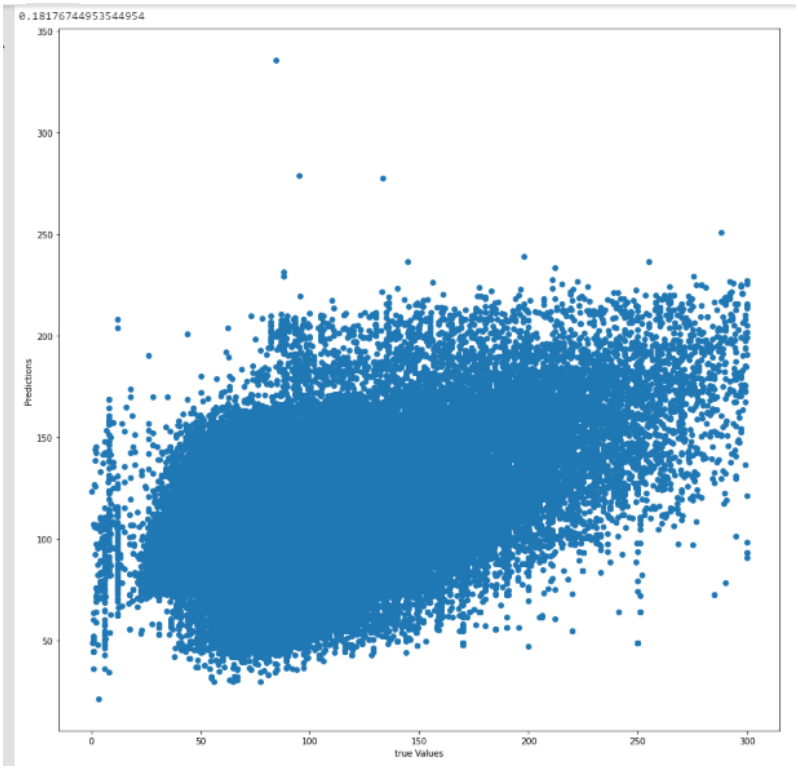


Figura 42. Regresión Polinómica (sin estandarizar). Fuente: contenido propio.

Un aspecto a destacar es que la tendencia de los datos es a seguir la recta de regresión y posicionarse en torno a ella, si bien debido a su abundancia y a las múltiples variables independientes, las métricas de dispersión como la varianza y la desviación típica son altos.

4.5 Resultados Regresión con reducción de la dimensionalidad (PCA)

El PCA (del término en inglés *Principal Component Analysis*) [63] es una técnica que busca reducir la cantidad de datos sin perder representatividad de los mismos.

El proceso del PCA finaliza cuando se localiza un vector que maximiza la varianza en el primer eje, y un segundo vector ortogonal al anterior y de inferior varianza.

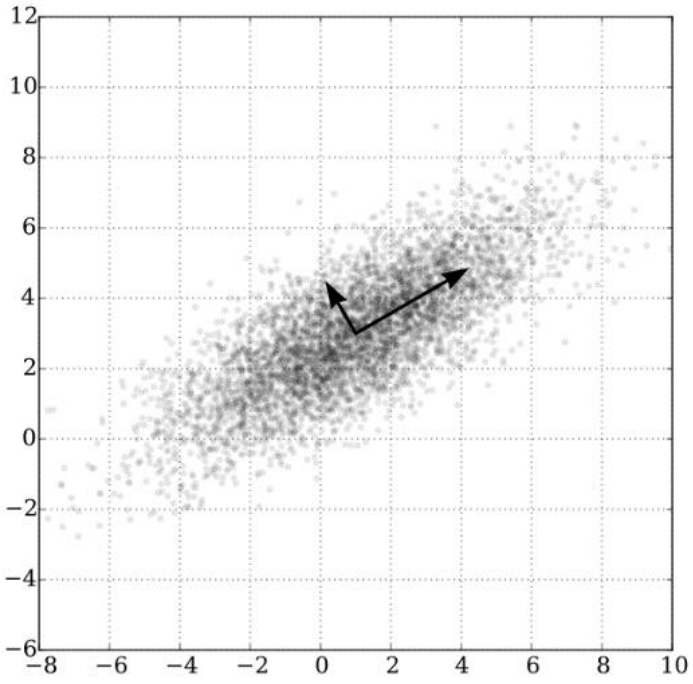


Figura 43. Ilustración Análisis del Componente Principal. Fuente: *[64]*

4.5.1 Resultados PCA con estandarización

Para poder contrastar los resultados, se ha optado por implementar dos versiones del Análisis del Componente Principal, con y sin estandarización. Se han obtenido los siguientes resultados en la ejecución de los diferentes algoritmos:

```
[ ]  from sklearn.decomposition import PCA

     pca = PCA(n_components = 3)


     x_train = pca.fit_transform(x_train)
     x_test = pca.fit_transform(x_test)

     print('x_train ', x_train)
     print('x_test ', x_test)
```

Figura 44. Proceso de Análisis del Componente Principal. Fuente: contenido propio.

Algoritmo	**Coeficiente de Determinación R^2**	**Error cuadrático medio (ECM)**
Regresión Lineal	0,533	962,63
K-Vecinos Próximos	0,526	963,65
Regresión Polinomial	-1,13	15272,56
Ridge	0,526	31,773
LASSO	0,507	32,451

Tabla 4. Resultados PCA con estandarización. Fuente: contenido propio.

En general la mejora en los resultados de regresión es notable con Análisis del Componente Principal. Con coeficientes de R^2 que superan 0,5.

```
from sklearn.linear_model import Lasso

lasso = Lasso()

lasso.fit(x_train,y_train)

scores = cross_val_score(lasso, X, y, cv=5)
print ('Cross-validated scores:', scores)


lasso_predictions = cross_val_predict(lasso, X, y, cv=5)
plt.scatter(y, lasso_predictions)
plt.xlabel('true Values')
plt.ylabel('Predictions')

rmse = np.sqrt(metrics.mean_squared_error(y, lasso_predictions))
r2 = metrics.r2_score(y, lasso_predictions)
print(rmse)
print(r2)
```

```
Cross-validated scores: [0.51683021 0.46575487 0.41701565 0.40570619 0.47392699]
32.45126415061695
0.48966030567482277
```

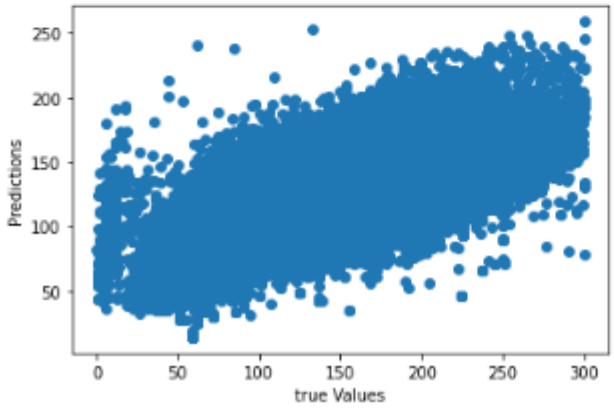


Figura 45. Modelo LASSO en Análisis Componente Principal (con estandarización). Fuente: contenido propio.

4.5.2 PCA sin estandarización

Algoritmo	Coeficiente de Determinación R^2	Error cuadrático medio (ECM)
Regresión Lineal	**0,533**	**962,72**
K-Vecinos próximos	**0,216**	**1616,199**
Regresión Polinomial	**0,510**	**31,77**
Ridge	**0,511**	**31,75**
LASSO	**0,397**	**35,25**

Tabla 5. Resultados PCA sin normalización. Fuente: Contenido propio.

Como se puede ver, no es notable el efecto de la estandarización, ya que sin ella el

PCA sigue comprimiendo los datos de tal manera que están bastante cohesionados y escalados.

```
from sklearn.linear_model import Lasso

lasso = Lasso()

lasso.fit(x_train,y_train)

scores = cross_val_score(lasso, X, y, cv=5)
print ('Cross-validated scores:', scores)


lasso_predictions = cross_val_predict(lasso, X, y, cv=5)
plt.scatter(y, lasso_predictions)
plt.xlabel('true Values')
plt.ylabel('Predictions')

rmse = np.sqrt(metrics.mean_squared_error(y, lasso_predictions))
r2 = metrics.r2_score(y, lasso_predictions)
print(rmse)
print(r2)
```

```
Cross-validated scores: [0.37881582 0.35940423 0.29971297 0.42464942 0.40010548]
35.258996059589116
0.3975292111882055
```

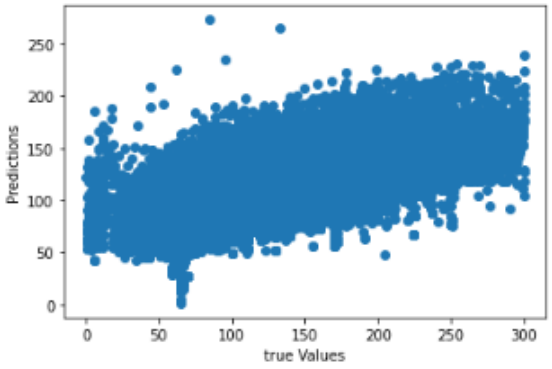


Figura 46. Algoritmo LASSO PCA sin normalización. Fuente: contenido propio

CAPÍTULO 5: Aplicación de clasificadores

5.1 Problema a estudiar

La pregunta planteada en este apartado se refiere a crear un modelo de predicción de las reservas que se van a cancelar. Dentro del conjunto de datos, los algoritmos examinarán unas variables de entrada y asociarán a cada fila un valor de cancelado o no cancelado.

5.2 Selección de variables

Existen métodos específicos de selección de variables en problemas de clasificación. En este ejemplo particular, se va a usar el *Extreme Gradient Boosting* [65].

El primer paso a la hora de aplicar un algoritmo de clasificación es hacer un ajuste de parámetros, mediante algoritmos que para un problema particular devuelven los valores que se le deben suministrar a cada clasificador para mejorar el rendimiento. Uno de los algoritmos que se emplean para esta tarea es el *GridSearchCV* [66].

```
## Encontrando parámetros para el Decision Tree

model_dtc_gs = DecisionTreeClassifier()
parameters_dtc = {
'criterion' : ['gini', 'entropy'],
'min_samples_split' : [2,4,6,8],
'min_samples_leaf': [1,2,3,4,5],
'max_features' : ['auto', 'sqrt']
 }

grid_search_dtc = GridSearchCV(estimator=model_dtc_gs, param_grid=parameters_dtc,
                               cv=5, scoring='f1', verbose=True, n_jobs =-1)
grid_search_dtc.fit(X, y)
grid_search_dtc.best_params_
```

```
Fitting 5 folds for each of 80 candidates, totalling 400 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done  46 tasks      | elapsed:    8.1s
[Parallel(n_jobs=-1)]: Done 196 tasks      | elapsed:   33.3s
[Parallel(n_jobs=-1)]: Done 400 out of 400 | elapsed:  1.2min finished
{'criterion': 'entropy',
 'max_features': 'sqrt',
 'min_samples_leaf': 3,
 'min_samples_split': 8}
```

Figura 47. Ajuste de parámetros mediante GridSearchCV. Fuente: contenido propio.

Vemos que para el ejemplo particular del clasificador *Decision Tree*, la función ha devuelto los parámetros a escoger para maximizar el rendimiento.

El siguiente paso consiste en enviar esos parámetros al algoritmo de selección de variables para que compute las variables que mejor correlacionan con la variable objetivo.

Como paso siguiente, la selección de variables se produce invocando una función que recibe la lista de parámetros óptimos obtenida en el paso 1. Entonces nos devolverá

```
params = {
    'criterion': 'giny',
    'learning_rate': 0.01,
    'max_depth': 5,
    'n_estimators': 100,
    'objective': 'binary:logistic',
}
model = XGBClassifier(parameters=params)
# fit the model
model.fit(X, y)
# perform permutation importance
result = permutation_importance(model, X, y, scoring='accuracy', n_repeats = 5, n_jobs=-1)
sorted_idx = result.importances_mean.argsort()

for i,v in enumerate(sorted_idx):
    print('Feature: %0d, Score: %.5f' % (i,v))
```

```
Feature: 0, Score: 4.00000
Feature: 1, Score: 14.00000
Feature: 2, Score: 9.00000
Feature: 3, Score: 28.00000
Feature: 4, Score: 3.00000
Feature: 5, Score: 22.00000
Feature: 6, Score: 0.00000
Feature: 7, Score: 5.00000
Feature: 8, Score: 10.00000
Feature: 9, Score: 7.00000
Feature: 10, Score: 13.00000
Feature: 11, Score: 8.00000
Feature: 12, Score: 27.00000
Feature: 13, Score: 6.00000
Feature: 14, Score: 24.00000
Feature: 15, Score: 16.00000
Feature: 16, Score: 23.00000
Feature: 17, Score: 17.00000
Feature: 18, Score: 19.00000
Feature: 19, Score: 2.00000
Feature: 20, Score: 18.00000
Feature: 21, Score: 25.00000
Feature: 22, Score: 21.00000
Feature: 23, Score: 15.00000
```

Figura 48. Selección de variables problema clasificación. Fuente: contenido propio.

5.3 Algoritmos Empleados

Se han escogido clasificadores que siguen distintas estrategias para construir el árbol de decisiones hacia la variable objetivo.

- **Random Forest** [67]

  También conocidos en castellano como “Bosques Aleatorios" es una combinación de árboles predictores tal que cada árbol depende de los

valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

- **Decision Tree** [68]

  Los árboles de decisión son representaciones gráficas de posibles soluciones a una decisión basadas en ciertas condiciones.

  El aprendizaje del árbol de decisiones utiliza un árbol de decisiones (como modelo predictivo) para pasar de las observaciones sobre un elemento (representado en las ramas) a las conclusiones sobre el valor objetivo del elemento (representado en las hojas).

- **XGBoost** [69]

  XGBoost Extreme Gradient Boosting es un algoritmo predictivo supervisado que utiliza el principio de boosting.

  La idea detrás del boosting es generar múltiples modelos de predicción débiles (baja correlación con la variable que se pretende clasificar) secuencialmente, y que cada uno de estos tome los resultados del modelo anterior, para generar un modelo más "fuerte", con mejor poder predictivo y mayor estabilidad en sus resultados.

- **Regresión Logística** [70]

  La regresión logística permite calcular la probabilidad de que la variable dependiente pertenezca a cada una de las dos categorías en función del valor que adquiera la variable independiente.

- **Naive Bayes** [71]

  En ellos se asume que las variables predictoras son independientes entre sí. En otras palabras, que la presencia de una cierta característica en un conjunto de datos no está en absoluto relacionada con la presencia de cualquier otra característica.

  Lo consiguen proporcionando una forma de calcular la probabilidad 'posterior' de que ocurra un cierto evento A, dadas algunas probabilidades de eventos 'anteriores'.

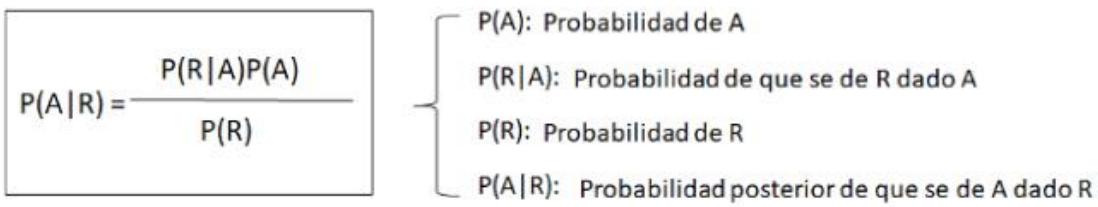


Figura 49. Fórmula Naive Bayes. Fuente: *[72]*

```
[ ] dtc_model = DecisionTreeClassifier(criterion= 'gini', min_samples_split=8,
                                       min_samples_leaf = 4, max_features = 'auto')
    # fit the model
    dtc_model.fit(X_train, y_train)

    #Predict Model
    predict_dtc = dtc_model.predict(X_test)
```

```
[ ]
    rf_model = RandomForestClassifier(min_samples_leaf = 6, min_samples_split=6,
                                      n_estimators = 100)

    # fit the model
    estimator= rf_model.fit(X_train, y_train)
    #Predict Model
    predict_rf = rf_model.predict(X_test)
```

```
    etc_model = ExtraTreesClassifier(min_samples_leaf = 7, min_samples_split=2,
                                     n_estimators = 100)
    # fit the model
    etc_model.fit(X_train, y_train)

    #Predict Model
    predict_etc = etc_model.predict(X_test)
```

```
[ ] # Extreme Gradient Boosting Model Building

    xgb_model = XGBClassifier(criterion = 'giny', learning_rate = 0.01, max_depth = 5, n_estimators = 100,
                              objective ='binary:logistic', subsample = 1.0)
    # fit the model
    xgb_model.fit(X_train, y_train)
    #Predict Model
    predict_xgb = xgb_model.predict(X_test)
```

```
    knn_model = KNeighborsClassifier(n_neighbors=9)
```

Figura 50. Implementación modelos clasificación. Fuente: contenido propio.

5.4 Comparativa de resultados

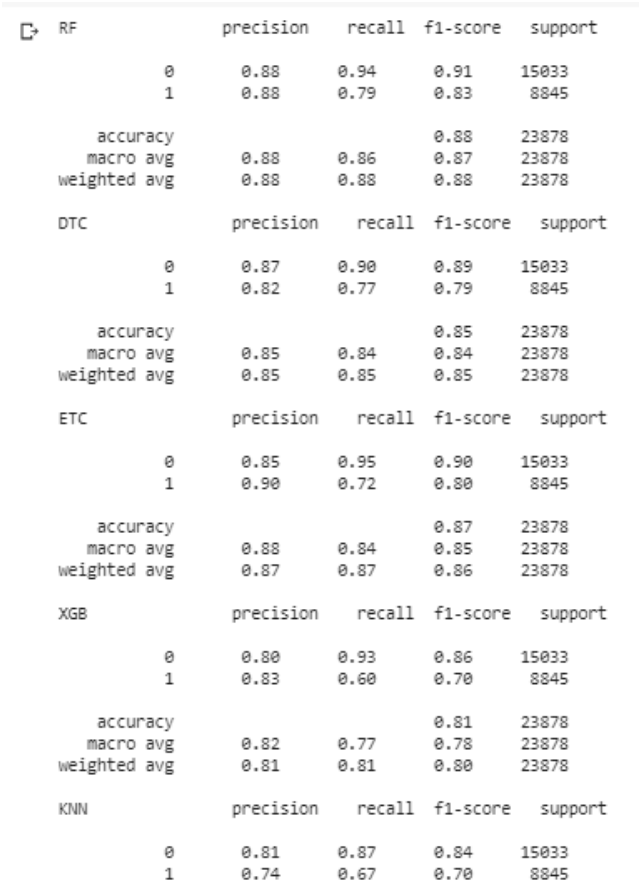
En figura anterior se presentan distintas medidas relativas a la precisión (número de verdaderos positivos sobre la suma entre estos últimos y los falsos positivos), el *recall* o exhaustividad, (verdaderos positivos entre falsos negativos y verdaderos positivos), entre otras.

De acuerdo a las anteriores medidas, *Random Forest* tiene la tasa de predicción

correcta más alta con un 88%.de casos correctamente clasificados.

Random Forest y el *Clasificador de Árboles Extra* comparten los más altos índices de precisión. Significa que ambos modelos predijeron correctamente alrededor del 88% de todas las etiquetas positivas.

Por otra parte, *Random Forest* tiene el mayor índice de exhaustividad. Significa que este modelo predijo correctamente el 79% de las observaciones positivas reales.



```
RF              precision    recall  f1-score   support

           0       0.88      0.94      0.91     15033
           1       0.88      0.79      0.83      8845

    accuracy                           0.88     23878
   macro avg       0.88      0.86      0.87     23878
weighted avg       0.88      0.88      0.88     23878

DTC             precision    recall  f1-score   support

           0       0.87      0.90      0.89     15033
           1       0.82      0.77      0.79      8845

    accuracy                           0.85     23878
   macro avg       0.85      0.84      0.84     23878
weighted avg       0.85      0.85      0.85     23878

ETC             precision    recall  f1-score   support

           0       0.85      0.95      0.90     15033
           1       0.90      0.72      0.80      8845

    accuracy                           0.87     23878
   macro avg       0.88      0.84      0.85     23878
weighted avg       0.87      0.87      0.86     23878

XGB             precision    recall  f1-score   support

           0       0.80      0.93      0.86     15033
           1       0.83      0.60      0.70      8845

    accuracy                           0.81     23878
   macro avg       0.82      0.77      0.78     23878
weighted avg       0.81      0.81      0.80     23878

KNN             precision    recall  f1-score   support

           0       0.81      0.87      0.84     15033
           1       0.74      0.67      0.70      8845
```

Figura 51. Algoritmos clasificación. Fuente: contenido propio.

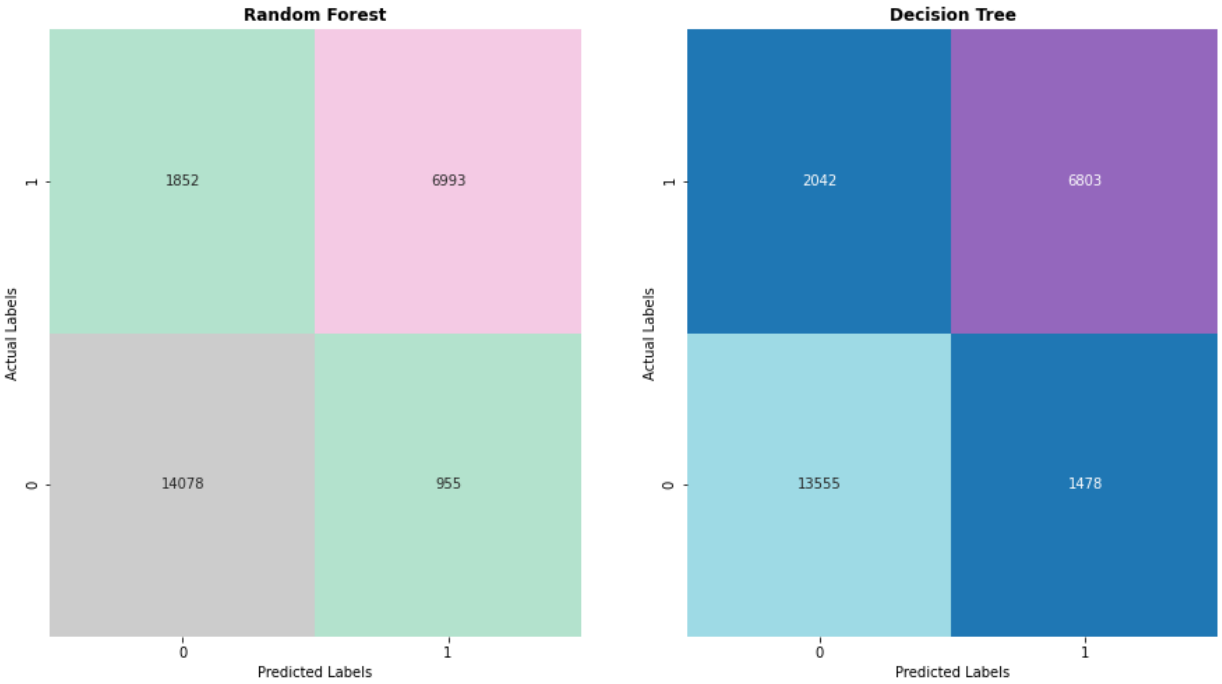


Figura 52. Matrix Confusión Random Forest y Decision Tree. Fuente: contenido propio.

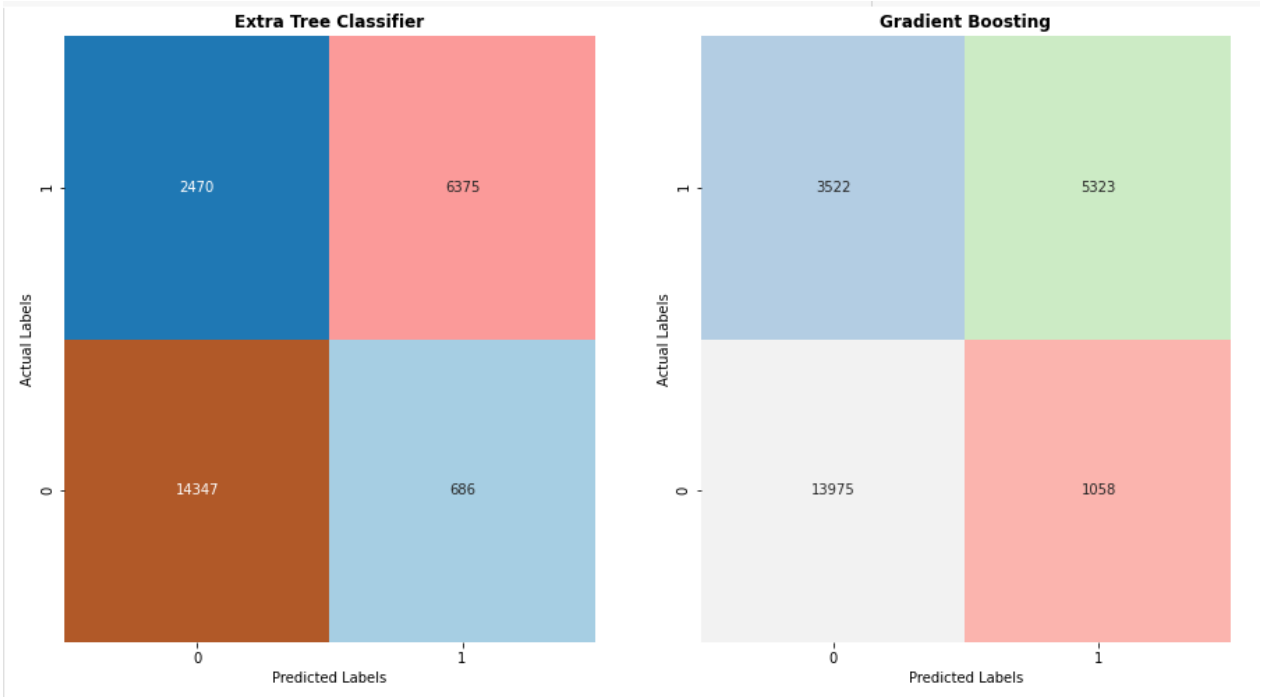


Figura 53. Extra Tree Classifier y Gradient Boosting. Fuente: contenido propio.

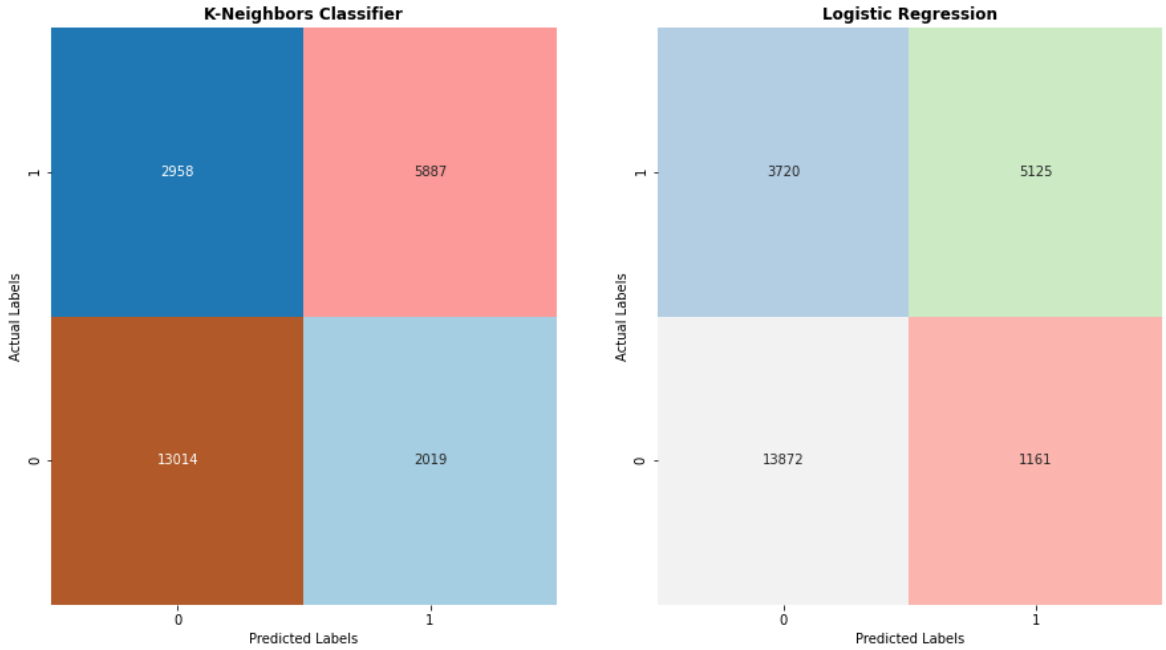


Figura 54. Matriz confusión K-vecinos próximos y regresión logística. Fuente: contenido propio.

CAPÍTULO 6

6.1 Conclusiones y líneas futuras

Como conclusión general, cabe decir que se han podido construir modelos que aproximaban la comprensión y resolución de los problemas planteados.

En el apartado de la regresión, se han afrontado dificultades para conseguir la selección de datos que mejor se ajustara al objetivo. Los algoritmos de selección de características han contribuido positivamente a minimizar el conjunto de variables en el proceso.

Asimismo, a lo largo del proceso de implementación de los algoritmos, se han descubierto singularidades en los datos, como los valores atípicos del precio de alojamiento. La codificación *Dummy* para las variables categóricas, etc.

En el apartado de la regresión, los resultados demostraron ser más robustos cuando se aplicó una reducción de la dimensionalidad a dos variables. Obteniéndose unos coeficientes de correlación entre los datos reales y estimados más altos.

En cuanto a la clasificación, el problema de predecir la cancelación se pudo tratar con una selección de algoritmos que en general, gracias al ajuste de parámetros con *GridSearchCV*, y la posterior selección de variables con *Extreme Gradient Boosting*, dieron buenos resultados en cuanto a precisión, exhaustividad y demás métricas.

El Random Forest ha podido predecir correctamente si una reserva se cancela o no en un 80% de los datos.

Por último, cabe señalar que se trata de un problema complejo dada la relación entre el tamaño de las variables y el número de filas. Los algoritmos de preprocesado y selección de variables han sido de gran ayuda para simplificar el problema.

En un futuro cercano, como líneas de trabajo a explorar se encuentran las otras categorías del aprendizaje automático, el no supervisado y el semisupervisado. Ver si hay patrones sistemáticos en las reservas. Estudiar asociaciones y agrupamientos en los datos. Qué tipo de reservas son más comunes, y si existen varios grupos de clientes en función del precio abonado por el alojamiento.

Por último, cabe señalar que se trata de un problema complejo dado la relación entre el tamaño de las variables y el número de filas. Los algoritmos de preprocesado y selección de variables han sido de gran ayuda para simplificar el problema.

En un futuro cercano, como líneas de trabajo a explorar se encuentran las otras

categorías del aprendizaje automático, el no supervisado y el semisupervisado. Ver si hay patrones sistemáticos en las reservas. Estudiar asociaciones y agrupamientos en los datos. Qué tipo de reservas son las más comunes, y si hay diferentes grupos dependiendo del precio pagado por el alojamiento.

6.2 Conclusions and future work

As a general conclusion, it can be said that it has been possible to build models that approximate the understanding and resolution of the problems posed.

In the regression section, difficulties have been faced to achieve the selection of data that best fits the objective. The algorithms of selection of characteristics have contributed positively to minimize the set of variables in the process.

Also, throughout the process of implementing the algorithms, singularities in the data have been discovered, such as outliers of the accommodation price. Dummy coding for categorical variables, etc.

In the regression section, the results proved to be more robust when a reduction in dimensionality to two variables was applied. Higher correlation coefficients between actual and estimated data were obtained.

As for the classification, the problem of predicting cancellation could be treated with a selection of algorithms that in general, thanks to the adjustment of parameters with GridSearchCV, and the subsequent selection of variables with Extreme Gradient Boosting, gave good results in terms of accuracy, completeness and other metrics.

Random Forest has been able to correctly predict whether or not a reservation is cancelled in 80% of the data.

7. Presupuesto

7.1 Coste de Hardware

Tipo	Descripción	Precio
Portátil Lenovo G50-80	Ordenador utilizado para realizar el proyecto	Vida útil ordenador: 4 años Coste: 700 €

		Coste amortizado = $\frac{700}{(12 \cdot 4)} \cdot 8 = 117$ € Cuantificable en tiempo de vida, un consumo de 8 meses de trabajo en este proyecto, en términos de CPU, Tarjeta Gráfica, memoria RAM y batería de carga.

Tabla 6. Coste del Hardware. Tabla: Contenido propio.

7.2 Coste de Recursos Humanos

Horas de trabajo estimadas	Coste por hora	Total
350	15 €	5250 €

Tabla 7. Coste de Recursos Humanos. Tabla: Contenido propio

7.3 Coste Total

Hardware	Recursos Humanos	Total
117 €	5250 €	5367 €

Tabla 8. Costes Totales. Tabla: contenido propio.

Apéndice 1

Enlace al repositorio de GitHub que contiene los datos utilizados y todos los cuadernos de *Google Collab*.

Bibliografía

[1] N. Antonio, A. de Almeida y L. Nunes, «Hotel booking demand datasets,» *Science Direct,* vol. 22, nº ISSN 2352-3409, pp. 41-49, 2019.

[2] World Health Organization, «WHO Coronavirus Disease (COVID-19) Dashboard,» 9 Septiembre 2020. [En línea]. Available: https://covid19.who.int/.

[3] Instituto Nacional de Estadística, «España en cifras. Año 2020,» 30 Enero 2020. [En línea]. Available: https://www.ine.es/ss/Satellite?L=es_ES&c=INEPublicacion_C&cid=1259924856416&p=1254735110672&pagename=ProductosYServicios%2FPYSLayout&param1=PYSDetalleGratuitas. [Último acceso: 9 Septiembre 2020].

[4] Instituto Nacional de Estadística. Frontur. , «Estadística de Movimientos Turísticos en Fronteras Diciembre 2019 (FRONTUR),» 3 Febrero 2020. [En línea]. Available: https://www.ine.es/daco/daco42/frontur/frontur1219.pdf. [Último acceso: 9 Septiembre 2020].

[5] Instituto Nacional de Estadística. , «España en cifras. Año 2019. Página 50,» 30 Enero 2019. [En línea]. Available: https://www.ine.es/prodyser/espa_cifras/2019/50/. [Último acceso: 9 Septiembre 2020].

[6] TecnoHotelNews, «Inteligencia Artificial: ejemplos de cómo está cambiando a los hoteles,» tecnohotelnews.com, 10 Octubre 2018. [En línea]. Available: https://tecnohotelnews.com/2018/10/08/inteligencia-artificial-hoteles-ejemplos/. [Último acceso: 9 Septiembre 2020].

[7] A. Wilson, «Guide to Airbnb vs Hotels for 2019,» skyscanner.com, 9 Diciembre 2018. [En línea]. Available: https://www.skyscanner.com/tips-and-inspiration/hotels/airbnb-vs-hotels. [Último acceso: 9 Septiembre 2020].

[8] Hilton Hotels & Resorts, «LightStay – A Decade of Managing our Environmental and Social Impact,» newsroom.hilton.com, 26 Agosto 2019. [En línea]. Available: https://newsroom.hilton.com/brand-communications/news/lightstay-a-decade-of-managing-our-environmental-and-social-impact. [Último acceso: 9 Septiembre 2020].

[9] Hilton Hotels & Resorts, «Hilton Wins Product of the Year from

Environmental Leader for Corporate Responsibility Measurement Platform 'LightStay',» newsroom.hilton.com, 23 Junio 2016. [En línea]. Available: https://newsroom.hilton.com/corporate/news/hilton-wins-product-of-the-year-from-environmental-leader-for-corporate-responsibility-measurement-platform-lightstay. [Último acceso: 9 Septiembre 2020].

[10] Hospitality Tech Website, «Wynn Las Vegas to Add Amazon Echo to All Hotel Rooms,» hospitalitytech.com, 1 Abril 2017. [En línea]. Available: https://hospitalitytech.com/wynn-las-vegas-add-amazon-echo-all-hotel-rooms. [Último acceso: 9 Septiembre 2020].

[11] Springwise Website, «The Clarion Hotel in Stockholm is piloting a voice-activated chatbot that services the needs of hotel guests.,» springwise.com, 28 Septiembre 2016. [En línea]. Available: https://www.springwise.com/hotel-chatbot-serves-guests/. [Último acceso: 9 Septiembre 2020].

[12] D. Creamer, «Chatbot Increases Hotels' Room Service Sales 10-50% and Gets Wooed By Guests,» hospitalitytech.com, 13 Diciembre 2019. [En línea]. Available: https://hospitalitytech.com/chatbot-increases-hotels-room-service-sales-10-50-and-gets-wooed-guests. [Último acceso: 9 Septiembre 2020].

[13] Airbnb, «Airbnb and The Rise of Millenial Travel,» *Airbnb Citizen,* nº 8, p. 9, 2016.

[14] en.wikipedia.org, «Average daily rate,» Wikipedia, 23 Agosto 2017. [En línea]. Available: https://en.wikipedia.org/wiki/Average_daily_rate. [Último acceso: 9 Septiembre 2020].

[15] es.wikipedia.org, «Aprendizaje basado en árboles de decisión,» Wikipedia, 29 Julio 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Aprendizaje_basado_en_%C3%A1rboles_de_decisi%C3%B3n. [Último acceso: 9 Septiembre 2020].

[16] en.wikipedia.org, «Outliers (Statistics),» Wikipedia, 7 Septiembre 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Outlier. [Último acceso: 8 Septiembre 2020].

[17] Python scikit-learn Community, «sklearn.neighbors.KNeighborsRegressor,» scikit-learn, 3 Agosto 2020. [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html. [Último acceso: 9 Septiembre 2020].

[18] en.wikipedia.org, «Tikhonov regularization,» Wikipedia, 23 Agosto 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Tikhonov_regularization. [Último acceso: 9 Septiembre 2020].

[19] en.wikipedia.org, «LASSO (estadística),» Wikipedia, 30 Octubre 2019. [En línea]. Available: https://es.wikipedia.org/wiki/LASSO_(estad%C3%ADstica). [Último acceso: 9 Septiembre 2020].

[20] es.wikipedia.org, «Hotel,» Wikipedia, 8 Septiembre 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Hotel. [Último acceso: 9 Septiembre 2020].

[21] Sistema de Calidad Turística Española, «Sistema de Calidad Turística Español (SCTE),» Gobierno de Canarias, Consejería de Turismo, Industria y Comercio, [En línea]. Available: http://www.gobiernodecanarias.org/turismo/dir_gral_ordenacion_promocion/calidad/scte/. [Último acceso: 9 Septiembre 2020].

[22] J. (. p. e. K. Mostipak, N. Antonio, A. de Almeida y L. Nunes, «Hotel booking demand,» Kaggle.com, 11 Febrero 2020. [En línea]. Available: https://www.kaggle.com/jessemostipak/hotel-booking-demand. [Último acceso: 8 Septiembre 2020].

[23] Airbnb, «Seattle Airbnb Open Data,» kaggle.com, 1 Noviembre 2016. [En línea]. Available: https://www.kaggle.com/airbnb/seattle. [Último acceso: 8 Septiembre 2020].

[24] Consejería de Turismo, Industria y Comercio, «Agencia de Calidad Turística de Canarias (ACTC),» Gobierno de Canarias, [En línea]. Available: http://www.gobiernodecanarias.org/turismo/dir_gral_ordenacion_promocion/calidad/ACTC/index.html. [Último acceso: 6 Septiembre 2020].

[25] Instituto Canario de Estadística, «Encuesta de Expectativas de la Actividad Hotelera. Sector servicios,» Gobierno de Canarias, [En línea]. Available: http://www.gobiernodecanarias.org/istac/estadisticas/sectorservicios/. [Último acceso: 6 Septiembre 2020].

[26] Escuela de formación Alcazarén, «Gestión hotelera y liderazgo, algo más que dirigir un alojamiento turístico,» Alcazarén, 30 Marzo 2020. [En línea]. Available: https://www.alcazarenformacion.es/gestion-hotelera-liderazgo-alojamiento-turistico/. [Último acceso: 5 Septiembre 2020].

[27] Y. Alarcón Soto, «Data science scheme based on the Conway's Venn diagram (Conway, 2010).,» 2019.

[28] K. Kadupitige, «Intersection of HPC and Machine Learning,» Digital Science Center, 2017.

[29] D. Bzdok, D. Altman y M. Krzywinski, «Statistics versus machine,» *Nature,* vol. 15, nº 4, pp. 233-234, 2018.

[30] en.wikipedia.org, «Domain knowledge,» Wikipedia, 26 Septiembre 2019. [En línea]. Available: https://en.wikipedia.org/wiki/Domain_knowledge. [Último acceso: 8 Septiembre 2020].

[31] en.wikipedia.org, «Artificial intelligence,» Wikipedia, 8 Septiembre 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Artificial_intelligence. [Último acceso: 9 Septiembre 2020].

[32] D. R. Gomis, «Uso de la IA en la medicina,» Universitat Oberta de Catalunya, 5 Septiembre 2019. [En línea]. Available: http://cienciasdelasalud.blogs.uoc.edu/inteligencia-artificial-en-medicina/. [Último acceso: 9 Septiembre 2020].

[33] es.wikipedia.org, «Ciencia de datos,» Wikipedia, 27 Junio 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Ciencia_de_datos. [Último acceso: 9 Septiembre 2020].

[34] Unidad de Datos de Telefónica, «¿Qué son los Datos Masivos o Big Data?,» Unidad de Datos de Telefónica, [En línea]. Available: https://luca-d3.com/es/data-speaks/diccionario-tecnologico/datos-masivos. [Último acceso: 9 Septiembre 2020].

[35] C. Juan, «¿Cuáles son las 5 V's del Big Data?,» IEBSchool, 3 Noviembre 2016. [En línea]. Available: https://www.iebschool.com/blog/5-vs-del-big-data/. [Último acceso: 10 Septiembre 2020].

[36] en.wikipedia.org, «Global Distribution System,» Wikipedia, 5 Mayo 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Global_distribution_system. [Último acceso: 10 Septiembre 2020].

[37] en.wikipedia.org, «Aprendizaje Automático,» Wikipedia, 8 Agosto 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico. [Último acceso: 10 Septiembre 2020].

[38] NetApp, «¿Qué es el aprendizaje automático?,» Corporación NetApp, [En línea]. Available: https://www.netapp.com/es/info/what-is-machine-learning-ml.aspx. [Último acceso: 11 Septiembre 2020].

[39] T. O. Ayodele, «Types of Machine Learning Algorithms,» Septiembre, 10,

2020.

[40] http://www.cognub.com, «Machine Learning Algorithm,» 2018.

[41] J. Brownlee, «Linear Regression Formula,» 2016.

[42] N. Antonio, «Nuno António,» ISCTE-IUL, [En línea]. Available: https://ciencia.iscte-iul.pt/authors/nuno-antonio/cv. [Último acceso: 8 Septiembre 2020].

[43] A. De Almeida, «Ana de Almeida,» ISCTE-IUL, [En línea]. Available: https://ciencia.iscte-iul.pt/authors/ana-de-almeida/cv. [Último acceso: 5 Septiembre 2020].

[44] L. Nunes, «Luis Nunes,» ISCTE-IUL, [En línea]. Available: https://ciencia.iscte-iul.pt/authors/luis-miguel-martins-nunes/cv. [Último acceso: 6 Septiembre 2020].

[45] N. Antonio, A. De Almeida y L. Nunes, «Esquema Sistema Gestión Reservas de los hoteles (PMS),» 2019.

[46] Google Collab, «Google Collab,» colab.research.google.com, [En línea]. Available: https://colab.research.google.com/notebooks/intro.ipynb#recent=true. [Último acceso: 13 Julio 2020].

[47] NumPy.org, «NumPy,» NumPy Community, 20 Junio 2020. [En línea]. Available: https://numpy.org/. [Último acceso: 11 Agosto 2020].

[48] SciPy.org, «SciPy,» SciPy Community, 10 Septiembre 2020. [En línea]. Available: https://www.scipy.org/. [Último acceso: 11 Septiembre 2020].

[49] Pandas Python Library, «pandas - Python Data Analysis Library,» pandas.pydata.org, 8 Septiembre 2020. [En línea]. Available: https://pandas.pydata.org/. [Último acceso: 10 Septiembre 2020].

[50] scikit-learn communiy, «scikit-learn Machine Learning for Python,» scikit-learn.org, 2020 Agosto 4. [En línea]. Available: https://scikit-learn.org/stable/. [Último acceso: 23 Agosto 2020].

[51] J. Hunter, D. Dale, M. Droettboom y E. Firing, «Matplotlib: Python Plotting,» matplotlib.org, 14 Agosto 2020. [En línea]. Available: https://matplotlib.org/contents.html. [Último acceso: 19 Agosto 2020].

[52] M. Waskom, «seaborn: statistical data visualization,» Seaborn Python org, 1 Septiembre 2020. [En línea]. Available: https://seaborn.pydata.org/. [Último acceso: 4 Septiembre 2020].

[53] Python organization, «Python Programming Language,» python.org, 20

Julio 2020. [En línea]. Available: https://www.python.org/. [Último acceso: 5 Septiembre 2020].

[54] UCLA: Statistical Consulting Group., «What is Dumy Coding,» University of California, Los Angeles, 22 Agosto 2016. [En línea]. Available: https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faqwhat-is-dummy-coding/. [Último acceso: 1 Septiembre 2020].

[55] scikit-learn documentation team, «R^2 score, the coefficient of determination,» 2016.

[56] Scikit-learn documentation team, «Mean squared error,» 2016.

[57] Á. Román, «Machine Learning Supervisado: Fundamentos de la Regresión Lineal,» 2019.

[58] A. Singh, «Distance Formula used in KNN-Regression. Vidhya Analytics,» 2018.

[59] A. Agarwal, «Polynomial Regression Formula,» 2018.

[60] B. Saptashwa, «Linear model with n features for output prediction,» 2018.

[61] en.wikipedia.org, «Correlación de Pearson,» Wikipedia, 22 Agosto 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient. [Último acceso: 1 Septiembre 2020].

[62] J. Avila Camacho, «Fórmula estandarización,» 2020.

[63] es.wikipedia.org, «Análisis de Componentes Principales,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales#:~:text=En%20estad%C3%ADstica%2C%20el%20an%C3%A1lisis%20de,%C2%ABcomponentes%C2%BB)%20no%20correlacionadas.. [Último acceso: 17 Julio 2020].

[64] Nicoguaro, «Ilustración Análisis Componente Principal,» 2016.

[65] R. P. Sheridan, W. Min Wang, A. Liaw, J. Ma y E. M. Gifford, «Extreme Gradient Boosting as a Method for Quantitative Structure–Activity Relationships,» ACS Publications, 2016.

[66] scikit-learn development community, «sklearn.model_selection.GridSearchCV,» scikit-learn Python library, [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearch

CV.html. [Último acceso: 10 Septiembre 2020].

[67] en.wikipedia.org, «Random Forest,» Wikipedia, 11 Agosto 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Random_forest. [Último acceso: 7 Septiembre 2020].

[68] en.wikipedia.org, «Decision Tree learning,» Wikipedia, [En línea]. Available: https://en.wikipedia.org/wiki/Decision_tree_learning. [Último acceso: 9 Septiembre 2020].

[69] XGBoost Documentation Team, «XGBoost Documentation,» XGBoost ML Library, [En línea]. Available: https://xgboost.readthedocs.io/en/latest/. [Último acceso: 9 Septiembre 2020].

[70] «Logistic Regression,» Wikipedia, [En línea]. Available: https://en.wikipedia.org/wiki/Logistic_regression. [Último acceso: 6 Septiembre 2020].

[71] en.wikipedia.org, «Naive Bayes Classifier,» Wikipedia, 29 Agosto 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier. [Último acceso: 8 Septiembre 2020].

[72] Á. Román, «Algoritmos Naive Bayes: Fundamentos e Implementación,» 2019.

[73] https://www.python.org/, «Python.org,» python.org, 20 Julio 2020. [En línea]. Available: https://www.python.org/. [Último acceso: 18 Agosto 2020].

[74] scikit-learn development community, «sklearn.model_selection.train_test_split,» scikit-learn organization, [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html. [Último acceso: 29 Agosto 2020].