

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



Modelo de Predicción de Demanda y Maximización de Ingresos para Hoteles

TESIS

QUE PARA OBTENER EL GRADO DE

Maestro en Ciencia de Datos

PRESENTA

Oscar Daniel Ezequiel Camarena Gómez

ASESOR

José David Lampón Ortega

MÉXICO, D.F.

2018

“Con fundamento en los artículos 21 y 27 de la Ley Federal de Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada **“Modelo de Predicción de Demanda y Maximización de Ingresos para Hoteles”**, otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la biblioteca Raúl Baillères Jr., autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una prestación”

Oscar Daniel Ezequiel Camarena Gómez

Fecha

Firma

Resumen

Abstract

Índice general

1. Introducción	1
2. Marco teórico	6
3. Metodología	40
4. Diseño y Desarrollo	43
5. Interpretación y Resultados	76
6. Conclusiones	90
A. Implementación de modelo de pronóstico de demanda utilizando un modelo lineal generalizado con liga Poisson	91
B. Implementación de modelo de pronóstico de demanda utilizando una regresión de Ridge	96
C. Implementación de modelo de pronóstico de demanda utilizando un modelo de análisis de series de tiempo SARIMA	101
D. Implementación de modelo de optimización de ingresos	106
References	110

Índice de figuras

2.1. Ejemplo de curva de pickup	14
2.2. Ejemplo de red neuronal	19
2.3. Comportamiento del multiplicador de precio basado en el tiempo	29
2.4. Comportamiento del multiplicador de precio basado en la capacidad	30
2.5. Comportamiento del multiplicador de precio basado en la longitud de estancia	31
2.6. Comportamiento del multiplicador de precio basado en el tamaño del grupo	32
2.7. Flujo de check-ins y check-outs para el día k.	37
4.1. Arquitectura del modelo de predicción de demanda y maximización de ingresos	44
4.2. Curva de pickup	49
4.3. % Ocupación 2017	51
4.4. Ocupación por día de la semana	52
4.5. Tarifa Promedio 2017	53
4.6. Precios de la competencia en el tiempo	54
4.7. Precios de la competencia en el tiempo	55
4.8. Validación de predicciones utilizando regresión de Ridge	63
4.9. Generación de predicciones utilizando regresión de Ridge	64

ÍNDICE DE FIGURAS

4.10. Serie de tiempo para la ocupación de la propiedad desde 2013 hasta 2018	66
4.11. Descomposición de la serie de tiempo de ocupación	67
4.12. Validación de las predicciones generadas por el modelo SARIMA	69
4.13. Predicciones generadas por el modelo SARIMA	70
4.14. Asignacion de inventario	71
5.1. Curva de pickup real vs pronosticada (1)	80
5.2. Curva de pickup real vs pronosticada (2)	81
5.3. Curva de pickup real vs pronosticada (3)	81
5.4. Curva de pickup real vs pronosticada (4)	82
5.5. Efecto de α sobre las predicciones generadas	84
5.6. Diagnóstico del modelo SARIMA(1, 0, 0)x(1, 0, 1, 7)	85

Índice de tablas

2.1. Modelos de pronóstico por categoría	16
4.1. Resumen de datos obtenidos del PMS	48
4.2. DataSet detalle de reservas	48
4.3. Datos de entrada para el modelo de pronóstico de demanda . . .	56
4.4. Pronóstico de cuartos noche ocupados (Regresión con liga Poisson)	57
4.5. Cuartos noche ocupados vs tiempo	59
4.6. Conjunto de datos de entrada para la regresión de Ridge	60
4.7. Variables adicionales de entrada para la regresión de Ridge . . .	61
4.8. Valores residuales cuadráticos para los distintos valores de α . . .	62
4.9. Pronóstico de cuartos noche ocupados (Regresión de Ridge) . . .	64
4.10. Pronóstico de cuartos noche ocupados (SARIMA)	70
4.11. Matriz de asignacion de precio por inventario disponible	74
5.1. Resultados arrojados por el modelo de predicción de demanda . .	77
5.2. Interpretación de resultados arrojados por el modelo de pronóstico de demanda	79
5.3. Matriz de asignacion de precio por inventario disponible	87

Capítulo 1

Introducción

Una práctica común dentro de la industria de la hotelería es la maximización del ingreso, mejor conocida como *revenue management*. Esta surge de la necesidad de poder obtener vender cada uno de los cuartos dentro del inventario al mayor precio posible y de esta manera poder maximizar el ingreso percibido por la propiedad derivado de la venta de cuartos (El-Gayar N, et al. 2008). La técnica de *revenue management* tiene como base el análisis de información generada por la misma propiedad así como información ajena a ella para poder vender el producto adecuado, al cliente adecuado, en el momento y precio adecuados. Estas estrategias ayudan al hotelero en su gestión diaria de diversas maneras. Le permite establecer una estrategia coherente de precios a todos los niveles, controlar la distribución multicanal, realizar previsiones de la demanda, conocer mejor las necesidades y patrones de reserva de los diversos segmentos de clientes y disponer de más información sobre el mercado y el entorno que lo envuelve. Los sistemas de *revenue management* aplican principios básicos de economía para configurar los precios y el control del inventario que se pretende vender, de hecho, hay tres categorías básicas de administración de demanda que son utilizadas dentro del *revenue management*:

- *Decisiones Estructurales*: Se decide qué formato de venta se va a utilizar; a qué segmento de mercado se va a dirigir la venta; cuáles serán las condiciones de la venta que se ofrecerán.

CAPÍTULO 1: INTRODUCCIÓN

- *Decisiones de Precio:* Cómo se asignará el precio para los elementos dentro del inventario; diferentes categorías de precios en el mismo inventario; cambios en los precios del inventario en el tiempo.
- *Decisiones de Cantidad de Inventario:* Se decide si se acepta o se rechaza una oferta de compra; cuanto inventario se destina a cada uno de los segmentos de mercado; cuando retirar un producto del mercado para venderlo posteriormente.

Tradicionalmente los hoteles utilizan los controles de decisión de cantidad de inventario como su estrategia por "default" (Aziz, et al. 2011), sin embargo la tecnología ha estado cambiando el mercado y la manera en que un hotel vende su inventario, ya que hoy en día los costos asociados al cambio de precio del inventario y el tiempo para distribuirlos ha bajado considerablemente gracias a la adopción de un ecosistema totalmente electrónico (Bitran, Caldentey 2003). Antes del auge del *revenue management* la aerolínea *British Airways* experimentaba aplicando promociones en distintos productos dentro de su inventario para generar demanda en los asientos que sin esta promoción hubieran volado vacíos. Posteriormente la dirección de la aerolínea *American Airlines* dió un paso más adelante y crearon una practica llamada *yield management*, la cual se enfoca principalmente en maximizar los ingresos de cada ruta vendida mediante un control del inventario basado en análisis de la información histórica disponible. *American Airlines* hizo fuertes inversiones en lo que ellos llamaron *yield management* y lograron tener modelos de pronóstico de demanda robustos, así como fuertes políticas de control de inventario y estrategias de sobreventa. Estas inversiones realizadas aunado al acto de desregularización de aerolíneas (1978) llevó a que *American Airlines* tuviera tarifas muy competitivas disponibles para el mercado, promoviendo una sana competencia entre las diferentes aerolíneas de la época. Luego del éxito de la implementación de las técnicas de *yield management* en las aerolíneas, otras industrias comenzaron a implementar este tipo de técnicas. Una de estas industrias fue la hotelera, la cuál cuenta con una problemática similar a la presentada en las aerolíneas: inventario perecedero, clientes reservando cuartos con tiempo de anticipación, competencia ofreciendo tarifas de bajo costo, y grandes esfuerzos para poder balancear la oferta y la demanda. Sin embargo, el problema presentado en la industria de hoteles es

CAPÍTULO 1: INTRODUCCIÓN

más complejo aún ya que se debe tener en cuenta que un cliente puede permanecer en la propiedad más de una noche, afectando la oferta del hotel en días posteriores a las llegadas a sus huéspedes. Las técnicas de *revenue management* / *yield management* han sido estudiadas y explotadas de manera exitosa en la industria de la aerolínea, sin embargo hay mucho trabajo por hacer aún en otras industrias, incluyendo la hotelera ya que en la medida en que esta técnica se afine los ingresos que este sector perciba por concepto de renta de habitaciones puede crecer sustancialmente. Los sistemas existentes de *revenue management* tienen dos componentes principales: un módulo que se encarga de pronósticar la ocupación del hotel en días futuros y un módulo que toma este pronóstico como base para recomendar precios para el inventario de tal forma que se maximice el ingreso total de la propiedad. Existen dos técnicas que son comúnmente utilizadas para hacer pronósticos de ocupación:

- **Análisis de indicadores históricos:** Se construyen modelos de series de tiempo con los principales indicadores del hotel (habitaciones ocupadas, reservaciones recibidas, etc) que ayudan a entender la temporalidad del hotel y el comportamiento de la demanda de esa propiedad en días específicos. Este tipo de análisis suele tener errores muy altos Weatherford (1998).
- **Regresiones generalizadas, y modelos de *machine learning*:** En los casos en dónde se cuenta con información desagregada del hotel (información de cada una de las reservaciones) es posible aplicar modelos basados en regresiones generalizadas o redes neuronales, en las cuales se puede modelar el comportamiento de la demanda del hotel utilizando toda la información disponible de la propiedad reduciendo los errores de los pronósticos arrojados por el modelo Caicedo-Torres, Payares (2016).

Es importante mencionar que una mejora del 10 % en la exactitud del pronóstico de la ocupación puede llevar a un incremento de entre el 0.5 % y 3.0 % en el ingreso del hotel (Weatherford, et al. 2003), es por ello que resulta de gran interés la mejora continua de los modelos de pronóstico de ocupación.

Objetivo

El objetivo de el presente trabajo es presentar una propuesta de modelo de pronóstico de ocupación y maximización de ingresos para hoteles, mismo que pudiera ser encapsulado dentro de un sistema de *revenue management*. Se analizó el comportamiento de tres modelos de pronóstico diferentes. El primero está basado en una regresión lineal generalizada con liga Poisson (enfoque tradicional); el segundo se basa en un regresión de Ridge (enfoque de *Machine Learning* y el tercero se trata de un modelo *ARIMA* (enfoque de análisis de series de tiempo). El modelo con mejor desempeño alimenta al módulo de optimización de ingresos y recomendación de tarifas.

Durante esta investigación se tuvo acceso a la base de datos de reservaciones para un hotel de negocios ubicado al sur de la Ciudad de México con información a partir del 01 de enero de 2013 hasta el 14 de agosto del 2018. También se utilizó una matriz de la calidad vs el precio de la propiedad y su set competitivo (definido como el conjunto de hoteles dentro de un radio de 5 km a la redonda y que compiten por la demanda del mismo mercado objetivo) y un calendario de eventos de la plaza en la cuál se encuentra ubicado el hotel.

La primera etapa de la investigación consistió en realizar un análisis exploratorio de los datos, donde se detectó el nivel de limpieza y el comportamiento de las distintas variables de interés. Una vez concluido el análisis se generaron tres modelos de pronóstico de ocupación y se midió el desempeño de cada uno de ellos. El modelo con el mejor desempeño se encarga de entregar una matriz de pronóstico de ocupación vs fecha objetivo que sirve como datos de entrada para el modelo de maximización de ingresos y recomendacion de tarifa, en el cuál resuelve un problema de optimización con restricciones utilizando el método *SLSQP* teniendo como función objetivo:

$$\sum_{i=0}^n p_i * o_i$$

En donde i es la índice de la noche, p_i es el precio del cuarto para la i -ésima noche y o_i es la ocupación pronosticada (demanda) para la i -ésima noche al precio p_i

En este trabajo de investigación se presentarán los resultados de cada uno de

CAPÍTULO 1: INTRODUCCIÓN

los modelos y los detalles sobre las limitantes así como los siguientes pasos para poder mejorar el desempeño de cada uno de ellos.

Justificación

Si bien los modelos de pronóstico de ocupación y los sistemas de *revenue management* han madurado a lo largo de más de 30 años de haber sido implementados por primera vez en distintas industrias, aún hay mucho trabajo por hacer ya que el poder de cómputo y la capacidad de procesamiento ha aumentado significativamente a lo largo del tiempo abriendo nuevas oportunidades y facilitando la implementación de nuevas técnicas para el manejo de datos y el modelado estadístico que puede ser de gran ayuda durante la implementación de este tipo de modelos.

El mercado hotelero es altamente dinámico, genera inmensas cantidades de información día con día, misma que es almacenada en servidores y pocas veces es utilizada para poder obtener una ventaja competitiva. De ser exitoso el modelo aquí implementado se podría aprovechar la investigación realizada para generar un marco de trabajo que ayude a este sector en particular a aprovechar la información generada detrás de sus datos, de tal suerte que puedan obtener mayores ingresos por los servicios ofrecidos generando un ambiente de competencia sana entre los distintos jugadores de este mercado.

Capítulo 2

Marco teórico

En este capítulo se introducirá y discutirá el trabajo previo realizado y los conceptos necesarios para comprender las distintas técnicas y enfoques utilizados para la construcción de un modelo de predicción de demanda y maximización de ingresos así como el impacto que los resultados tienen dentro de la industria hotelera. Se dará una breve explicación del proceso de gestión de una propiedad mencionando cuales son los principales indicadores de control del proceso, así como las variables estudiadas para la toma de decisiones.

Gestion de una propiedad

El proceso de gestión de propiedades o de hoteles comprende una serie de actividades que tienen como objetivo el garantizar la rentabilidad de un proyecto de arrendamiento. Dentro de las actividades que se llevan a cabo están las siguientes:

- Gestión de *canales de venta*
- Gestión de *segmentos de mercado*
- Gestión de *precios por tarifa*
- Análisis del *comportamiento de la plaza*

CAPÍTULO 2: MARCO TEÓRICO

- Análisis de *indicadores principales*

A continuación profundizaremos en cada una de las actividades que forman parte de este proceso.

Gestión de canales de venta

El equipo administrativo del hotel o propiedad, deben analizar diariamente el comportamiento de sus canales de venta o distribución. Típicamente un hotel tiene un catálogo estándar de canales de distribución:

- Sitio Web
- Call Center / FrontDesk
- Agencias de viajes en línea
- Globalizadores Agencias de viajes

Cada uno de los canales de venta van dirigidos a un segmento de mercado en específico y lo que el equipo de administración debe hacer es asegurar que la demanda del hotel se mantenga lo suficientemente alta para poder asegurar la continuidad operativa del negocio. Para lograr esto se genera una estrategia de distribución, en la cuál se designa parte del inventario a cada uno de los canales y se fija un precio a cada una de las tarifas a ofertar. Los precios para un producto en específico tienden a variar entre los diferentes canales y esto tiene que ver con el costo asociado a la operación de cada uno de los canales. Por ejemplo, el sitio web típicamente es un canal de reservación propio del hotel, lo que significa que las ventas generadas por este canal son libres de comisiones; por el contrario, las agencias de viajes en línea tienden a cobrar una comisión por la venta generada por esos canales.

Lo que la administración del hotel debe conseguir es poder tener la mezcla de ventas entre sus canales que maximice el ingreso, de tal forma que los costos generados por las ventas no sean mayores a un porcentaje de los ingresos generados por la propiedad, generalmente este porcentaje es definido por la gerencia del hotel o por el equipo comercial que administra la cadena de hoteles. Es con esta restricción donde se descartaría una asignación total del inventario a

CAPÍTULO 2: MARCO TEÓRICO

las agencias de viajes en línea, ya que la comisión de este canal puede ser de hasta el 25% del total de la venta, lo cual podría representar una pérdida de hasta $\frac{1}{4}$ de las ganancias que se pudieron haber obtenido. Si por el contrario asignáramos todo el inventario disponible a los canales propios del hotel, si bien no se paga una comisión, la demanda generada por estos canales es inferior a la demanda generada por las agencias de viajes en línea.

Gestión de segmentos de mercado

La gestión de los segmentos de mercado es similar a la gestión de los canales de venta ya que estos están relacionados a los canales de distribución.

Las propiedades típicamente distinguen y segmentan a sus clientes en las siguientes categorías:

- *Clientes directos*: Aquellos clientes que reservan en algún canal directo del hotel
- *Clientes de negocios*: Aquellos clientes que pertenecen a una empresa que tiene alguna tarifa convenida con el hotel. Típicamente esta tarifa está sujeta a una producción de cuartos durante un periodo
- *Clientes de mayoreo*: Aquellos clientes que reservan utilizando un canal mayorista; puede ser una agencia de viajes en línea o mediante un agente de viajes.
- *Grupos*: Aquellos clientes que llegan a la propiedad como parte de un grupo.

Para gestionar eficientemente los segmentos de mercado se debe crear una estrategia de asignación de precios a las diferentes tarifas que van dirigidas a cada uno de ellos. Por ejemplo, un *cliente directo* que llega al hotel sin reservación, es propenso a pagar un precio más alto por una habitación disponible, al contrario de un cliente que viene como parte de un grupo que reservó con mayor tiempo de anticipación con una tarifa mucho más baja. El equipo administrativo debe cuidar los segmentos de mercado que ocupan la propiedad ya que el tener un grupo muy grande ocupándola significa muchas veces tener un hotel lleno a una

CAPÍTULO 2: MARCO TEÓRICO

tarifa muy baja y con mayor desgaste en las habitaciones, y por el otro lado, no siempre es posible llenar un hotel con clientes que llegan sin reservación los cuales están dispuestos a comprar una habitación a un precio mas elevado.

Gestión de precios por tarifa

Una de las actividades mas importantes del proceso de gestión de una propiedad es la gestión de los precios por tarifa ya que generalmente las tarifas disponibles en un hotel están asociadas a un segmento de mercado y a un canal de reservación. Para poder realizar esta actividad se debe establecer una estrategia en la cual se define si se quiere incrementar la ocupación de la propiedad o si se quiere incrementar el *RevPAR*(revenue per available room). Una vez establecida la estrategia inicial se deben modificar los precios para las tarifas que impactan los canales y segmentos donde se quiere fomentar la demanda o incrementar el precio. Si el objetivo es incrementar la ocupación de la propiedad, lo mas probable es que se decida bajar los precios de las tarifas o bien, crear promociones sujetas a diversas restricciones. En cambio, si el objetivo es incrementar el *RevPAR*, muy probablemente subirán los precios en algunas tarifas y se cerrará disponibilidad en algunas tarifas en donde este incremento no se pueda llevar a cabo por temas contractuales, por ejemplo las tarifas convenidas con otras empresas o clientes.

Es importante mencionar que la ejecución de esta tarea dentro del proceso de gestión depende del análisis continuo de la información histórica generada por el hotel (comportamiento de la propiedad en años anteriores en la misma fecha sujeta al estudio), conocimiento de la plaza en donde se ubica la propiedad (depende de la experiencia del equipo administrativo del hotel), información ajena al hotel (situación marco económica, eventos cercanos a la plaza, desempeño de la competencia cercana a la propiedad, etc). Muchas veces el conocer esta información implica un gran esfuerzo por parte del equipo ya que la información tiende a estar distribuída en diversos sistemas o en ocasiones deberán recabarla físicamente (visitando a su competencia o haciendo llamadas telefónicas) dejando un riesgo latente de recibir información que no es precisa.

Otra de las fuertes dependencias para la ejecución efectiva de esta tarea es contar con sistemas que distribuyan los nuevos precios en todos los canales disponibles

CAPÍTULO 2: MARCO TEÓRICO

en poco tiempo y con poco esfuerzo ya que de lo contrario no tendría sentido realizar esta labor si al final los precios no pueden ser cambiados dentro de un tiempo en el que el mercado pueda responder de acuerdo a lo planeado durante el análisis de la información. Una vez que los precios son alterados, el equipo deberá analizar nuevamente la información para asegurar que los cambios tuvieron los efectos deseados, de lo contrario, se deberá cambiar la estrategia lo antes posible para evitar afectar el desempeño de la propiedad.

Análisis del comportamiento de la plaza

En el sector de la hotelería en México, se le llama plaza a la ubicación geográfica en dónde se encuentra una propiedad y típicamente agrupa a propiedades independientes o de otras marcas (set competitivo), oficinas, corporativos, centros comerciales, y lugares de interés turístico. La plaza genera gran cantidad de información que es de interés para el equipo administrativo de una propiedad, ya que el desempeño financiero de esta depende mucho de la situación que presenta la plaza en un momento determinado.

Los hoteles que comparten una plaza generalmente están dispuestos a compartir información de su propiedad con otros, de esa forma pueden tener un conocimiento mas amplio sobre la realidad que se vive en ella. Si una propiedad requiere obtener datos de su competencia, basta con una simple llamada telefónica a la propiedad de interés para solicitar los datos de interés, por ejemplo: *cuartos ocupados*, *cuartos disponibles*, *cuartos fuera de servicio*, *ingresos generados por la venta de habitaciones*. Con esta información se pueden calcular algunos indicadores que son indispensables para el análisis del comportamiento de la plaza:

- % de Ocupación: $\frac{\text{cuartos ocupados}}{\text{cuartos disponibles}}$
- ADR (Average Daily Rate o Tarifa Promedio): $\frac{\text{ingresos}}{\text{cuartos ocupados}}$
- RevPAR (Revenue per available room o Tarifa Efectiva): $\frac{\text{ingresos}}{\text{cuartos disponibles}}$

Para poder conocer la situación de una propiedad frente a la competencia en una plaza se generan otros indicadores que facilitan este tipo de análisis.

- Penetración de ocupación: $\frac{\% \text{ ocupacion}}{\% \text{ ocupacion plaza}}$

CAPÍTULO 2: MARCO TEÓRICO

- Penetración de tarifa: $\frac{\text{tarifa promedio}}{\text{tarifa promedio plaza}}$

El hotel o propiedad que cuente con los valores mas altos para estos indicadores será el hotel con mayor demanda y mayores ingresos dentro de la plaza, sin embargo, para que el análisis sea efectivo se debe escoger al set competitivo de manera adecuada, de tal forma que los hoteles contenidos dentro de este set sean de la misma categoría (para el caso de la cadena estudiada deberán ser hoteles 3 o 4 estrellas de servicios limitados) y vayan dirigidos al mismo público objetivo.

Análisis de principales indicadores

Como se mencionó anteriormente, existen indicadores de desempeño de una propiedad u hotel que deben ser estudiados de manera diaria para poder tomar decisiones que mejoren el desempeño de esta, asegurando una continuidad operativa y la rentabilidad del inmueble:

- % ocupación
- ADR
- REvpar

Típicamente durante el análisis realizado, se compara el desempeño de cada uno de estos indicadores de manera diaria, mensual y anual, comparando los valores contra el año anterior y buscando siempre una variación positiva entre años. En caso de tener variaciones negativas se debe buscar la causa de esta variación. Se puede tratar de una baja en la demanda general de la plaza, un periodo de recesión financiera generalizada, un nuevo competidor en la plaza, etc. Una vez identificado la causa se deben tomar las acciones necesarias para generar demanda a la propiedad en cuestión. Típicamente la forma más efectiva de hacerlo es modificar los precios de tal forma que el mercado responda favorablemente sin afectar la rentabilidad del inmueble.

Antecedentes: Modelos de Revenue Management

Podemos definir la práctica de *revenue management* como la aplicación de técnicas analíticas que intentan predecir el comportamiento de un consumidor al nivel de un micro mercado, optimizando la disponibilidad de productos y sus precios de tal forma que maximicen el crecimiento de los ingresos. El principal objetivo de esta práctica es vender el producto indicado al cliente indicado al tiempo indicado en el precio indicado. La esencia de esta práctica radica en entender la percepción del producto que tienen los clientes y alinear los precios de estos productos de acuerdo a esta haciendo una correcta segmentación de los clientes objetivos.

Es importante mencionar que la disciplina de *revenue management* se compone de dos partes principales:

1. Pronóstico de ocupación
2. Asignación de precios

En las siguientes secciones discutiremos acerca de los trabajos previos sobre los dos principales componentes de esta disciplina.

Trabajo previo: modelos de pronóstico de ocupación

La base de una práctica efectiva de *revenue management* es un buen modelo de pronóstico de demanda, ya que el resultado de este será la entrada del modelo de optimización de precio de tal suerte que mientras menor sea el error en el pronóstico de la demanda, mejor será la optimización de los precios para cada uno de los productos sujetos al análisis.

Se tiene registro de trabajo previo para el desarrollo de modelos de pronóstico de demanda en diferentes industrias, algunas de ellas son: *líneas aéreas*, *retail*, *telecomunicaciones* y *hospitalidad*, siendo la primera la que cuenta con modelos de pronóstico más maduros, este hecho obedece a que en esta industria es en dónde a principios de los años 80's nace la disciplina de *revenue management*.

La venta de asientos en un vuelo es un caso dónde naturalmente se puede utilizar un modelo de optimización de ingresos ya que cuenta con ciertas características

CAPÍTULO 2: MARCO TEÓRICO

que hacen que el modelado del problema sea muy intuitivo:

- Se cuenta con un inventario reducido
- El inventario es perecedero (una vez que el avión despegue, los asientos vacíos no generan un ingreso)
- Se cuenta con un amplio segmento de mercado que genera demanda sobre el producto
- Los asientos se compran con distintos tiempos de anticipación (hay clientes que compran con un mayor tiempo de anticipación que otros)
- Se conoce el tiempo de estancia de cada uno de los clientes en la aeronave.

Las condiciones mencionadas anteriormente facilitan la construcción de un modelo de pronóstico de demanda. A continuación haremos una breve descripción de la evolución de este tipo de modelos a lo largo del tiempo.

Beckmann y Bobkowski (1958) presentaron modelos estadísticos que describen reservaciones de pasajeros, cancelación de reservaciones y *no shows*. En su trabajo los autores comparan modelos utilizando diferentes distribuciones probabilísticas: *Poisson*, *Binomial Negativa* y *Gamma* aplicadas en datos generados por aerolíneas con la finalidad de modelar las reservaciones efectivas y propone una condición óptima para un nivel de sobreventa para un vuelo en particular.

Posteriormente, otros autores continuaron explorando nuevas metodologías para poder construir modelos de predicción de demanda. Uno de los autores más citados en las investigaciones de modelos de pronóstico de demanda es Lee (1990) que presenta un artículo en donde explica dos enfoques para resolver el problema de pronóstico de ocupación en aerolíneas:

- Modelos de series de tiempo
- Modelos basados en información de reservaciones

Los modelos basados en series de tiempo consideran únicamente las series de tiempo con la información de llegadas o porcentajes de ocupación aplicando modelos de series de tiempo (como suavizamiento exponencial, ARIMA, etc). En estos casos no se utiliza la información propia de las reservaciones.

CAPÍTULO 2: MARCO TEÓRICO

Para entender el enfoque basado en reservaciones se debe comprender el concepto de *pick-up* de un hotel. El *pick-up* está definido como el aumento o disminución de habitaciones reservadas entre dos fechas concretas y para un periodo determinado. Por ejemplo, al 21 de abril el número de habitaciones reservadas para el 1 de mayo es de 3 habitaciones y al 26 de abril las habitaciones reservadas para la misma fecha son ya 13. Por lo tanto, el *pick-up* para el 1 de mayo entre el 21 de abril al 26 de abril es de 10 habitaciones.

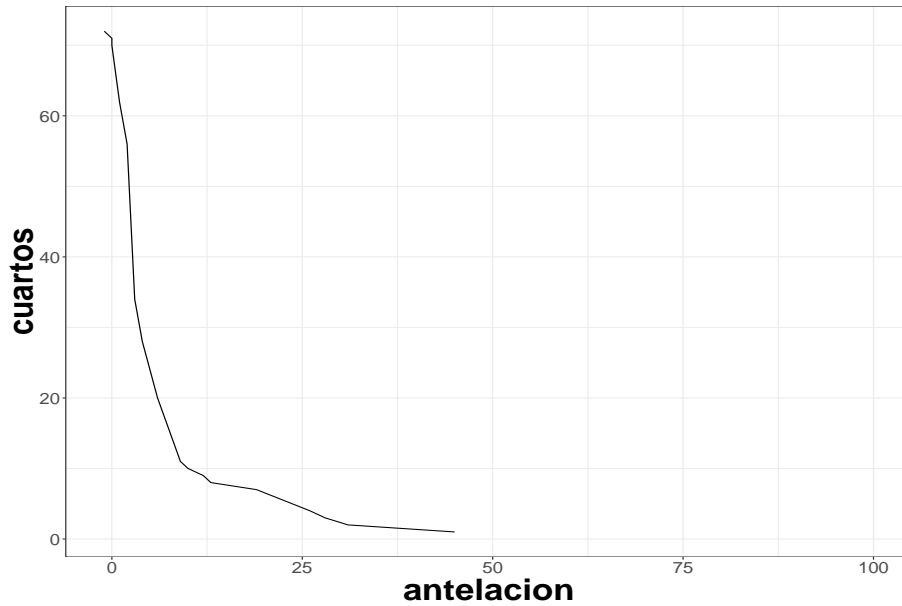


Figura 2.1: Ejemplo de curva de pickup

El enfoque basado en reservaciones hace uso de la información contenida en las reservaciones para pronosticar llegadas futuras. En este tipo de modelos típicamente se considera el *pick-up*. Esto quiere decir que dadas K reservaciones para un día futuro T , esperamos tener un *pick-up* de N reservaciones más desde este momento hasta el momento T . El pronóstico entonces será $K + N$.

Existen dos versiones del modelo de *pick-up* (Weatherford y Kimes, 2003). La versión aditiva del modelo en donde a las reservaciones registradas en el hotel se le suman el promedio de reservaciones que típicamente llegan en un periodo entre la fecha actual y la fecha de llegada del huésped (tomando la temporalidad del

CAPÍTULO 2: MARCO TEÓRICO

hotel en cuenta), y la versión multiplicativa, la cual es muy similar, con la única diferencia de únicamente se toma en cuenta una fracción de las reservaciones actualmente registradas en el hotel.

Existen trabajos en los cuales se presenta un modelo avanzado en el cual se combinan modelos de series de tiempo y modelos basados en reservaciones, tal es el caso del artículo presentado por **Flides y Ord** (2002) en el cuál se concluye que la combinación de ambos modelos generalmente tienen un mejor desempeño en cuanto al pronóstico de la demanda. Otro trabajo que presenta un enfoque similar es el realizado por **Sa** (1987) en el cuál se utiliza una regresión multiple para desarrollar un modelo de pronóstico combinado, en este caso, la variable dependiente eran las reservaciones pendientes por llegar mientras la variable independiente incluía el número de reservaciones confirmadas, un índice de temporalidad, índice semanal y un promedio del comportamiento histórico de las reservaciones pendientes. Esta regresión se corría para varios días antes de la llegada de los clientes ($t = 7, 14, 21, \text{ y } 28$).

Es importante hacer las siguientes observaciones sobre los modelos comentados anteriormente:

- Los modelos de regesión lineal asumen que existe una correlación entre el número de reservaciones confirmadas y el número final de reservaciones (en el día 0), de tal forma que: $\text{pronostico}_{dia0} = a + b * \text{reservaciones}_{diaN}$
- Los modelos de regresión logarítmica asumen que existe una correlación entre el número de reservaciones confirmadas y el número final de reservaciones (en el día 0), de tal forma que: $\log(\text{pronostico}_{dia0}) = a + b * \log(\text{reservaciones}_{diaN})$
- El modelo aditivo suma las reservaciones actuales al promedio histórico del *pickup* de reservas desde el día de la lectura *diaN* hasta el día de la estancia *dia0*: $\text{pronostico}_{dia0} = \text{reservas}_{diaN} + \sum_{i=0}^n \frac{\text{pickup}_i}{n}$
- El modelo multiplicativo multiplica las reservaciones actuales por la proporción del promedio histórico de reservaciones desde el *diaN* hasta el día de la estancia *dia0*: $\text{pronostico}_{dia0} = \text{reservas}_{diaN} * \frac{\text{reservas}_n - \sum_{i=0}^n \frac{\text{pickup}_i}{n}}{\text{reservas}_n}$

La tabla 2.1 resume los modelos más comunes utilizados para pronosticar ocu-

CAPÍTULO 2: MARCO TEÓRICO

Categoría	Modelo
Historico	1. Mismo día año anterior 2. Promedios móviles 3. Suavizamiento exponencial 4. ARIMA
Avanzado	1. Aditivo a. Pickup Clasico b. Pickup Avanzado 2. Multiplicativo a. Curva de reservaciones
Combinados	1. Promedios ponderados de modelo histórico y modelo avanzado 2. Regresión 3. Modelo de información completa

Tabla 2.1: Modelos de pronóstico por categoría

pación, agrupados por categoría.

Desempeño de modelos de pronóstico

Se han realizado estudios con el objetivo de comparar el desempeño de los modelos previamente presentados, a continuación presentaremos los resultados obtenidos.

Wickham (1995) estudió la efectividad de una variedad de modelos de pronóstico de ocupación utilizando datos de aerolíneas. Durante este trabajo se comparan modelos históricos de ocupación (utilizando promedios simples y promedios ponderados) y modelos de *pickup* (clasicos y avanzados) llegando a la conclusión de que los segundos son más efectivos, al menos para los datos aplicados durante su análisis.

Weahterford (1998) comparó los modelos aditivos frente a los modelos multiplicativos y modelos de regresión dentro del contexto de las aerolíneas y encontró que los modelos aditivos y de regresión tienen un mejor desempeño que los multiplicativos.

Modelos de machine learning

Si bien las técnicas estadísticas anteriormente mencionadas pueden predecir efectivamente niveles de ocupación y demanda, se requieren de amplios conocimientos en estadística y de largos procedimientos para poderlas aplicar de manera tal que funcionen correctamente o de otra manera se puede optar por el uso de *software* comercial el cuál resulta muy costoso la mayoría de las veces. Para solventar estos hechos, algunos autores presentan un enfoque diferente para construir modelos de predicción de ocupación, tal es el caso de Caicedo-Torres, Payares (2016) quienes proponen el uso de algoritmos de *Machine Learning* para construir modelos predictivos de ocupación y demanda en la industria de la hospitalidad. La ventaja de estos modelos es que estan listos para ser utilizados por el staff administrativo de la propiedad sin la necesidad de contar con amplios conocimientos de estadística. Además, estos modelos tienen la ventaja de poder ser empaquetados dentro de aplicaciones de bajo costo, o de ser ejecutados en una infraestructura basada en la nube, lo cual significan soluciones rentables para el sector de hospitalidad.

A continuación presentaremos a detalle algunos algoritmos de M.L. utilizados para la construcción de este tipo de modelos.

Regresión de Ridge

La regresión de ridge es un algoritmo de regresión que incluye un término de regularización en su función de costos con el fin de lograr una mejor aproximación al momento de generalizar el modelo. La regresión de Ridge penaliza el algoritmo con un factor proporcional a la norma Euclidiana (L2) del vector de parámetros.

$$J(\theta) = \sum_{i=1}^n (\theta^T x_i - y_i)^2 + \alpha \|\theta\|^2$$

Dónde el factor de regularización α controla la importancia relativa de la penalidad compleja. Para optimizar esta ecuación se debe encontrar la derivada de la función de costos e igualarla a cero, de tal forma que el vector de parámetros que minimiza el error es:

$$\theta = (X^T X + \lambda I)^{-1} X^T y$$

CAPÍTULO 2: MARCO TEÓRICO

Las nuevas predicciones se pueden calcular como sigue:

$$h(X) = \theta^T x = \sum_{i=1}^n \theta_i x_i$$

Regresión de Ridge con Kernel

Este método consta de aplicar el truco del Kernel a la regresión previamente comentada, esto con la finalidad de proyectar los datos originales en un espectro de variables sin tener que explícitamente calcular las transformaciones. Para esto, se comienza con el vector de pesos optimizados para la Regresión de Ridge:

$$\theta = (X^T X + \lambda I)^{-1} X^T y$$

Se manipula algebraicamente para obtener:

$$\theta = X^T (X X^T + \lambda I)^{-1} y$$

De tal forma que una predicción puede ser obtenida por:

$$h(x') = \theta^T x' = y^T (X X^T + \lambda I)^{-1} X x'$$

Una función de Kernel evalúa a un producto punto de la forma

$$f(x_1, x_2) = \phi(x_1) \cdot \phi(x_2)$$

sin tener que explícitamente calcular la transformación ϕ . Si definimos $K_{i,j} = f(x_i, x_j)$ y $k_i = f(x_i, x')$, donde x_i es la i -ésima fila de la matriz X , entonces

$$h(x') = \theta^T x' = y^T (K + \lambda I)^{-1} k$$

Ajustamos un modelo en un espacio dimensional mayor inducido por ϕ sin tener que incurrir en una complejidad computacional adicional. Se pueden utilizar varios Kernels, entre los más populares están el Kernel polinomial

$$k(x_1, x_2) = (\gamma x_1^T x_2 + \beta)^\delta$$

Y el Kernel Gaussiano

$$k(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\delta^2}\right)$$

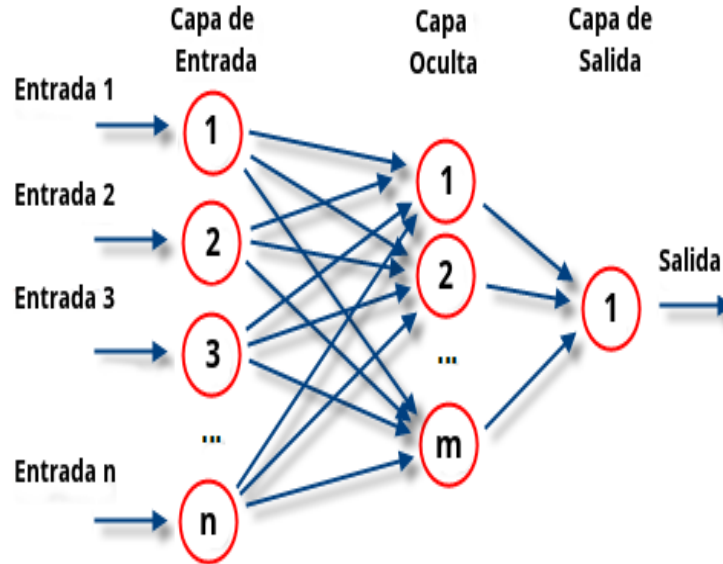


Figura 2.2: Ejemplo de red neuronal

Red neuronal multicapa

Una red neuronal es un modelo de M.L. capaz de clasificar y realizar regresión. Esta compuesta por n capas totalmente conectadas de neuronas artificiales formando un grafo conectado (en el caso más simple se tiene una capa oculta y una capa de salida). En este modelo, la salida de cada capa de neuronas será la entrada de la siguiente capa. A continuación se muestra una representación de una red neuronal multicapa.

Cada una de las neuronas dentro de la red contiene una función que define su estado, dicha función es llamada *función de activación*. En este caso, la función de activación es dada por la siguiente ecuación

$$y = f\left(\sum_i^n w_i x_i + b\right)$$

donde f es una función de activación. Entre las opciones más populares de funciones de activación se encuentran: la sigmoide $y = (1 + \exp(-x))^{-1}$ y la función lineal $y = x$. La elección de la función de activación en la capa de salida define el tipo de red neuronal (si clasifica o realiza regresión)

CAPÍTULO 2: MARCO TEÓRICO

Las redes neuronales son entrenadas mediante un algoritmo llamado *propagación hacia atrás* (*backpropagation*) que es una aplicación del algoritmo de descenso por gradiente que toma en cuenta la contribución de las neuronas de la capa oculta al error E .

Red neuronal de base radial

Las redes neuronales de base radial (RBF) usualmente tienen una sola capa oculta que emplea una serie de funciones llamadas *funciones de base radial* como detectores de características. Las redes RBF pueden ser utilizadas para realizar regresiones o clasificaciones. En este modelo la capa oculta utiliza función gaussiana como función de activación ϕ :

$$\phi_j(x) = \exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}\right)$$

Y como función de salida, se tiene la siguiente ecuación:

$$y(x) = \sum_{j=0}^M w_{kj} \phi_j(x)$$

donde w_{kj} es el vector de pesos que conecta la neurona j de la capa oculta con la neurona k de la capa de salida.

El entrenamiento de este tipo de modelos consiste en encontrar un conjunto de valores para los parámetros μ_j , σ_j y w_{jk} que maximizan el desempeño de la red.

Desempeño de algoritmos de M.L.

Generalmente este tipo de modelos son entrenados, validados y probados siguiendo un proceso llamado *validación cruzada*. Este proceso consiste en dividir en dos conjuntos complementarios los datos de muestra, para realizar el análisis de un subconjunto (denominado datos de entrenamiento o *training set*), y validar el análisis en el otro subconjunto (denominado datos de prueba o *test set*), de forma que la función de aproximación sólo se ajusta con el conjunto de datos de entrenamiento y a partir de aquí calcula los valores de salida para el conjunto de datos de prueba (valores que no ha analizado antes).

CAPÍTULO 2: MARCO TEÓRICO

La medida que permite evaluar el desempeño de este tipo de modelos es el *MAPE* (Mean Absolute Percentage Error) definida como:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - h_t}{y_t} \right|$$

Los resultados mostraron que la *regresión de Ridge* una transformación polinomial de grado dos tuvo el mejor desempeño con un *MAPE* del 8.20 % durante la validación y un *MAPE* del 8.65 % durante las pruebas.

Por otra parte el modelo la *regresión de Ridge con Kernel* tuvo un *MAPE* de 8.69 % durante la validación, seguido por la *red neuronal* con un *MAPE* del 12.89 % y finalmente la *red neuronal RBF* con un *MAPE* de 26.32 %.

Los resultados obtenidos durante esta investigación fueron bastante prometedores y sustentaron el uso de modelos de M.L. como modelos de caja negra para estimar la ocupación futura de una propiedad sin requerir conocimiento o experiencia estadística del estaff del hotel, permitiendo nuevos avances en las técnicas de *revenue management* en el sector de hospitalidad.

Modelos de Big Data

En el sector de turismo y hospitalidad se han hecho esfuerzos para obtener modelos de pronóstico de llegadas y ocupación basados en información generada a partir de las búsquedas en páginas web y el tráfico generado en ellas. Si bien, los resultados obtenidos han confirmado la validéz de algunas fuentes de datos generados en línea, poco trabajo se ha hecho cruzando diferentes fuentes de datos, por ejemplo, utilizar los datos de las búsquedas realizadas en *Google* combinados con el tráfico generado en el sitio web del hotel, los cuales deben estar altamente correlacionados (Tierney y Pan 2012).

Existen algunos trabajos en donde se adoptan dos métodos principales para obtener la ocupación de un hotel a partir de datos generados en línea. Los métodos utilizados son: *ARIMAX* y *MS DR*

Modelos ARIMA

Los modelos *ARIMA* (*autoregressive integrated moving average*) son modelos dinámicos de series temporales utilizados para hacer pronósticos basados en información histórica de la variable a predecir. En otras palabras, este tipo de modelos asumen que los valores futuros para la variable a predecir dependen linealmente de los valores pasados y no de variables independientes. Los modelos *ARIMAX* son una generalización de los modelos *ARIMA* en dónde se incluyen predictores independientes. El modelo *ARIMAX* es similar a un modelo de regresión multivariada pero permite tomar ventaja de la autocorrelación que puede estar presente en los residuos de la regresión para mejorar el pronóstico.

Los modelos *ARIMA* / *ARIMAX* están compuestos por dos partes principales: Uno o más términos autoregresivos (AR) y uno o más términos de promedios móviles (MA - moving averages). Los términos autoregresivos de la variable dependiente son valores anteriores para esa variable que tienen una relación significativa con el valor más reciente, mientras que los promedios móviles son residuos (errores pasados) de estimaciones previas. La *I* dentro del acrónimo *ARIMA* quiere decir que el modelo es *integrado* lo cual indica que las series de tiempo han sido diferenciadas una o más veces para hacerlas estacionarias antes de empezar el modelo.

Definición de Modelo ARMA / ARIMA / ARMAX

El modelo *ARIMA* se define como:

$$Y_t = -(\Delta^d Y_t - Y_t) + \phi_0 + \sum_{i=1}^p \phi_i \Delta^d Y_{t-i} - \sum_{i=1}^q \theta_i \epsilon_{t-i} + \epsilon_t$$

En donde:

- d corresponde a las d diferencias que son necesarias para convertir la serie original en estacionaria
- ϕ_1, \dots, ϕ_p son los parámetros pertenecientes a la parte *autorregresiva* del modelo
- $\theta_1, \dots, \theta_q$, los parámetros pertenecientes a la parte de *promedios móviles* del modelo

CAPÍTULO 2: MARCO TEÓRICO

- ϕ_0 es una constante
- ϵ_t es el término del error (llamado también innovación o perturbación estocástica)

Los modelos *ARIMA* / *ARMA* / *ARIMAX* son una opción común para realizar análisis de series de tiempo arrojando resultados satisfactorios, sin embargo, este tipo de modelos no son capaces de representar patrones dinámicos no lineales. Sería poco efectivo intentar analizar los datos de las tasas de crecimiento del *PIB* utilizando modelos *ARMA* ya que estos datos no presentan un simple patrón lineal en su comportamiento. Las tasas de crecimiento del *PIB* tienen distintos comportamientos dependiendo del momento económico del país. Durante una expansión económica la tasa de crecimiento fluctúa alrededor de valores más altos que durante los periodos de contracción, dónde fluctúa en niveles inferiores y menos persistentes. Para este tipo de casos se deben utilizar modelos no lineales de series de tiempo.

Modelo de cambios Markov

Uno de los modelos no lineales de series de tiempo más populares es el *Modelo de cambios Markov*. Este modelo involucra múltiples estructuras (ecuaciones) capaces de caracterizar los comportamientos de las series de tiempo en distintos regímenes permitiendo el cambio entre dichas estructuras, haciendo que el modelo sea capaz de capturar comportamientos de patrones dinámicos complejos. Una de las características de este modelo es que el mecanismo que permite el cambio entre las distintas estructuras definidas, es controlado por una variable de estado no observada que sigue una cadena de Markov de primer orden, dicha propiedad indica que el valor actual de la variable de estado depende de su valor inmediatamente anterior. Dadas estas propiedades, el modelo de cambios Markov es un candidato perfecto para describir datos que muestran distintos patrones dinámico en diferentes periodos de tiempo.

Implementación de modelos ARMAX / MSDR

En el estudio presentado por Pan y Yang (2017) se presentan dos modelos de series de tiempo en el que se incorporan tres fuentes de información generada

CAPÍTULO 2: MARCO TEÓRICO

en línea para predecir la ocupación de un hotel. Los datos de entrada contenían datos que describían:

- Tráfico generado en el sitio web de la propiedad
- Información relacionada con las peticiones realizadas a los motores de búsquedas (*Google*)
- Información del clima en el destino en dónde se ubica la propiedad

El primer modelo presentado es un modelo *ARMAX* en donde se utilizan variables generadas con *big data* como predictores directos de la ocupación. El modelo se definió de la siguiente manera:

$$y_t = \alpha + x_t\beta + \mu_t$$
$$\mu_t = \sum_{i=1}^m \rho_i \mu_{t-i} + \sum_{j=1}^n \theta_j \epsilon_{t-j} + \epsilon_t$$

donde y_t es la variable dependiente y x_t es un vector de variables exógenas independientes. Si β es igual a 0, el modelo se convierte en un modelo ARMA(m,n) estándar.

El segundo modelo presentado en este trabajo es el MSDR especificado como:

$$y_t = \tau_s + x_t\alpha + z_t\beta_s + \epsilon_s$$

dónde τ_s es el estado de intercepción; x_t es un vector de variables exógenas con coeficientes independientes del estado; z_t es un vector de variables exógenas con coeficientes dependientes del estado β_s ; ϵ_s es un error normal *i.i.d* con media 0 y una varianza dependiente del estado de σ^2 . Este modelo permite a las variables en el vector z_t con parámetros dependientes del estado tomar diferentes comportamientos basados en el estado que responden a un proceso de Markov con J estados y una probabilidad transicional p_{ij} especificada como sigue:

$$p_{ij} = \frac{1}{1 + \sum_{m=1}^k \exp(-q_{im})} \text{ si } j = k$$
$$p_{ij} = \frac{\exp(-q_{ij})}{1 + \sum_{m=1}^k \exp(-q_{im})} \text{ si } j \neq k$$

CAPÍTULO 2: MARCO TEÓRICO

donde q_{ij} es el parámetro transformado definido como

$$q_{ij} = -\log\left(\frac{p_{ij}}{p_{ik}}\right)$$

Para empezar con la estimación del modelo se utiliza el algoritmo EM *expectation-maximization* el cuál arroja el valor inicial para los parámetros del modelo. Este algoritmo es utilizado en modelos estadísticos para encontrar parámetros máximos verosímiles de variables latentes o que no han sido observadas. A continuación se explica brevemente el algoritmo iterativo de EM.

Dado un modelo estadístico que genera un conjunto de datos observados X , un conjunto de datos no observados, latentes o faltantes Z , un vector de parámetros desconocidos θ y una función de verosimilitud $L(\theta; X, Z) = p(X, Z|\theta)$, el estimador máximo verosímil de los parámetros desconocidos se determina maximizando la verosimilitud marginal de los datos observados:

$$L(\theta; X) = p(X|\theta) = \int p(X, Z|\theta) dZ$$

Sin embargo, esta cantidad es a menudo intratable. (Por ejemplo, si Z es una secuencia de eventos de tal forma que el número de valores crece exponencialmente con la longitud de la secuencia, haciendo extremadamente difícil obtener un cálculo exacto).

El algoritmo EM busca encontrar el estimador máximo verosímil de la verosimilitud marginal aplicando los siguientes dos pasos de manera iterativa:

1. *Esperanza - Paso E*: Define $Q(\theta|\theta^t)$ como el valor esperado del logaritmo de la función de verosimilitud de θ , con respecto a la distribución condicional de Z dado X y los parámetros actuales para θ^t , es decir:

$$Q(\theta|\theta^t) = E_{Z|X, \theta^t}[\log L(\theta; X, Z)]$$

2. *Maximización - Paso M*: Encontrar los parámetros que maximizan la siguiente ecuación:

$$\theta^{t+1} = \operatorname{argmax}_{\theta} Q(\theta|\theta^t)$$

Una vez encontrados los parámetros iniciales, se predicen las probabilidades de cada uno de los estados, actualizando la verosimilitud en cada paso. Un paso

CAPÍTULO 2: MARCO TEÓRICO

clave de esta estimación es el cálculo de la verosimilitud de los estados latentes la cual es obtenida mediante la iteración de la verosimilitud condicional.

Para medir el desempeño de ambos modelos se utilizó la medida *MAPE*

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - h_t}{y_t} \right|$$

Los resultados arrojan que el modelo ARMAX tuvo un mejor desempeño con un *MAPE* del 3.804 % comparado con el 4.714 % obtenido por el modelo MSDR

Se puede concluir que los modelos basados en big data presentan una alternativa para poder generar pronósticos de ocupación y demanda utilizando información exógena a los hoteles, sin embargo se debe tomar en cuenta que el costo asociado a la configuración y puesta en marcha de este tipo de modelos es alto ya que se debe de contar con una infraestructura que pueda soportar el manejo de grandes volúmenes de información.

Modelos de optimizacion de ingresos

Recordemos que un systema de gestión de ingresos para un hotel se puede clasificar dentro de dos grandes grupos (Abdel Aziz et al, 2011). En el primer grupo tendremos todos los sistemas que controlan la cantidad de inventario que se ofrece a cada segmento de cliente. Es decir, se asigna dinámicamente un porcentaje de inventario con un precio fijo distinto para cada uno de los segmentos estudiados por el hotel. El segundo grupo se distingue por asignarle un precio de manera dinámica al inventario que cuenta con características similares. El precio será ajustado continuamente en el tiempo basando los cambios en las variaciones de la oferta y la demanda del producto.

En esta sección se presentaran los modelos utilizados en trabajos anteriores para poder asignar precios dinámicos a un inventario con el fin de maximizar los ingresos de una propiedad.

Modelo de multiplicadores de precio

El modelo de multiplicadores de precio propone un enfoque sencillo para resolver el problema de maximización de ingresos de una propiedad.

CAPÍTULO 2: MARCO TEÓRICO

Este modelo propone tener un precio base de referencia (dependiendo de la temporalidad del hotel) y modificarlo utilizando un conjunto definido de multiplicadores (cuyo valor se encuentra alrededor del 1) y que son influenciados por variables sensibles a la operación de la propiedad, por ejemplo el % ocupación, el tiempo de antelación de una reserva, etc., de tal forma que cada multiplicador ajusta el precio por encima o por debajo del precio de referencia.

El objetivo de este modelo es maximizar los ingresos de la propiedad tomando en cuenta la demanda y la elasticidad del precio. Este modelo tiene la ventaja de enmarcar el precio en términos de *primas* o *descuentos* en un tiempo variable o sobre un precio de referencia temporal definido por el staff de la propiedad y al ser un modelo muy sencillo de entender permite que el gerente de la propiedad transmita un poco de su experiencia al modelo al ser el quien define el precio de referencia para cada una de las temporadas del hotel.

En trabajos previos se proponen las siguientes variables que influyen en el efecto en los precios:

- Tiempo de antelación de una reserva
- Habitaciones disponibles al momento de crear una reservación
- Tiempo de estancia de una reserva (LoS)
- Número de cuartos reservados (tamaño de grupo)

A partir de estas variables de influencia se construyen multiplicadores (1 multiplicador por cada variable). Para simplificar la formulación del problema, los multiplicadores usualmente son formulados como funciones lineales de las variables de influencia. Las funciones lineales son elegidas tomando en cuenta una relación lógica entre las variables de influencia y el comportamiento de la propiedad, por ejemplo, si al momento de hacer una reserva hay una alta disponibilidad de habitaciones entonces se esperaría que el multiplicador del precio lo haga bajar con la finalidad de incrementar la demanda de la propiedad.

El modelo recomendará un precio tomando en cuenta el precio de referencia dado por el gerente de la propiedad y los multiplicadores utilizando la siguiente fórmula:

CAPÍTULO 2: MARCO TEÓRICO

$$Precio_{final} = P_{Referencia} * M_{tiempo} * M_{capacidad} * M_{LoS} * M_{Grupo}$$

A continuación se describe a detalle el comportamiento de cada uno de los multiplicadores, esto con el fin de poder sentar las bases necesarias para poder fijar sus valores en un modelo operativo.

Multiplicador de precio basado en el tiempo

El tiempo que va de la fecha de reservación a la fecha de llegada es una variable muy importante dentro del cálculo del precio de la habitación. En la gráfica presentada a continuación se muestra el valor del multiplicador del precio (eje y) contra el tiempo que queda antes de la llegada del huésped al hotel (eje x). Al principio de la ventana de reservación, los precios deben ser bajos con la finalidad de agotar el inventario rápidamente, es por ello que el multiplicador empieza en un nivel bajo y_2^t . Esto puede ser considerado como un descuento *.early bird*". A medida que el tiempo avanza y la fecha de llegada es más cercana el descuento es cada vez menor, entonces el valor del multiplicador aumenta. El incremento en el valor del multiplicador continua hasta unos pocos días antes de la fecha de llegada (alcanzando su valor máximo en y_3^t). Es en este punto en donde los cuartos vacantes tienen baja probabilidad de ser vendidos en el tiempo restante, por lo que es prudente fijar un precio bajo para los cuartos vacantes con el fin de evitar que se queden sin ser ocupados. Es por ello que el valor del multiplicador disminuye hasta que alcanza su punto mínimo en y_1^t en la fecha de llegada.

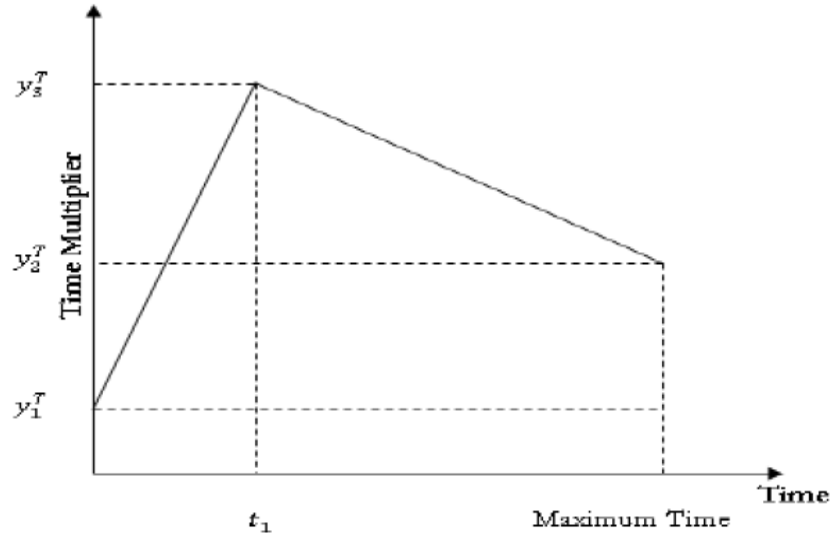


Figura 2.3: Comportamiento del multiplicador de precio basado en el tiempo

Multiplicador de precio basado en la capacidad

En lo que respecta al multiplicador del precio basado en la capacidad, se asume que una forma lineal con pendiente negativa, como se muestra en la figura a continuación.

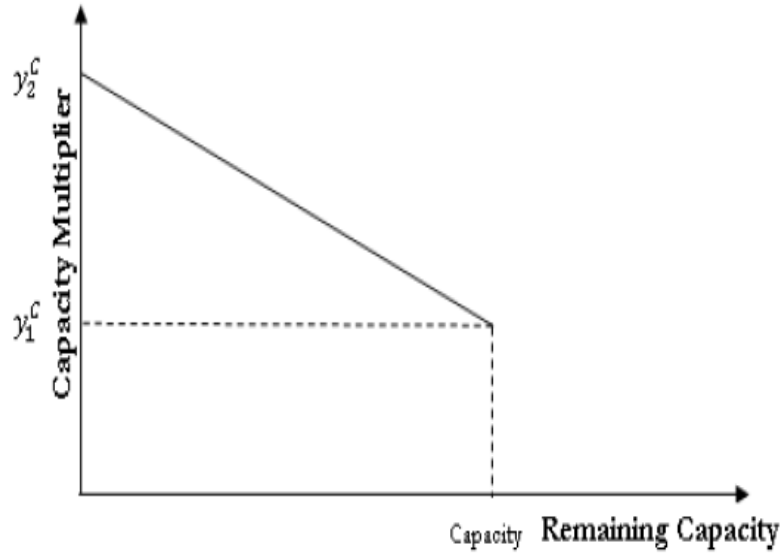


Figura 2.4: Comportamiento del multiplicador de precio basado en la capacidad

El eje x representa el número total de cuartos vacantes para la fecha objetivo, mientras que el eje y representa el valor del multiplicador. Si se cuentan con un alto número de cuartos vacantes se deberán ofrecer incentivos para venderlos, es por aquí donde el valor de este multiplicador alcanza su punto más bajo y_1^c . El valor del multiplicador deberá incrementar a medida que la capacidad restante del hotel decrece, hasta que alcanza su valor máximo y_2^c donde ya no hay más cuartos por vender.

Multiplicador de precio basado en la longitud de estancia

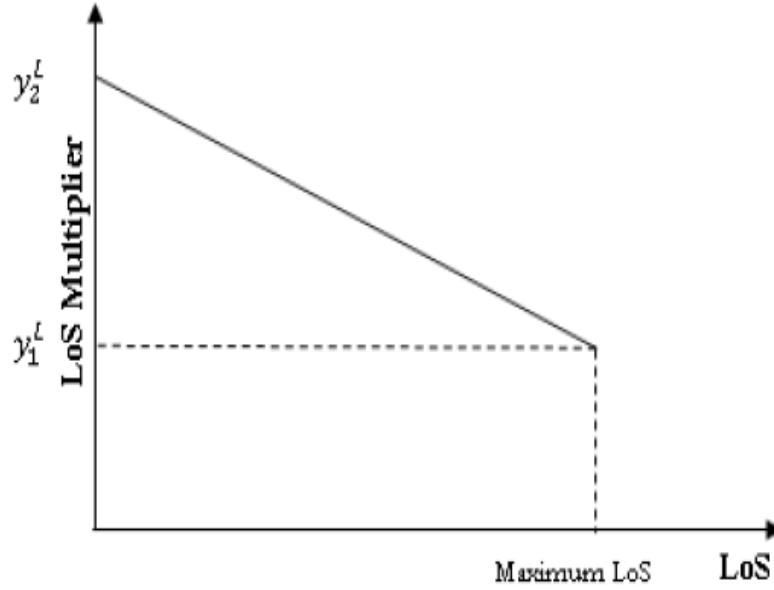


Figura 2.5: Comportamiento del multiplicador de precio basado en la longitud de estancia

EL precio debería ser monotónicamente decreciente conforme la longitud de estancia es mayor con la finalidad de incentivar estancias más largas (lo que garantiza un ingreso mayor). La figura presentada anteriormente muestra la relación sugerida para el multiplicador de precio basado en la longitud de estancia. Lo que se puede observar de la figura es que el valor del multiplicador varía linealmente desde un valor máximo γ_2^L para una longitud de estancia de 1 día, a un valor mínimo γ_1^L para el valor de la longitud de estancia mas larga típicamente encontrada en la propiedad.

Multiplicador de precio basado en el tamaño del grupo

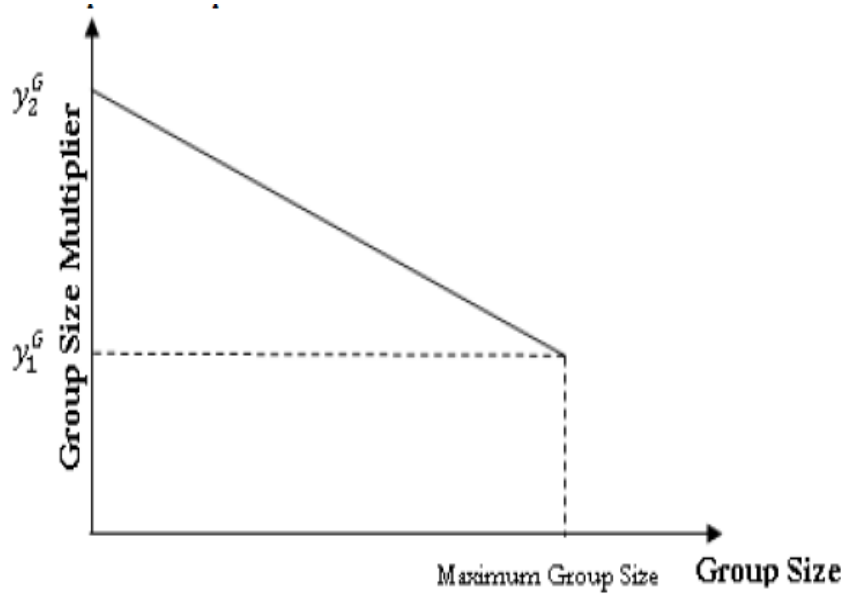


Figura 2.6: Comportamiento del multiplicador de precio basado en el tamaño del grupo

Se considera que la función que describe el comportamiento del multiplicador del precio basado en el tamaño del grupo varía linealmente desde un valor máximo y_2^G donde no existe un grupo, hasta su mínimo y_1^G para el tamaño máximo de un grupo típicamente encontrado en esa propiedad.

Variables de optimización y restricciones

Dentro de este modelo se asume que el valor promedio de cada una de las funciones de los multiplicadores de precio equivale a 1. Esto debido a que los multiplicadores son considerados factores de corrección que multiplicarán al precio de referencia. Es por este hecho que los valores de los multiplicadores deben ser asignados alrededor de 1, donde un valor mayor a 1 significa un aumento en el precio y un valor menor a 1 significa un descuento en el precio.

Las variables de optimización son las siguientes: *multiplicador basado en el tiem-*

CAPÍTULO 2: MARCO TEÓRICO

$po: y_1^T, y_2^T$ & t_1 ; *multiplicador basado en la capacidad*: y_2^C ; *multiplicador basado en la longitud de estancia*: y_2^L ; *multiplicador basado en el tamaño de grupo*: y_2^G . El algoritmo de optimización deberá fijar el valor de estas variables para poder maximizar el ingreso obtenido. Para poder modelar el proceso de optimización se deben tomar en cuenta las siguientes restricciones:

- Multiplicador basado en el tiempo: $0 \leq y_1^T \leq y_2^T \leq y_3^T$; $0 \leq t_1 \leq T_0$
- Multiplicador basado en la capacidad: $1 \leq y_2^C \leq C_0$
- Multiplicador basado en la longitud de estancia: $1 \leq y_2^L \leq L_0$
- Multiplicador basado en el tamaño del grupo: $1 \leq y_2^G \leq G_0$

Optimización

Para comenzar con la optimización, se deben tomar en cuenta las reservaciones presentes y pasadas generadas en la propiedad. Una vez analizada esta información se simularán reservaciones futuras. Se asume que los precios para las reservaciones generadas siguen la tendencia de los precios de referencia durante las diferentes temporalidades de la propiedad obtenidas de las reservaciones históricas. Posteriormente, el algoritmo de optimización prueba diferentes valores para los multiplicadores y fija el precio para los nuevos cuartos reservados. Hay que tomar en cuenta que cada cuarto reservado tiene un precio diferente debido a que cada cuarto tiene condiciones específicas como el tiempo de antelación con el que se hizo la reserva, la longitud de la estancia, la capacidad disponible del hotel, etc. Al sumar el ingreso generado por cada uno de los cuartos reservados se obtiene el conjunto de valores para cada uno de los multiplicadores de precio, de tal forma que el algoritmo continúa buscando valores hasta obtener el ingreso máximo.

Como se puede observar, este modelo es sencillo, sin embargo tiene algunas limitantes:

- Las funciones lineales de los multiplicadores deberán ser afinadas continuamente
- Todo el modelo depende de la elección de un precio inicial dado

CAPÍTULO 2: MARCO TEÓRICO

- Se debe estimar la elasticidad del precio, esta estimación tendrá un error

Sin embargo, una gran ventaja de este modelo es la forma transparente en la que opera, permitiendo que el equipo administrativo de las propiedades se familiarice con estos sistemas.

Modelo de maximización sujeto a restricciones

Un hotel típicamente designa un conjunto definido de categorías de precio, posteriormente asigna un inventario a cada categoría, de tal forma que las reservaciones que se hacen con un mayor tiempo de anticipación reciben un precio mas bajo y conforme el inventario se va agotando se reservan las habitaciones con una categoría de precio mayor. El problema con este enfoque radica en que si se asigna un gran número de cuartos a la categoría de precios mas baja, se tendrá un gran número de reservaciones pero a costo de una pérdida en el ingreso generado, y por el contrario, si se asigna un mayor número a la categoría con el precio mas alto se dejaran mas cuartos sin vender.

Este tipo de modelos permite al equipo que administra la propiedad encontrar el precio óptimo para cada categoría en cada noche de tal forma que el ingreso se maximice. Esto conlleva a un sofisticado problema de optimización que tome en cuenta las reservaciones futuras y su probabilidad de ocurrencia. Los precios son dinámicos y cambian día con día. Hay que mencionar que para que este modelo sea certero depende de una buena estimación de la demanda futura de los cuartos.

Para formular el modelo, se define una estancia por los parámetros (a, L) , donde a es la primera noche de estancia y L es la duración de la estancia, de tal forma que el modelo queda definido como sigue:

Maximizar

$$\sum_{l=1}^{Max\ l} P_l O_l$$

Sujeto a

$$O_l \leq C_l \quad \forall l$$

$$P_l \geq 0 \quad \forall l$$

CAPÍTULO 2: MARCO TEÓRICO

Dónde las variables de decisión del modelo son:

P_l : Precio para la noche $l \forall l$

Y las variables auxiliares:

$X_{a,L}$: El numero de cuartos asignado a una estancia del tipo (a, L) definido como:

$$X_{a,L} = d_{a,L} \left(\frac{\sum_{l=a}^{a+L-1} P_l}{L * P_{nominal}} \right)^e$$

O_l : Es el número de habitaciones reservadas para una noche dada y está definida como sigue

$$O_l = \sum_{a,L \in N_l} X_{a,L}$$

Y las entradas del modelo:

$P_{nominal}$: Precio nominal de la propiedad (usualmente la tarifa promedio histórica)

e : Elasticidad entre precio y demanda

$d_{a,L}$: La demanda de la estancias tipo a, L

N_l : El número de estancias que utilizan la noche l

C_l : El número de cuartos disponibles en el hotel

La salida de este modelo serán los precios optimos para cada noche.

Es importante notar que este modelo conlleva un sofisticado problema de optimización que requiere un poder de cómputo considerable para poder ser resuelto y además se debe iterar sobre distintos valores para el parámetro de elasticidad, obtenido a partir del análisis del comportamiento histórico del precio vs la ocupación de la propiedad, para encontrar el valor que garantice una solución factible del problema.

Modelo de optimización de redes

El modelo de optimización de redes es una formula de programación estocástica que permite capturar la aleatoriedad de la demanda desconocida, por ejemplo, el número de llegadas a la propiedad para cierto día o la longitud de estancia para una reserva dada)

CAPÍTULO 2: MARCO TEÓRICO

Antes de entrar al detalle de este tipo de modelos debemos de tomar en cuenta la siguiente notación:

- $x_{i,j}$ es el número de reservaciones confirmadas para hacer *check-in* en el día i y para hacer *check-out* en el día j , dónde $0 \leq i \leq j \leq T$. $i = 0, 1, 2, \dots, T-1$ es el índice de tiempo para *check-in* y $j = 1, 2, 3, \dots, T$ es el índice para *check-out*
- C es la capacidad total del hotel (Cuartos disponibles).
- $R_{i,j}$ es el ingreso ganado por la reservación con *check-in* en el día i y *check-out* en el día j .
- $U_{i,j}$ es la demanda de reseravs con *check-in* del día i y *check-out* el día j .

Nótese que:

- $\sum_{j=i+1}^T x_{i,j}$ es el número de *check-ins* en el día i .
- $\sum_{i=0}^{j-1} x_{i,j}$ es el número de *check-outs* en el día j .

Se asume que no hay clientes con estancias en días anteriores al día 0 y que todos los clientes salen del hotel antes, o en el día T . También se asume que cualquier cliente debe permanecer en el hotel por lo menos una noche.

Formulación de red estocástica

Los *check-ins* y *check-outs* pueen ser modelados como flujos de entrada y de salida de los nodos en una red. En la figura 2.1 se considera un día particular, el día k ($k = 1, 2, \dots, (T-1)$).

La siguiente ecuacion modela la ocupacion del hotel para el día k con $k = 1, 2, 3, \dots, (T-1)$:

$$\sum_{i=0}^{k-1} \sum_{j=k+1}^T x_{i,j} + \sum_{j=k+1}^T x_{k,j} - \sum_{i=0}^{k-1} x_{i,k}$$

Debemos considerar también que el hotel tiene una capacidad limitada, por lo tanto se tienen las siguientes restricciones para el día k :

$$\sum_{i=0}^{k-1} \sum_{j=k+1}^T x_{i,j} + \sum_{j=k+1}^T x_{k,j} - \sum_{i=0}^{k-1} x_{i,k} \leq C$$

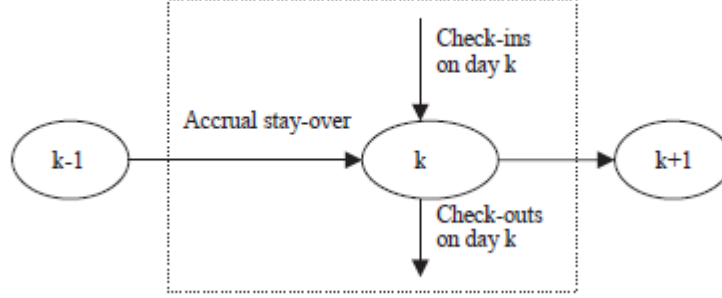


Figura 2.7: Flujo de check-ins y check-outs para el día k.

Para el día 0, se asume que no hay *check-outs* ni estancias que crucen a ese día, de tal forma que se tiene la siguiente ecuación para el día 0:

$$\sum_{j=1}^T x_{0,j} \leq C$$

El modelo de optimización se define como sigue:

Maximizar:

$$\sum_{i=0}^{T-1} \sum_{j=i+1}^T R_{i,j} x_{i,j}$$

Sujeto a:

$$\sum_{i=0}^{k-1} \sum_{j=k+1}^T x_{i,j} + \sum_{j=k+1}^T x_{k,j} - \sum_{i=0}^{k-1} x_{i,k} \leq C$$

$$\sum_{j=1}^T x_{0,j} \leq C$$

$$x_{i,j} \leq U_{i,j}$$

$$x_{i,j} \geq 0$$

$$\forall 0 \leq i < j \leq T$$

Si bien este problema pareciera ser un problema de integración lineal, se debe considerar que los parámetros $U_{i,j}$ son desconocidos al inicio del periodo de planeación. Mas aún, los ingresos pudieran no ser fijos como se quisiera ya que

CAPÍTULO 2: MARCO TEÓRICO

recordemos que se pueden fijar distintos precios para la misma habitación, lo que puede resultar en una variación de la demanda del producto. Una manera de resolver este problema pudiera ser reemplazando los parámetros por su mejor estimador puntual, en primera instancia utilizando la esperanza de $E(U_{i,j})$.

Muchas veces utilizando este método se pueden obtener resultados razonables, sin embargo esto no siempre garantiza una solución factible y para solucionar esto se han utilizado otras herramientas como la *Optimización Robusta* que es un enfoque proactivo usado para resolver problemas estocásticos.

Resumen

Los modelos de optimización de ingresos han tenido un gran avance en la industria de las aerolíneas, sin embargo, aún hay un trecho muy largo por recorrer para poder aplicarlos con el mismo nivel de madurez en la industria de la hotelería.

Como se revisó anteriormente, existen muchas propuestas de enfoques diferentes para la implementación del modelo de optimización de ingresos en hoteles y la elección de alguno de ellos dependerá de los siguientes factores:

- Habilidades estadísticas del equipo administrativo de la propiedad
- Nivel de desagregación de datos con los que cuenta la propiedad
- Capacidad de cómputo con el que la propiedad cuenta para el cálculo del modelo
- Nivel de automatización de procesos dentro de la propiedad, es decir, si cuenta con la capacidad de poder hacer cambios de precios e inventarios en línea

Para fines de este trabajo de investigación utilizaremos, dados los alcances buscados por la cadena de hoteles en cuestión, se optó por la construcción de un modelo convencional compuesto por dos módulos: un modelo de regresión poisson para el cálculo del pronóstico de la demanda, y un modelo de maximización sujeto a restricciones para el cálculo de precios óptimos por día de reservación. Esto nos facilitó la explicación del funcionamiento del modelo al equipo encargado de la gestión comercial de los hoteles, permitiendo que ellos aportaran

CAPÍTULO 2: MARCO TEÓRICO

parte de su conocimiento sobre el comportamiento comercial de los hoteles y las plazas durante la etapa de construcción, enriqueciendo el modelo final.

El primer módulo pronósticará la demanda utilizando datos para un hotel de negocios ubicado en una de las avenidas más importantes de la Ciudad de México. El set de datos con el que se contó durante esta investigación contenía el detalle de cada una de las reservaciones realizadas para esta propiedad desde 2014 hasta el 11 de agosto de 2018.

El segundo módulo asignará los precios óptimos para un tipo de habitación estándar tomando en cuenta los días de antelación con los que se reserva una habitación en la propiedad, de tal forma que al finalizar el proceso se cuente con una matriz de días de antelación vs precio óptimo.

En los apartados posteriores se explicará a detalle la construcción del modelo propuesto y los resultados obtenidos.

Capítulo 3

Metodología

En el presente capítulo se presentará la metodología utilizada para la creación de un módulo propuesto para la predicción de demanda y maximización de ingresos en hoteles. La metodología guiará la estructura de este trabajo con el fin de hacer de él una contribución científica. La presente investigación se puede dividir en dos partes. La primera, consistió en el desarrollo un modelo capaz de predecir la demanda y ofrecer recomendaciones de precios para maximizar el ingreso diario de una propiedad. La segunda parte consistió en probar dicho modelo y discutir los resultados obtenidos con la finalidad de abrir nuevas líneas de investigación.

Diseño del modelo

Para poder construir un modelo de predicción de demanda y maximización de ingresos para los hoteles que fueron sujetos al estudio, se siguieron una serie de pasos descritos a continuación:

Entendimiento del negocio

El primer paso durante la construcción del modelo fue el entendimiento del negocio ya que para poder interpretar los datos es necesario comprender el negocio que está implícito en los datos. Durante esta etapa se trabajó con el

CAPÍTULO 3: METODOLOGÍA

equipo de gestión comercial de las propiedades para entender qué datos son analizados durante el proceso de toma de decisiones, de dónde son obtenidos estos datos y qué variables son de mayor interés para el negocio.

Identificación de las fuentes de datos

Una vez comprendido el negocio y detectadas las variables de interés durante el proceso de la toma de decisiones, se identificaron las fuentes de datos, las cuales son nutridas por dos principales sistemas dentro de las propiedades: la central de reservaciones y el sistema de gestión de la propiedad. El primero es el sistema que sirve para dar servicio a todos los huéspedes que buscan una habitación mediante los canales de reservación disponibles (sitio web, call center, aplicaciones móviles, correos electrónicos, etc) y el segundo es que gestiona la propiedad y da servicio al huésped desde que llega hasta que se retira de la propiedad. Estos sistemas contienen toda la información del proceso desde que se encuentra en estatus de reservación hasta que llega a un estatus de *check out*.

Análisis exploratorio de datos

Luego de haber identificado las principales fuentes de datos, se diseñaron los procesos de extracción de información necesarios para obtener los datos que serían utilizados para la construcción del modelo. Para poder entender los datos, se ejecutó un análisis exploratorio de datos, en dónde se pudo observar el comportamiento de las variables de interés, su distribución probabilística, se detectaron temporalidades en los datos, nivel de limpieza de datos y datos atípicos que pudieran generar problemas al momento de la implementación del modelo final.

Construcción del modelo

Para la construcción del modelo se tuvieron varias iteraciones en el desarrollo. La primera, consistió en el diseño de los procesos *ETL* para la construcción de los *sets de datos* que alimentan a los tres modelos de pronóstico de demanda propuestos. La segunda iteración se enfocó en la construcción de los tres modelos de pronóstico de demanda, durante esta etapa se analizaron distintos enfoques

CAPÍTULO 3: METODOLOGÍA

como: regresiones lineales generalizadas, modelos de series de tiempo (ARIMA) y finalmente regresiones de aprendizaje de máquina (regresión de Ridge), siendo este el modelo que mostró un mejor desempeño. Durante la tercera iteración se construyó el modelo de optimización de ingresos. Este modelo toma como datos de entrada los resultados obtenidos por el modelo de predicción de demanda y está definido como un problema de maximización de una función sujeta a distintas restricciones. La última iteración consistió en la ejecución de el modelo integral para poder medir el desempeño y poder obtener los resultados finales.

Pruebas y Resultados

Durante esta etapa se ejecutó el modelo para pronósticar la demanda y maximizar los ingresos para un mes posterior a la fecha de ejecución, los resultados fueron comparados con el desempeño real de la propiedad para obtener el *MA-PE* del modelo construido. En el capítulo de *Resultados* del presente trabajo se presentarán los detalles de las pruebas ejecutadas y los resultados obtenidos.

Capítulo 4

Diseño y Desarrollo

Este capítulo describirá el diseño y desarrollo del modelo de predicción de demanda y optimización de ingresos, desde la arquitectura de la solución propuesta, los *data sets* utilizados y finalmente los módulos de la solución y su funcionamiento.

Arquitectura del Modelo

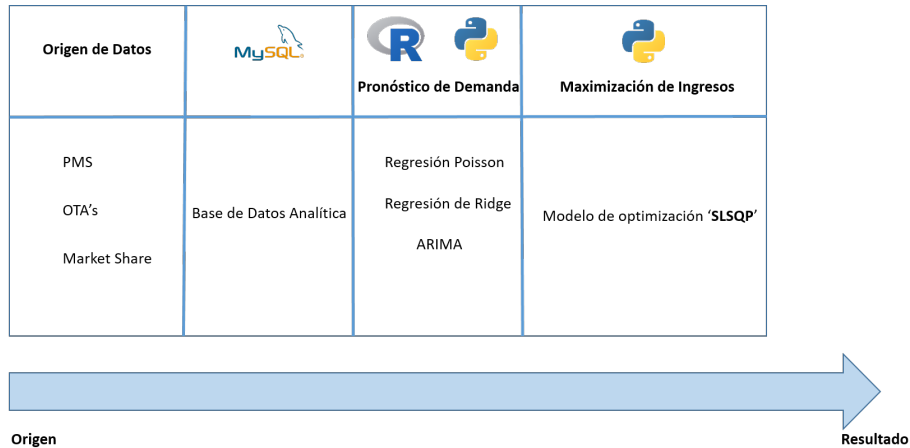


Figura 4.1: Arquitectura del modelo de predicción de demanda y maximización de ingresos

Descripción de la arquitectura del modelo

Como se puede observar en la figura 4.1, el modelo está compuesto por 4 capas o módulos principales. La primera capa son los sistemas transaccionales que contienen información clave para el funcionamiento del modelo propuesto. El primer sistema de interés es el *PMS* o *Property Management System*. Este sistema es el encargado de gestionar las propiedades y captar las reservaciones que se hacen en cada una de ellas desde los distintos canales de reservación. En este sistema podemos encontrar información de las estancias pasadas de los huéspedes en las distintas propiedades, información de huéspedes que se encuentran en la propiedad, historial de consumos, habitaciones asignadas, formas de pago de cada uno de los consumos, reservas canceladas y huéspedes que reservaron pero no se presentaron en la propiedad. Es de este sistema de donde se obtendrá la información que describe el comportamiento de la demanda de la propiedad que es de interés para este trabajo.

El segundo sistema analizado fueron un conjunto de sitios web de agencias de viajes en línea. Estos sitios contienen información de la tarifa publicada en la red

CAPÍTULO 4: DISEÑO Y DESARROLLO

para diferentes tipos de habitación de la propiedad estudiada en este trabajo así como de su set competitivo (el conjunto de hoteles cercanos a la propiedad estudiada que tienen el mismo mercado objetivo). Para poder obtener la información de estos sitios, se consumió un servicio web que dado un identificador de una propiedad y un rango de fechas, devuelve el precio asociado a cada tipo de habitación disponible en el rango de fechas indicadas. Esta información es de vital importancia para poder obtener parámetros objetivos de los precios asociados al producto ofrecido en el mercado y así evitar que el modelo de optimización de ingresos devuelva precios sugeridos que se saquen al producto estudiado del mercado objetivo.

Finalmente dentro de esta capa se cuenta con un tercer sistema que provee información al modelo. Se trata de un aplicativo de intercambio de información entre propiedades del set competitivo sin importar la marca a la que pertenezcan. Todos los días al iniciar el día, los gerentes de cada propiedad hacen una llamada telefónica a las propiedades con las que compiten y recaban información sobre habitaciones ocupadas e ingresos obtenidos por concepto de renta habitación y con esto se calculan diferentes kpi's: % ocupación, tarifa promedio, tarifa efectiva, etc. Dicha información es de interés para el modelo ya que con ella se puede obtener una estimación de la demanda de cuartos que existe en la plaza donde se encuentra la propiedad de interés para este trabajo de investigación así como los precios con los cuales se esta vendiendo el producto en el mercado objetivo de esta plaza.

Una vez obtenida la información de los sistemas transaccionales, se ejecuta un proceso de *Transformación y Carga* de información que depósita la información de interés en una base de datos analítica (segunda capa). Esta base de datos está diseñada para poder analizar la información de manera ágil. Es aquí donde se puede cruzar la información de los diferentes sistemas ya que durante el proceso de *Transformación y Carga* se homolgan los identificadores de las distintas propiedades antes de ser depositadas en el modelo de datos analítico.

La información es procesada en *R* y *python* con el fin de alimentar tres modelos que pronóstican demanda de una propiedad. El primer modelo esta basado en una regresion lineal generalizada con liga Poisson, el segundo se trata de un modelo de inteligencia artificial basado en una regresión de ridge, y finalmente se cuenta con un modelo de análisis de series de tiempo *ARIMA*. Una vez

CAPÍTULO 4: DISEÑO Y DESARROLLO

ejecutados, se elige el resultado del modelo con el mejor desempeño tomando en cuenta el *MAPE* de cada uno de los resultados arrojados por los modelos, para finalmente alimentar al modelo de optimización de ingresos desarrollad en *python* con ayuda de la librería *scipy*.

Al finalizar la ejecución de esta aplicación el usuario obtiene como resultado una matriz en donde se visualizan días de venta futura vs el inventario disponible y el precio asignado a cada nivel de inventario disponible.

En las siguientes secciones de este capítulo se describirá de manera detallada la capa 2 y capa 3 de la solución propuesta.

Regresión lineal generalizada con liga Poisson

Para la construcción del modelo de predicción de demanda basado en una regresión lineal generalizada con liga Poisson se utilizaron tres data sets que contienen la información suficiente para poder hacer un pronóstico efectivo de la demanda de una propiedad.

Detalle de reservaciones

El primer data set utilizado fue extraído directamente del sistema de gestión de propiedades (PMS) y contiene información detallada de todas las reservaciones generadas en la propiedad, así como la información de las estancias pasadas. Para poder obtener esta información se realizaron procesos automáticos de extracción, transformación y carga de información.

A continuación se presenta el detalle de la información obtenida del PMS así como el diccionario de datos y un breve resumen de la información en cada una de las columnas.

Diccionario de Datos

- *rsrv_code*: Código de confirmación de la reservación
- *date_create*: Fecha y hora de creación de la reserva
- *date_in*: Fecha de llegada a la propiedad

CAPÍTULO 4: DISEÑO Y DESARROLLO

- `date_out`: Fecha de salida de la propiedad
- `nights`: Número de noches de la reservación en el hotel
- `prop_code`: Código de la propiedad dentro del sistema de gestión de propiedades
- `mkt_sgm`: Segmento de mercado del huésped amparado en la reservación
- `Dia_Sem`: Día de la semana de la fecha de llegada del huésped
- `rate_code`: Código de tarifa
- `bucket`: Categoría de tarifa
- `Ingresos`: Ingresos obtenidos por la renta habitación de la reservación
- `rsrv_src`: Canal de generación de la reservación
- `rsrv_type`: Tipo de reservación
- `room_type`: Tipo de habitación
- `PAX`: Número de personas amparadas por la reservación

CAPÍTULO 4: DISEÑO Y DESARROLLO

Resumen de datos

Variable	Tipo de dato	Min	Max	Media
rsrv_code	Big Int	6265739	9053354	NA
date_create	Fecha	2016-07.20	2018-01-01	NA
date_in	Fecha	2017-01-01	2017-12-31	NA
date_out	Fecha	2017-01-02	2018-01-01	NA
prop_code	Factor	NA	NA	NA
mkt_sgm	Factor	NA	NA	NA
Dia_sem	Factor	NA	NA	NA
rate_code	Factor	NA	NA	NA
bucket	Factor	NA	NA	NA
rsrv_src	Factor	NA	NA	NA
rsrv_type	Factor	NA	NA	NA
Ingresos	Float	0	3900	1449
PAX	Integer	0	5	1.271

Tabla 4.1: Resumen de datos obtenidos del PMS

Es importante resaltar que los datos fueron trabajados para presentarse de manera desagregada, es decir, si una reservación ampara una estancia de 5 noches, en el *data set final* se presenta un registro por cada noche de la estancia, por ejemplo:

rsrv_code	date_create	date_in	date_out	nights	room_type
6265739	2016-07-20 18:31:59	2017-04-07	2017-04-08	1	NSK
6265739	2016-07-20 18:31:59	2017-04-08	2017-04-09	1	NSK
6265739	2016-07-20 18:31:59	2017-04-09	2017-04-10	1	NSK
6265739	2016-07-20 18:31:59	2017-04-10	2017-04-11	1	NSK
6265739	2016-07-20 18:31:59	2017-04-11	2017-04-12	1	NSK
6265739	2016-07-20 18:31:59	2017-04-12	2017-04-13	1	NSK

Tabla 4.2: DataSet detalle de reservas

Como se puede observar, el número de reservación es el mismo en los seis registros, lo mismo ocurre con la fecha de creación, las noches y el tipo de habitación,

sin embargo, la fecha de entrada y la fecha de salida va cambiando en cada registro, lo que nos permite medir la demanda generada por aquellas reservaciones que contemplan más de una noche de estancia en la propiedad.

Curvas de Pickup

Del data set anterior podemos obtener las curvas de pickup del hotel. Estas curvas describen el comportamiento de las reservaciones para la propiedad en cuestión ya que mediante ellas podemos saber con cuanto tiempo de anticipación se comienzan a reservar las habitaciones de la propiedad. A continuación se muestran las curvas de pickup para esta propiedad:

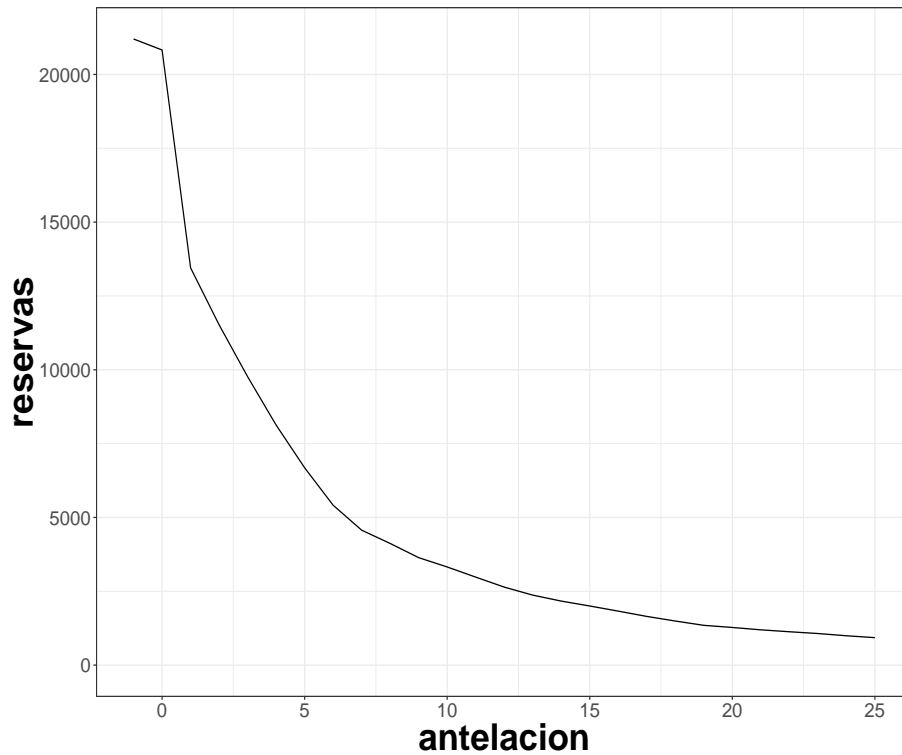


Figura 4.2: Curva de pickup

Analizando las gráficas presentadas anteriormente podemos concluir que la demanda de esta propiedad comienza 25 días antes de la llegada del huésped al

CAPÍTULO 4: DISEÑO Y DESARROLLO

hotel, e incrementa considerablemente 5 días antes de la llegada del huésped al hotel. Este dato nos indica que el huésped comienza a buscar una habitación dentro de esta propiedad en un lapso no mayor a 25 días antes de emprender su viaje y la demanda de habitaciones incrementa considerablemente 5 días antes de la fecha de llegada a la propiedad.

Las curvas de *pickup* son importantes dentro del proceso de toma de decisiones ya que le indican al equipo que gestiona la propiedad el tiempo de antelación con el cual deberían llevar a cabo las acciones necesarias para poder optimizar el ingreso generado por la renta de habitaciones.

Ocupación de la propiedad

El segundo *data set* con el que se trabajó contiene las líneas de tiempo de los niveles de ocupación de la propiedad. Este set de datos es de suma importancia ya que proporciona información relevante con respecto a las temporalidades del hotel, mismas que deben ser reflejadas en el resultado del modelo.

Las temporalidades del hotel pueden ser desde periodos con alta o baja ocupación, inclusive por día de la semana ya que al ser un hotel enfocado a viajeros de negocio, se esperaría tener una alta ocupación los días laborales (Lunes a Jueves) y una baja ocupación los fines de semana (Viernes a Domingo).

A continuación se presenta el contenido de este *data set*:

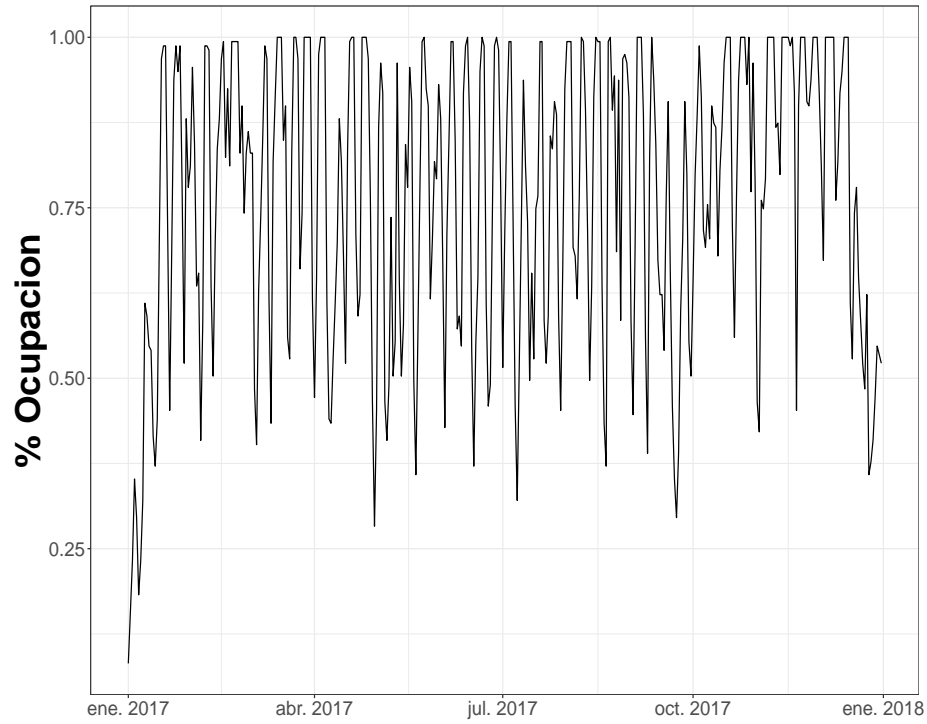


Figura 4.3: % Ocupación 2017

Como se puede observar en la gráfica presentada anteriormente, esta propiedad tiene altos niveles de ocupación (alredor del 75 % de ocupación) agotando en repetidas ocasiones su inventario disponible (159 cuartos). Las caídas en los niveles de ocupación ocurren los fines de semana, lo cual confirma que el mercado objetivo de la propiedad estudiada es el turismo de negocios.

A continuación se presenta una gráfica de niveles de ocupación por día de la semana:

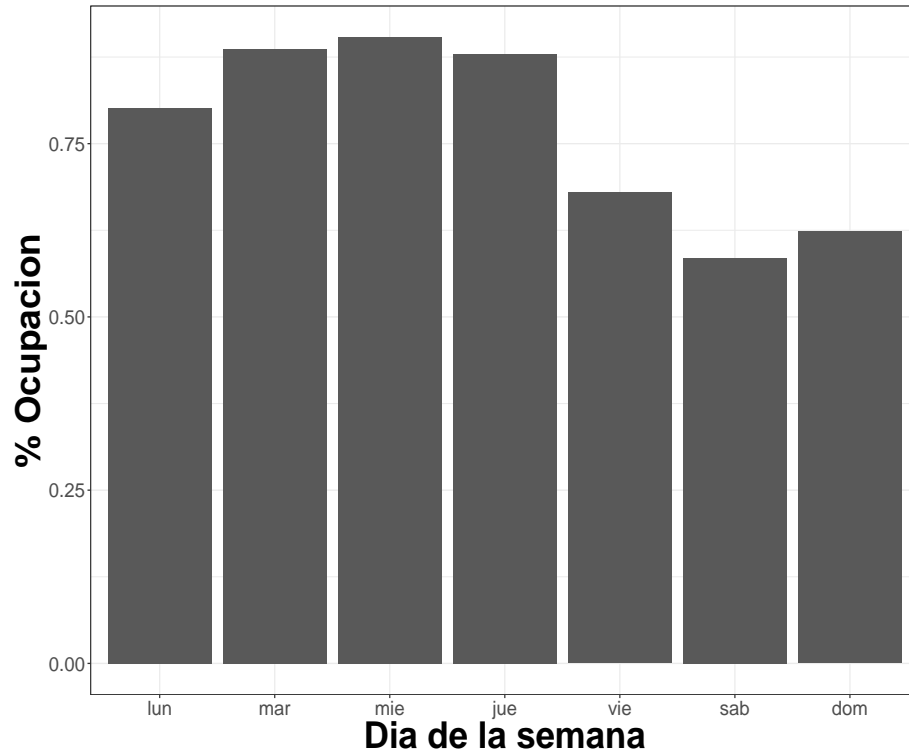


Figura 4.4: Ocupación por día de la semana

Tarifa Promedio de la propiedad

El tercer conjunto de datos utilizado contiene los valores de la tarifa promedio *ADR: Average Daily Rate* en el tiempo. Estos datos fueron obtenidos directamente del PMS y es el resultado de sumar los ingresos generados por las habitaciones divididos por el número total de cuartos vendidos.

A continuación se presenta la línea de tiempo con los valores para la tarifa promedio a lo largo del tiempo.

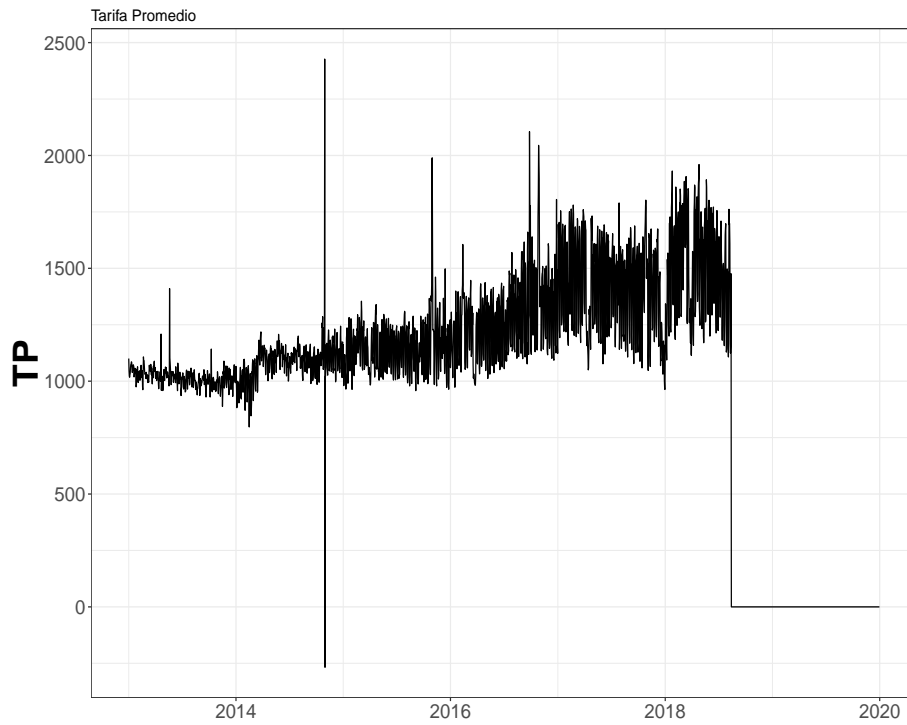


Figura 4.5: Tarifa Promedio 2017

De esta última gráfica se puede observar que los precios siguen una tendencia positiva con respecto al tiempo, a medida que pasa el tiempo, los precios incrementan, sin embargo podemos notar también que a medida que pasa el tiempo los precios varían más. Si contrastamos la información presentada en la gráfica de la tarifa promedio vs el tiempo (Figura 4.4) y la presentada en la gráfica de ocupación vs el tiempo (Figura 4.2) podemos ver que aunque los precios han aumentado en esta propiedad, las tendencias en los niveles de ocupación permanecen constantes.

Precios Públicos en el tiempo

Como se mencionó anteriormente, una propiedad típicamente tiene diferentes precios disponibles para cada uno de los segmentos de mercado a los que atiende. La tarifa que se ofrece al público que llega al hotel sin una reservación previa se

CAPÍTULO 4: DISEÑO Y DESARROLLO

le conoce como *tarifa pública* y generalmente es la tarifa con el precio mas alto de la cual desprenden todas las demás tarifas, es decir, el conjunto de tarifas disponible en el hotel serán descuentos realizados sobre la *tarifa pública*.

El tercer data set utilizado durante el desarrollo del modelo, contiene la información de las tarifas públicas de la propiedad y su competencia a lo largo del tiempo. Esto nos ayudará a conocer el precio del producto ofrecido en el mercado objetivo, de tal forma que los precios arrojados por el modelo se encuentren siempre dentro del mercado.

Los precios de la competencia fueron obtenidos de las páginas electrónicas de las agencias en línea, booking.com y expedia.com principalmente.

A continuación se presentan las líneas de tiempo con los precios públicos para el conjunto de propiedades.

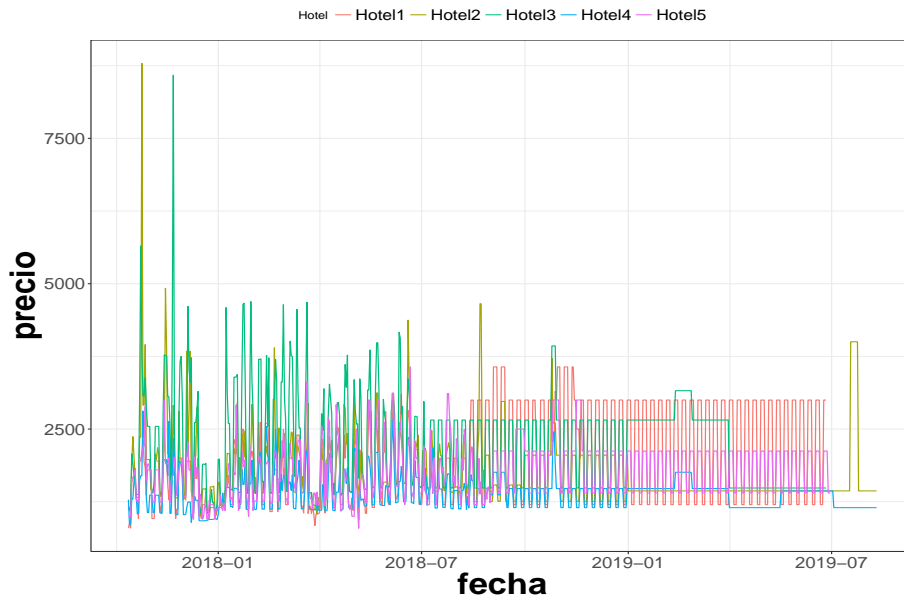


Figura 4.6: Precios de la competencia en el tiempo

En la gráfica anterior se puede observar que hay algunos datos atípicos ya que es poco probable encontrar habitaciones en este segmento de hoteles por arriba de los \$4,000 MXN. Para confirmar esto se realizó una gráfica de caja y bigotes (boxplot) en la cual se puede observar a primera vista los datos atípicos

contenidos en el *data set*.

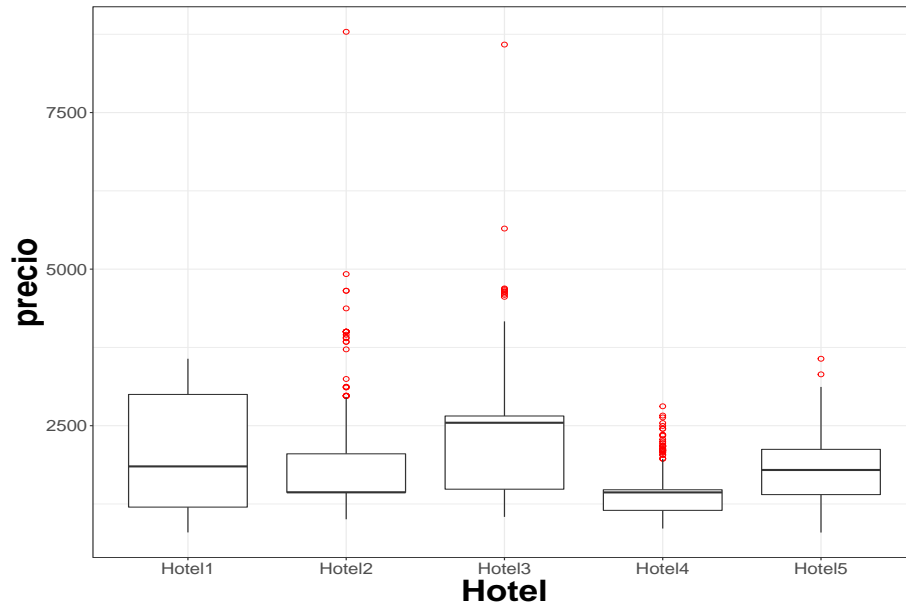


Figura 4.7: Precios de la competencia en el tiempo

Los datos atípicos fueron eliminados antes de proceder a trabajar con el modelo final.

En el siguiente apartado detallaremos cómo se utilizaron los datos presentados para la construcción del modelo de predicción de demanda y maximización de ingresos.

Modelo de Predicción de Demanda - glm con liga Poisson

El modelo de predicción de demanda toma como entrada los datos de las curvas de *pickup* para el hotel para cada día del periodo estudiado. Estas curvas describen la velocidad con la que una propiedad vende su inventario para un día en específico. El objetivo del modelo propuesto es utilizar estos datos para ajustar una curva que resuma el comportamiento de la propiedad en un día similar al estudiado y de esta forma generar una predicción de los cuartos ocupados en la propiedad y los días de antelación con los que serán vendidos.

CAPÍTULO 4: DISEÑO Y DESARROLLO

Este modelo está compuesto por dos módulos principales. El primero es una regresión *Poisson* definida como $\log(E(Y|X)) = \beta_0 + \beta_1 X$, que se alimentará con las curvas de *pickup* que describen el comportamiento del hotel. Como resultado de este módulo obtendremos un parámetro β_0 y un parámetro β_1 que describen la curva que mejor ajusta al *pickup* del hotel.

A continuación se muestra un resumen del conjunto de datos que alimentan al primer módulo del modelo de predicción de demanda:

date_create	date_in	antelacion	rooms
2017-03-16	2017-05-01	45	1
2017-03-30	2017-05-01	31	2
2017-04-02	2017-05-01	28	3
2017-04-04	2017-05-01	26	4
2017-04-11	2017-05-01	19	7
2017-04-17	2017-05-01	13	8
2017-04-18	2017-05-01	12	9
2017-04-20	2017-05-01	10	10
2017-04-21	2017-05-01	9	11
2017-04-24	2017-05-01	6	20
2017-04-25	2017-05-01	5	24
2017-04-26	2017-05-01	4	28
2017-04-27	2017-05-01	3	34
2017-04-28	2017-05-01	2	56
2017-04-29	2017-05-01	1	62
2017-04-30	2017-05-01	0	71
2017-05-01	2017-05-01	-1	72

Tabla 4.3: Datos de entrada para el modelo de pronóstico de demanda

Una vez obtenidos los parámetros β_0 y β_1 de la regresión *Poisson*, se procede a generar predictores potenciales de la demanda a partir de los datos con los que contamos para este análisis.

Los predictores definidos son:

- diaSem: Definido como factores que van de Lunes a Domingo

CAPÍTULO 4: DISEÑO Y DESARROLLO

- Mes: Definido como factores que van de Enero a Diciembre
- Eventos: Toma el valor 1 si la plaza cuenta con algun evento social, deportivo, o de entretenimiento en esa fecha, 0 en caso contrario
- PT: Definido como la tarifa promedio de la propiedad dividida entre la tarifa promedio de la competencia
- β_0 : El parámetro β_0 obtenido de la regresión *Poisson*
- β_1 : El parámetro β_1 obtenido de la regresión *Poisson*

Con estos predictores definidos se procede a alimentar una regresión lineal que calculará nuevos parámetros β_{0*} y β_{1*} . A partir de estos nuevos parámetros se construyen las curvas de *pickup* que ayuden a predecir la demanda de la propiedad en días futuros.

Una vez concluido el cálculo, se genera una matriz en la cuál se plasma la estimación del número de *cuartos* ocupados que en una ventana futura:

Dia	Cuartos Ocupados
2018-08-14	92
2018-08-15	78
2018-08-16	96
2018-08-17	99
2018-08-18	132
2018-08-19	151
2018-08-20	138
2018-08-21	88
2018-08-22	77

Tabla 4.4: Pronóstico de cuartos noche ocupados (Regresión con liga Poisson)

Regresión de Ridge

El segundo modelo propuesto se trata de una regresión de Ridge, el cuál es un método de *machine learning* que consiste en incluir en la función de minimiza-

CAPÍTULO 4: DISEÑO Y DESARROLLO

ción del error un término de penalización que permite evitar el sobreajuste del modelo y proporciona un menor error de generalización.

Este modelo, al tratarse de una técnica de *machine learning* requiere que los datos de entrada sean modelados de manera tal que existan n variables independientes que expliquen una variable dependiente y a la cuál se ajustará la regresión.

Datos de entrada

Para poder aplicar la *regresión de Ridge* se utilizó una serie temporal que incluye la información de los cuartos ocupados de la propiedad para cada día del año desde el 01 de enero de 2013 hasta el 14 de agosto del 2018, como se muestra a continuación:

CAPÍTULO 4: DISEÑO Y DESARROLLO

Fecha	Cuartos ocupados
01/01/13	31
02/01/13	38
03/01/13	45
04/01/13	39
05/01/13	42
06/01/13	36
07/01/13	68
08/01/13	68
09/01/13	99
10/01/13	126
11/01/13	70
12/01/13	51
13/01/13	52
14/01/13	104
15/01/13	117
16/01/13	145
17/01/13	156
18/01/13	109
19/01/13	91
20/01/13	81
21/01/13	120
22/01/13	159
23/01/13	159
24/01/13	130
25/01/13	109
26/01/13	90
27/01/13	84
28/01/13	119
29/01/13	151
30/01/13	159

Tabla 4.5: Cuartos noche ocupados vs tiempo

Para poder utilizar la regresión propuesta se deben transformar los datos de

CAPÍTULO 4: DISEÑO Y DESARROLLO

manera que tengamos una columna y que contiene la ocupación del día que se quiere pronósticar, explicada por la ocupación de los n días pasados. A continuación se muestra un ejemplo:

	fecha	t-4	t-3	t-2	t-1	y
4	05/01/13	31	39	45	40	43
5	06/01/13	39	45	40	43	37
6	07/01/13	45	40	43	37	68
7	08/01/13	40	43	37	68	69
8	09/01/13	43	37	68	69	99
9	10/01/13	37	68	69	99	127
10	11/01/13	68	69	99	127	70
11	12/01/13	69	99	127	70	51
12	13/01/13	99	127	70	51	53
13	14/01/13	127	70	51	53	105
14	15/01/13	70	51	53	105	118

Tabla 4.6: Conjunto de datos de entrada para la regresión de Ridge

Adicionalmente se agregaron variables para indicarle al modelo el año, mes y día del que se trata, así como variables *dummy* que indican el día de la semana del que se trata, dejando el conjunto de datos final como se muestra a continuación:

year	month	day	F	M	SAT	SUN	TH	TU	WED
2013	1	5	0	0	1	0	0	0	0
2013	1	6	0	0	0	1	0	0	0
2013	1	7	0	1	0	0	0	0	0
2013	1	8	0	0	0	0	0	1	0
2013	1	9	0	0	0	0	0	0	1
2013	1	10	0	0	0	0	1	0	0
2013	1	11	1	0	0	0	0	0	0
2013	1	12	0	0	1	0	0	0	0
2013	1	13	0	0	0	1	0	0	0
2013	1	14	0	1	0	0	0	0	0
2013	1	15	0	0	0	0	0	1	0

Tabla 4.7: Variables adicionales de entrada para la regresión de Ridge

Implementación del modelo

La regresión de Ridge fue implementada en *python 3.5.2* con ayuda de la librería **sklearn**. Para poder entrenar al modelo se dividió el conjunto de datos en dos partes: el conjunto de entrenamiento (que contiene el 66 % de los registros) y el conjunto de pruebas (que contiene el 33 % restante).

Para entrenar el modelo se definieron 10 valores diferentes para el parámetro α = 1e-15, 1e-10, 1e-8, 1e-4, 1e-3, 1e-2, .1, 5, 10, 20 y se entrenó el modelo para cada uno de los valores de α buscando el menor *rss* definido como:

$$\sum_{i=1}^n (y_i - f(x_i))^2$$

Donde y_i es el valor a predecir y $f(x_i)$ es el valor pronosticado por el modelo.

El proceso anteriormente descrito se iteró n veces. En cada iteración se incrementaron las variables explicativas para la variable de salida, es decir, se incrementó el número de días anteriores que intentan explicar la ocupación de la propiedad en el día 0, de tal forma que la ocupación en el día t_0 es explicada por la ocupación en los días t_{-1}, \dots, t_{-n} para toda n en 1, 2, 3, ..., 400

A continuación se muestran los *rss* obtenidos para cada valor de α con $n = 400$

CAPÍTULO 4: DISEÑO Y DESARROLLO

Alpha Value	Rss
1e-15	7.2368
1e-10	7.2368
1e-08	7.2368
0.0001	7.2369
0.001	7.2381
0.01	7.3057
0.1	8.3252
5	18.247
10	21.69
20	25.9621

Tabla 4.8: Valores residuales cuadráticos para los distintos valores de α

Como se puede observar, el valor del parámetro α con el menor rss es $1 * e^{-15}$, es por ello que se eligió este modelo como el final.

Para poder pronósticar utilizando esta técnica se debe de definir una ventana que le indique al modelo cuantos días posteriores al día 0 se quieren pronósticar. De esta forma el modelo pronosticará la ocupación día t_1 y utilizará el resultado para pronósticar el día t_2 y así sucesivamente hasta pronósticar los días dentro de la ventana definida.

Utilizando el procedimiento anteriormente descrito, se validó el modelo final con el conjunto de datos de prueba y se obtuvo un $MAPE = 15.1815$.

A continuación se muestra una gráfica de los resultados del pronóstico, utilizando el conjunto de pruebas vs la ocupación real del hotel.

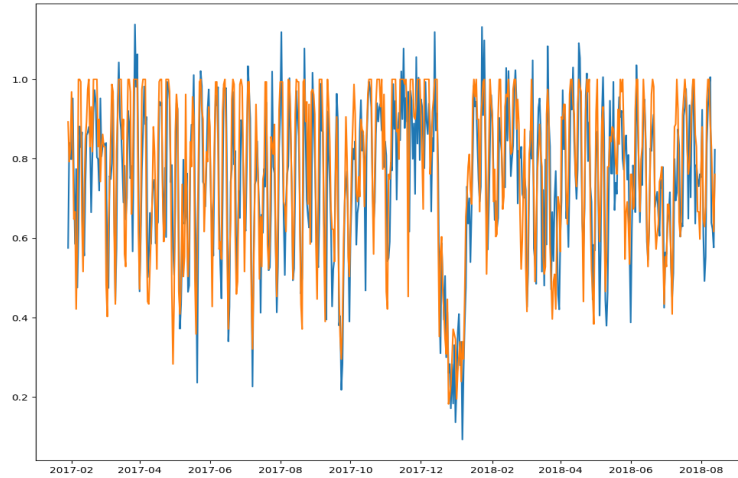


Figura 4.8: Validación de predicciones utilizando regresión de Ridge

Podemos notar que el modelo tiene un buen ajuste vs los datos reales, incluso cuando la ocupación real tiene una caída hacía finales del año 2017, el resultado del modelo presenta un comportamiento similar. Podemos notar que los pronósticos sobrepasan el 100 % de ocupación lo cuál indica que hay un alta probabilidad de poder sobrevender la propiedad en algunas fechas.

CAPÍTULO 4: DISEÑO Y DESARROLLO

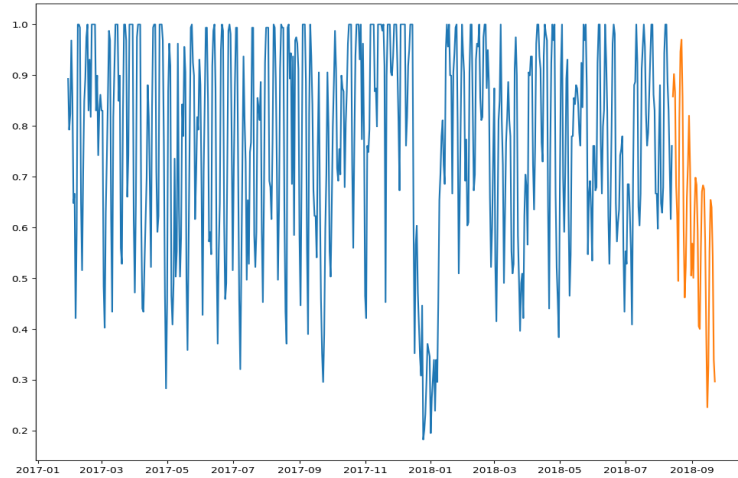


Figura 4.9: Generación de predicciones utilizando regresión de Ridge

Como resultado del modelo se genera una matriz en la cuál se plasma la estimación del número de *cuartos ocupados* que en la ventana futura definida:

Dia	Cuartos Ocupados
2018-08-14	136
2018-08-15	143
2018-08-16	138
2018-08-17	108
2018-08-18	99
2018-08-19	79
2018-08-20	130
2018-08-21	150
2018-08-22	154

Tabla 4.9: Pronóstico de cuartos noche ocupados (Regresión de Ridge)

Análisis de series de tiempo - SARIMA

El análisis de series de tiempo trata de extraer estadísticas significantes y otras características de los datos. El pronóstico a partir de series de tiempo utiliza un modelo estadístico para predecir valores futuros basandose en los datos previamente observados.

Este tipo de modelos son muy comunes para el análisis y pronóstico de datos no estacionarios como: indicadores económicos, clima, acciones, precios y ventas. Para fines de este trabajo se utilizará un modelo de series de tiempo para analizar y predecir valores para la ocupación del hotel que está siendo analizado.

Datos de entrada

Los datos de entrada para este modelo constan fueron modelados en forma de una matriz de fecha vs ocupación en esa fecha.

Graficando esta información podemos darnos cuenta que existen patrones marcados en los datos presentados:

CAPÍTULO 4: DISEÑO Y DESARROLLO

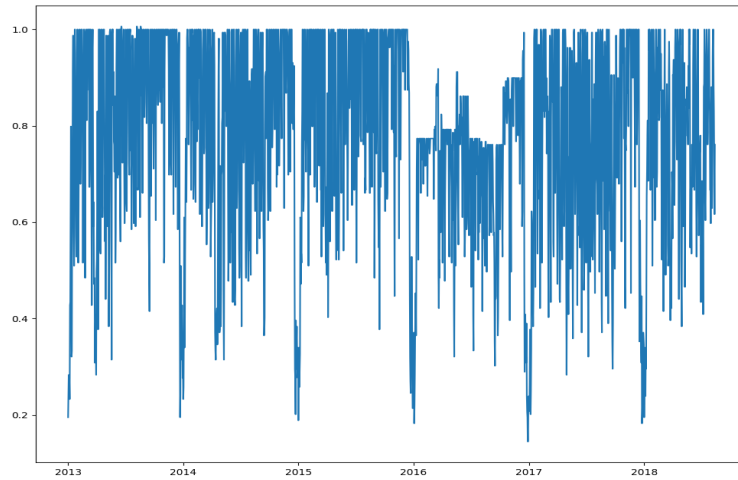


Figura 4.10: Serie de tiempo para la ocupación de la propiedad desde 2013 hasta 2018

Podemos observar que la ocupación del hotel es inestable, presentando temporadas de alta ocupación y algunas de baja ocupación (generalmente hacia finales del año corriente y principios del año siguiente).

Se puede ampliar en análisis descomponiendo esta serie de tiempo en su tendencia, temporalidad y el *ruido* contenido en la serie:

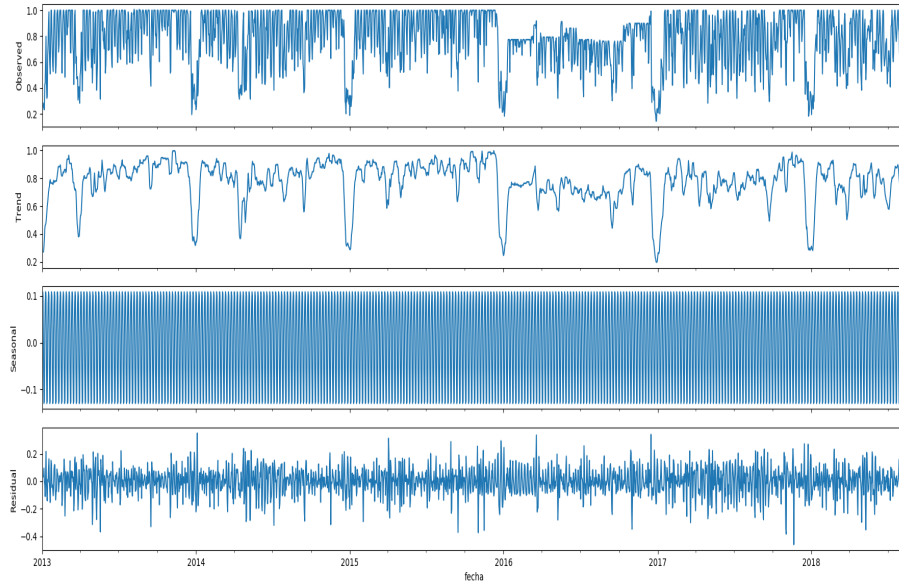


Figura 4.11: Descomposición de la serie de tiempo de ocupación

Esta descomposición confirma que la ocupación es inestable, sin embargo, en la gráfica de la tendencia se pueden observar las temporadas de alta ocupación y de baja ocupación, lo cual es de gran ayuda para entender el comportamiento de la ocupación en esta propiedad, además viendo estas gráficas podemos notar que los datos son estacionales, es decir, presentan un ciclo.

Implementación del modelo

Dada la estacionalidad presentada en los datos, el modelo que se implementará es un modelo *SARIMA* el cual es una variación del modelo *ARIMA* pero que es capaz de manejar la estacionalidad contenida en los datos.

El modelo *SARIMA* contiene siete hiperparámetros que deben ser optimizados para mejorar el desempeño del modelo. Estos siete parámetros son conocidos como $(p, d, q)(P, D, Q)m$. Los primeros tres definen la parte no estacionaria del modelo. Mientras que los últimos cuatro definen la parte estacionaria del modelo. Decimos que una serie de tiempo es estacionaria cuando la media y la varianza son constantes en el tiempo.

CAPÍTULO 4: DISEÑO Y DESARROLLO

El parámetro p implica que la salida del modelo en el tiempo t será una suma ponderada de los valores pasados más un término estocástico. Este valor es el número de *rezagos* que se usarán. El parámetro d es el número de diferencias no estacionarias aplicadas a la serie para convertirla en una serie estacionaria, en otras palabras, el modelo toma valores pasados y los sustrae del valor actual. Si tuviéramos una serie integrada de orden 2 y tomáramos diferencias repetidas dos veces para crear un proceso estacionario el modelo sería un ARIMA (0,2,0). El parámetro q es un promedio móvil de los errores previos. Este tipo promedios móviles son usados para modelar ruido que se disipa gradualmente dentro de un sistema.

La parte estacionaria del modelo $(P,D,Q)m$ tienen la misma estructura que la parte no estacionaria del modelo. En esta parte del modelo los tres factores operan únicamente a lo largo del periodo (m) definido para la estacionalidad del modelo. La estacionalidad es un patrón de cambios regular que se repite cada m periodos dentro de la serie

Para poder optimizar los hiperparámetros del modelo se implementó un algoritmo llamado *grid search* el cual busca el valor de los hiperparámetros que ofrecen el mejor desempeño del modelo, tomando el AIC de cada modelo como medida de desempeño. Este criterio es una medida de la calidad de un modelo estadístico para un conjunto dado de datos y está definido como:

$$AIC = 2k - 2\ln(L)$$

Dónde k es el número de parámetros en el modelo estadístico, y L es el máximo valor de la función de verosimilitud para el modelo estimado.

Se dividió el conjunto de datos de entrada en dos: entrenamiento y pruebas. El modelo SARIMA fue implementado utilizando la librería *statsmodels* en *python* 3.5.2. Una vez entrenado el modelo con el conjunto de entrenamiento para todos los valores contenidos en la lista de búsqueda del algoritmo de *Grid Search*, se eligió el modelo con el menor valor de AIC como el final.

El modelo fue validado utilizando el conjunto de pruebas y se pronosticaron valores para la ocupación de la propiedad en los años 2017 y 2018.

A continuación se muestra una gráfica con los resultados de la validación y se puede notar que los pronósticos obtenidos se alinean aceptablemente a la

CAPÍTULO 4: DISEÑO Y DESARROLLO

tendencia mostrada por los valores reales.

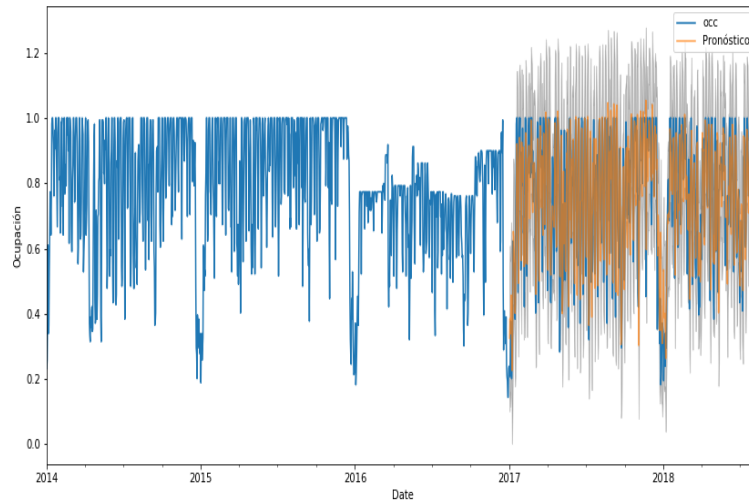


Figura 4.12: Validación de las predicciones generadas por el modelo SARIMA

El MAPE reportado por este modelo fue de 14.8880.

Posteriormente se pronosticaron valores para días mayores al 13 de agosto de 2018 obteniendo los resultados mostrados a continuación.

CAPÍTULO 4: DISEÑO Y DESARROLLO

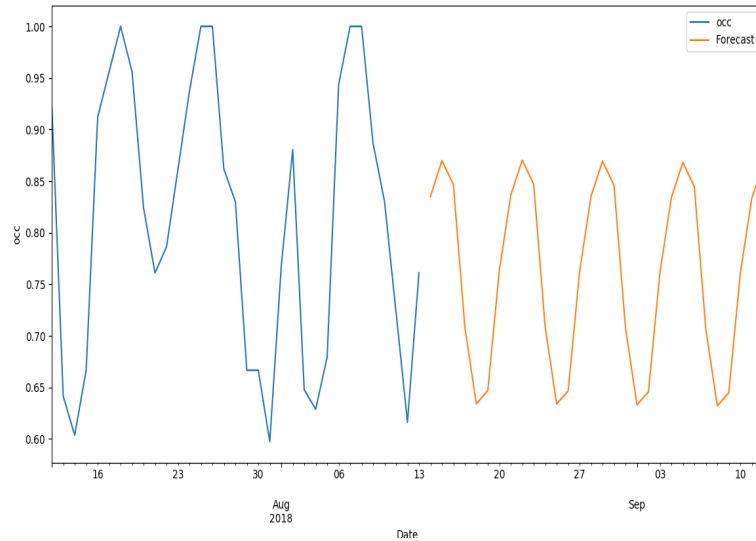


Figura 4.13: Predicciones generadas por el modelo SARIMA

Los resultados fueron almacenados en una matriz que contiene la fecha y el pronóstico de la ocupación para ese día con la finalidad de poder alimentar al modelo de maximización de ingresos.

Dia	Cuartos Ocupados
2018-08-14	130
2018-08-15	134
2018-08-16	129
2018-08-17	123
2018-08-18	114
2018-08-19	112
2018-08-20	114
2018-08-21	128
2018-08-22	132

Tabla 4.10: Pronóstico de cuartos noche ocupados (SARIMA)

Modelo de maximización de ingresos

Una vez pronosticado los cuartos ocupados para cada una de las propiedades de la cadena, se eligió el resultado del modelo de pronóstico de ocupación con el menor *MAPE* y este alimentó un modelo de pricing dinámico que entrega recomendaciones de precios para la tarifa pública tomando en cuenta la demanda pronosticada para cierto día del año.

Típicamente el gerente de la propiedad controla la cantidad de cuartos ofrecidos a diferentes precios, es decir, se asigna cierto número de cuartos a cada nivel de tarifa de tal forma que cuando el inventario asignado al precio mas bajo se agota, se consume el inventario asignado al siguiente nivel de tarifa que será mayor al primer precio ofertado.

A continuación se muestra un esquema donde se ejemplifica la idea mencionada anteriormente:

Deluxe King, 32 rooms city view

\$110	\$110	\$110	\$110	\$110	\$110	\$110	\$110
\$110	\$110	\$110	\$110	\$120	\$120	\$120	\$120
\$120	\$120	\$120	\$120	\$130	\$130	\$130	\$130
\$130	\$130	\$130	\$155	\$155	\$155	\$155	\$155

Figura 4.14: Asignacion de inventario

Como se puede observar, tenemos un inventario de 32 cuartos dividido en 4 niveles de precios:

- 12 cuartos son ofrecidos a un valor de 110 USD
- 8 cuartos son ofrecidos a un valor de 120 USD

CAPÍTULO 4: DISEÑO Y DESARROLLO

- 7 cuartos son ofrecidos a un valor de 130 USD
- 5 cuartos son ofrecidos a un valor de 155 USD

Formulación del modelo

Un problema de maximización de ingresos es básicamente un problema de optimización sujeto a restricciones, en este caso, se quiere maximizar la siguiente función objetivo:

$$\sum_{i=0}^n p_i * o_i$$

En donde i es la índice de la noche, p_i es el precio del cuarto para la i -ésima noche y o_i es la ocupación pronosticada (demanda) para la i -ésima noche al precio p_i . Asumimos también que la función de la ocupación (demanda) está sujeta al precio bajo la siguiente relación:

$$o = o_{nominal} * \left(\frac{p}{p_{nominal}} \right)^e$$

Donde $o_{nominal}$ corresponde a la ocupación pronosticada para una noche dada tomando como base un precio nominal ($p_{nominal}$). Para este caso el $p_{nominal}$ es igual a la tarifa pública promediada a lo largo del año. El valor de e (elasticidad) toma un valor = -2. En otras palabras, p incrementa en un 10 % y la demanda decrece cerca de un 20 %.

Existen las siguientes restricciones también:

$$o_i \leq Capacidad \quad \forall i$$

y

$$p_i > 0 \quad \forall i$$

En otras palabras, la ocupación no puede exceder a la capacidad del hotel y los precios deben ser siempre mayores a cero.

Una vez definidas la función objetivo y las restricciones, se ejecuta el modelo de optimización utilizando el método de *Programación cuadrática secuencial*, un algoritmo útil para resolver problemas de optimización no lineales.

El método de *programación cuadrática secuencial* es un método iterativo para problemas optimización no-lineales con restricciones. Este método es utilizado

CAPÍTULO 4: DISEÑO Y DESARROLLO

en problemas matemáticos en los cuáles la función objetivo y las restricciones son doblemente continuamente diferenciables.

En *programación cuadrática secuencial* se resuelven una serie de subproblemas de optimización de un modelo cuadrático de la función objetivo sujeta a una linearización de las restricciones. Si el problema no cuenta con restricciones, el método se reduce al *método de Newton* para encontrar un punto donde el gradiente de la función objetivo desaparece. Si el problema tiene restricciones de igualdad, el método es equivalente a aplicar el *método de Newton* para encontrar las condiciones óptimas de primer orden o las condiciones de *Karush-Kuhn-Tucker*

Las bases del algoritmo son las siguientes:

Se debe considerar un problema no-lineal de la forma:

$$\min_x f(x)$$

Sujeto a

$$b(x) \geq 0$$

$$c(x) = 0$$

El *Lagrangiano* para este problema es:

$$\mathcal{L}(x, \lambda, \sigma) = f(x) - \lambda^T b(x) - \sigma^T c(x)$$

Donde λ y σ son *multiplicadores lagrangianos*. En la iteración x_k , el algoritmo de programación cuadrática secuencial define la dirección de búsqueda apropiada d_k como solución al subproblema de programación cuadrática:

$$\min_d f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k, \sigma_k) d$$

Sujeto a

$$b(x_k) + \nabla b(x_k)^T d \geq 0$$

$$c(x_k) + \nabla c(x_k)^T d = 0$$

Para fines de este trabajo de investigación, se utilizó la implementación del método *SQP* contenido en el paquete *Scipy* en *Python*

CAPÍTULO 4: DISEÑO Y DESARROLLO

Para poder hacer una asignación dinámica de los precios, evitando tener que correr el modelo conforme la información del hotel es actualizada, se partió el inventario en 4 niveles de disponibilidad (159,120,80 y 40 cuartos disponibles). Para cada nivel de capacidad se resolvió la función objetivo para obtener 4 conjuntos de precios, de esta forma, el gerente de la propiedad puede saber a qué nivel de precio debe subir la tarifa pública de la propiedad conforme las habitaciones disponibles decrecen.

El modelo de maximización de ingresos arroja como resultado una matriz de día vs precio por niveles de inventario. A continuación se muestra un ejemplo:

Día	40.00	80.00	120.00	159.00
07/07/18	1476.07	1106.34	903.32	784.76
08/07/18	1552.06	1229.63	1003.99	872.21
09/07/18	1623.82	1394.27	1138.41	988.99
10/07/18	1667.92	1535.57	1253.79	1089.22
11/07/18	1710.59	1723.36	1407.12	1222.43
12/07/18	1654.45	1487.95	1214.90	1055.44
13/07/18	1531.62	1192.47	973.65	845.85
14/07/18	1498.23	1138.41	929.51	807.51
15/07/18	1598.19	1328.15	1084.43	942.09
16/07/18	1654.45	1487.95	1214.90	1055.44
17/07/18	1674.06	1558.84	1272.79	1105.73
18/07/18	1663.07	1517.89	1239.35	1076.68
19/07/18	1674.06	1558.84	1272.79	1105.73
20/07/18	1632.79	1419.85	1159.31	1007.14
21/07/18	1611.63	1361.61	1111.75	965.83
22/07/18	1589.45	1307.66	1067.70	927.56
23/07/18	1619.08	1381.30	1127.82	979.79

Tabla 4.11: Matriz de asignacion de precio por inventario disponible

De la matriz que arroja el modelo de optimización de precios podemos obtener el precio sugerido para cada nivel de inventario disponible por día pronosticado.

CAPÍTULO 4: DISEÑO Y DESARROLLO

Por ejemplo, para el día 7 de julio de 2018 el modelo sugiere un precio público para habitación sencilla de \$784.76 si el hotel tiene entre 121 y 159 habitaciones disponibles; \$903.32 si se tienen entre 81 y 120 habitaciones disponibles; \$1106.34 entre 41 y 80 habitaciones disponibles; y finalmente un precio de \$1476.07 entre 1 y 40 habitaciones disponibles.

Consideraciones del modelo

Este modelo se formuló considerando los siguientes puntos:

- Se consideran estancias de una sola noche
- No se toman en cuenta reservaciones para grupos ni cancelaciones
- Asumimos paridad de precios entre los canales, es decir, los precios no varían entre los canales de venta
- Se considera solamente un tipo de habitación, el incremento por cambio de habitación es un monto fijo conocido como gap
- Se asume un valor para la elasticidad (-2) que es un valor razonable para la industria

En el siguiente capítulo se discutirá acerca de la interpretación de los distintos modelos propuestos y los resultados obtenidos.

Capítulo 5

Interpretación y Resultados

En el presente capítulo se presentarán a detalle los resultados obtenidos por cada uno de los modelos propuestos para el pronóstico de demanda y el modelo para la maximización de ingresos. También se discutirá la forma en la que se deberán interpretar los resultados de los mismos y así evitar sacar conclusiones erróneas o fuera de contexto.

Interpretación de los modelos de pronóstico de ocupación

Como hemos mencionado en capítulos anteriores, la solución propuesta se compone de dos módulos principales, el primero de ellos se encarga de pronosticar la demanda de cuartos noche, mientras que el segundo módulo toma esa información y calcula los precios por habitación que maximizan el ingreso de la propiedad tomando en cuenta las restricciones definidas para la propiedad. A continuación explicaremos detalladamente los resultados obtenidos en cada uno de los modelos propuestos.

Modelo de regresión lineal generalizada con liga Poisson

Este modelo toma como entrada las curvas de *pickup* para una propiedad en específico, información de su ocupación histórica, líneas de tiempo de tarifa promedio, líneas de tiempo de tarifas publicas propias y de la competencia, etc. Al finalizar el procesamiento de la información se obtiene una matriz que contiene, entre otras variables, el parámetro β_0 y β_1 con el que se puede reconstruir la curva de pickup para un día futuro. De esta manera podemos obtener una buena aproximación de los cuartos que serán vendidos en cierto día y con qué velocidad se realizará esta venta.

A continuación se presenta un extracto de los resultados obtenidos por el modelo de pronóstico de demanda:

Hotel	Dia	AABeta0	AAbeta1	pred.beta0	pred.beta1
Hotel1	2018-01-01	2.335294	-2.48986e-06	3.1595	-0.1227
Hotel1	2018-01-02	2.984073	-2.81809e-06	3.9160	-0.1160
Hotel1	2018-01-03	3.301841	-3.53557e-06	3.8740	-0.1089
Hotel1	2018-01-04	3.782523	-2.29568e-06	4.1597	-0.1028
Hotel1	2018-01-05	3.761697	-1.08967e-06	3.6982	-0.0846

Tabla 5.1: Resultados arrojados por el modelo de predicción de demanda

Para poder reconstruir la curva de *pickup* pronosticada se debe tomar los parámetros *pred.beta0* y *pred.beta1* para evaluar la siguiente expresión:

$$E[y|x] = e^{\beta_0 + \beta_1 x}$$

Dónde:

- $E[y|x]$ = El valor esperado de cuartos noches para un día en específico dado x días de antelación
- β_0 = pred.beta0
- β_1 = pred.beta1
- x = días de antelación

Para construir las curvas de *pickup* pronosticadas se elige un día de la tabla de resultados arrojados por el modelo. Tomaremos los valores de **pred.beta0**

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

y `pred.beta1` para el día elegido. Posteriormente `x` tomará valores de -1 a 25 para representar la curva en un rango de -1 a 25 días de antelación, siendo -1 las llegadas que llegan a la propiedad después de las 23:59 pm del día elegido. Si tomamos el día 2018-01-04 tendríamos una tabla de interpretación como la que se muestra a continuación:

$$pred.beta0 = 4.1597$$

$$pred.beta1 = -0.1028$$

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

Antelación	Cuartos Ocupados (Predicción)
-1	71
0	64
1	58
2	52
3	47
4	42
5	38
6	35
7	31
8	28
9	25
10	23
11	21
12	19
13	17
14	15
15	14
16	12
17	11
18	10
19	9
20	8
21	7
22	7
23	6
24	5
25	5

Tabla 5.2: Interpretación de resultados arrojados por el modelo de pronóstico de demanda

Podemos ver que en esta expresión `pred.beta0` indica cuantos cuartos ocupados se tendrán para el día estudiado, mientras que `pred.beta1` dicta cuantos cuartos incrementan conforme X se acerca a 0 (día de la llegada de los huéspedes al

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

hotel).

Evaluando esta expresión en distintas fechas obtenemos las siguientes curvas de pronóstico de ocupación:

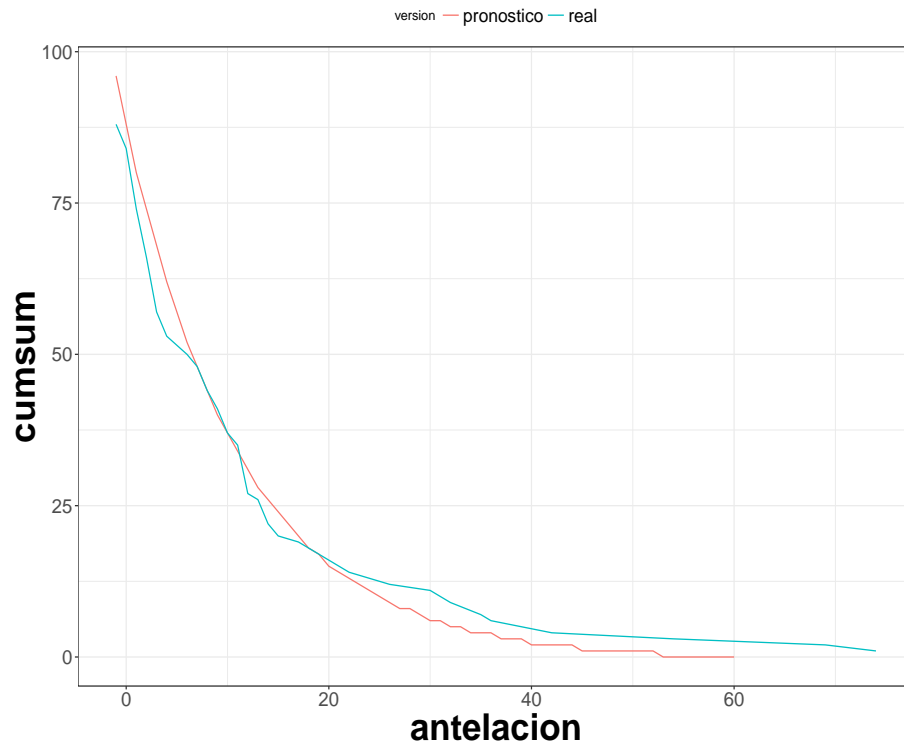


Figura 5.1: Curva de pickup real vs pronosticada (1)

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

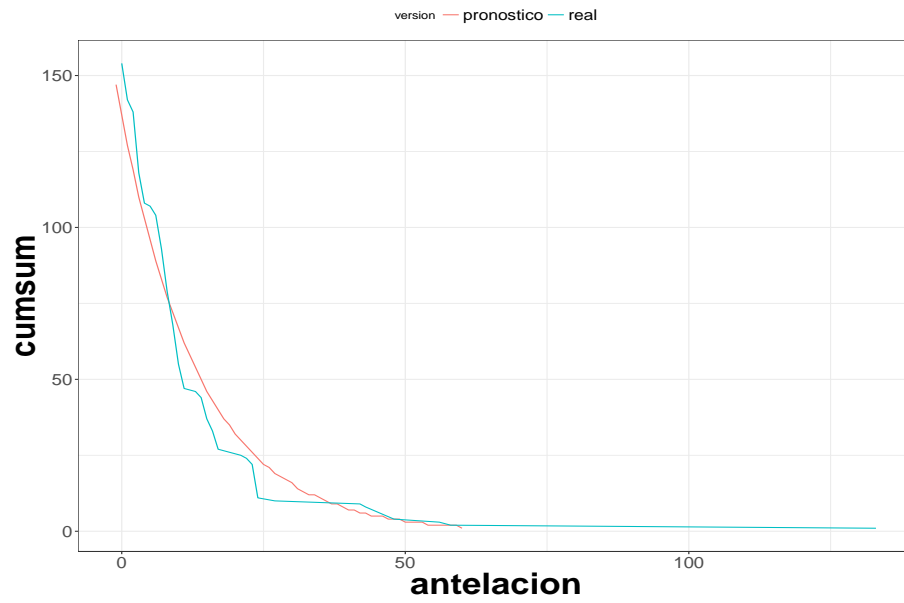


Figura 5.2: Curva de pickup real vs pronosticada (2)

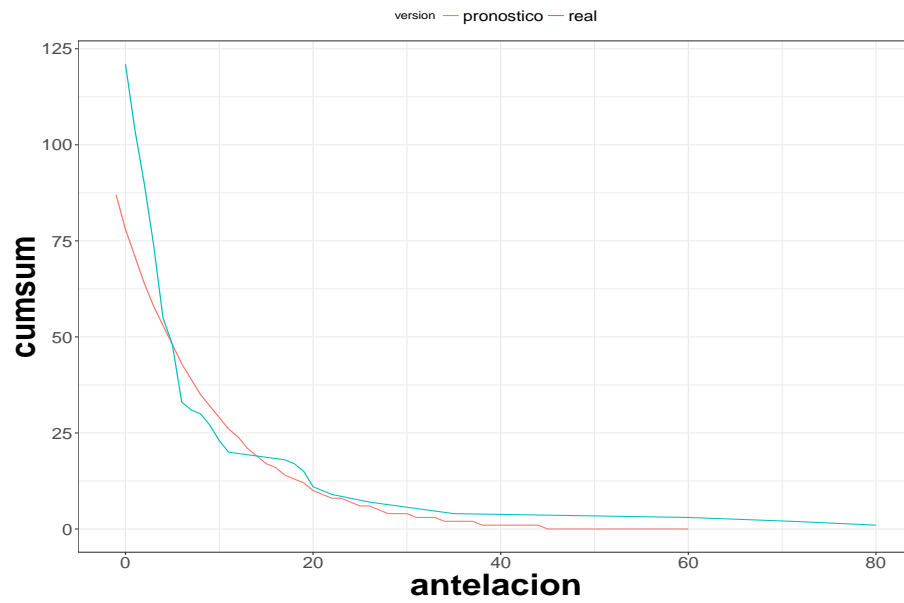


Figura 5.3: Curva de pickup real vs pronosticada (3)

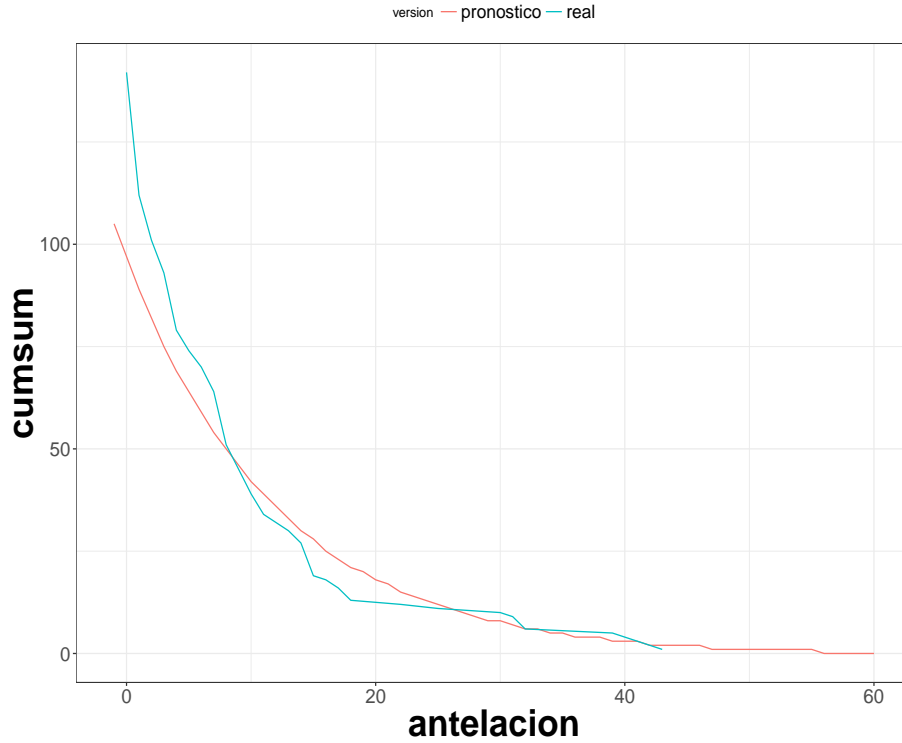


Figura 5.4: Curva de pickup real vs pronosticada (4)

Podemos observar que la curva pronosticada tiene un buen ajuste sobre la curva real en la mayoría de los casos.

Para realizar la validación del modelo se dividió el data set en dos partes, la primera parte se utilizó para el entrenamiento del modelo y la segunda parte para la validación del mismo, de esta forma se pudo comparar el pronóstico de demanda arrojado por el modelo contra la demanda real de la propiedad en 220 días para el año 2018.

Para evaluar el desempeño del modelo se utilizó la medida *MAPE* definida como:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - h_t}{y_t} \right|$$

Dónde:

- n = Número de puntos ajustados

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

- y_t = Cuartos noche vendidos en el tiempo t
- h_t = Venta de cuartos noche pronosticada en el tiempo t

El *MAPE* observado durante la validación del modelo fue de 17.67%.

Modelo de regresión de Ridge

Durante la implementación del modelo basado en una regresión de Ridge, se utilizaron varios valores para el parámetro α (factor de regularización). De los resultados que obtuvimos notamos que el modelo tuvo un mejor desempeño cuando $\alpha = 1 * e^{-15}$. El parámetro α también es conocido como el *coeficiente de regularización*, el cual es usado para mejorar el condicionamiento del problema y reduce la varianza de las predicciones generadas. Mientras más grande es α mayor es la regularización.

En la siguiente gráfica se muestra el efecto que tiene los distintos valores de α sobre las predicciones generadas:

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

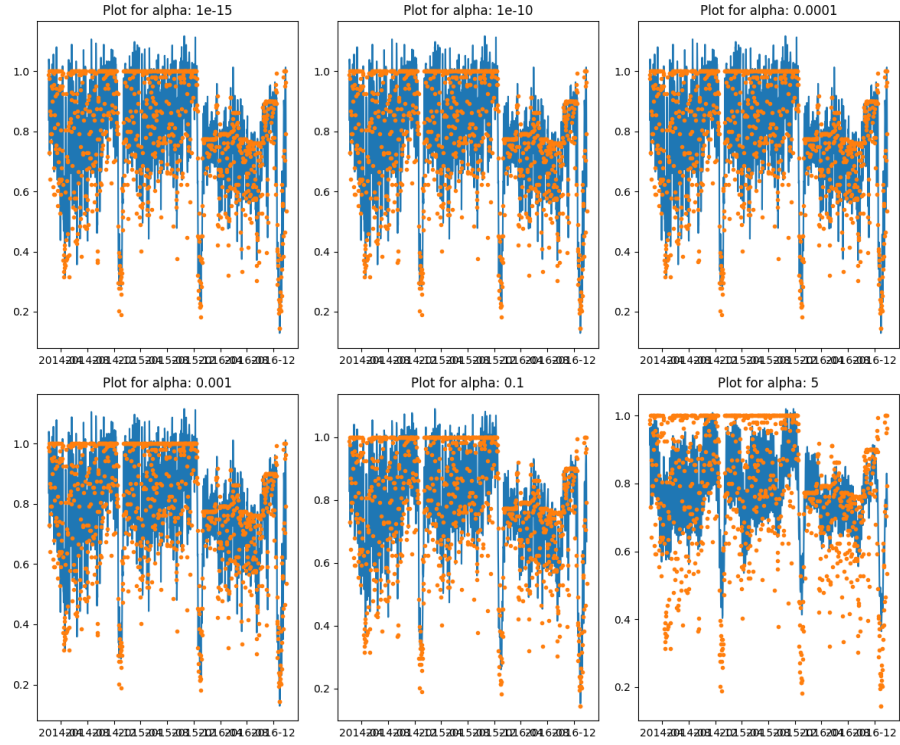


Figura 5.5: Efecto de α sobre las predicciones generadas

Como podemos observar, a mayor valor de α disminuye la varianza de las predicciones pero incrementa el sesgo. A menor valor de α tenemos un comportamiento opuesto.

La regresión de *Ridge* nos regresa una matriz de coeficientes que multiplican a las variables de entrada del modelo, de tal forma que $y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \dots, \beta_n * x_n$. Para generar predicciones para días futuros lo que el modelo internamente hace es tomar la matriz de coeficientes generados por la regresión y los multiplica por la matriz de predictores y como resultado arroja la \hat{y} .

El *MAPE* obtenido para este modelo con el conjunto de datos de validación, utilizando una ventana de rezagos de 400 días para pronósticar una ocupación futura y $\alpha = 1 * e^{-15}$ fue de 15.1815 %.

Modelo de análisis de series temporales SARIMA

Como se mencionó en el capítulo anterior, para poder encontrar el modelo con mejor desempeño se implementó un algoritmo de *grid search*. Dicho algoritmo toma todos los valores definidos para los hiperparámetros del modelo y lo ajusta sobre los datos de entrenamiento. Lo que este algoritmo busca es el conjunto de valores para los hiperparámetros que logran un mejor desempeño del modelo.

En este caso el modelo *SARIMA* con mejor desempeño fue el *SARIMA*(1, 0, 0)x(1, 0, 1, 7) con un $AIC=-3051.784$.

A continuación se presentan las gráficas de diagnóstico obtenidas para dicho modelo. Lo que se busca en estas gráficas es que los residuales no esten correlacionados y se distribuyan normalmente con media = 0. En caso de no tener estos resultados se debe considerar mejorar el modelo ya que podemos incurrir en un alto riesgo de *falsos positivos* en nuestros coeficientes.

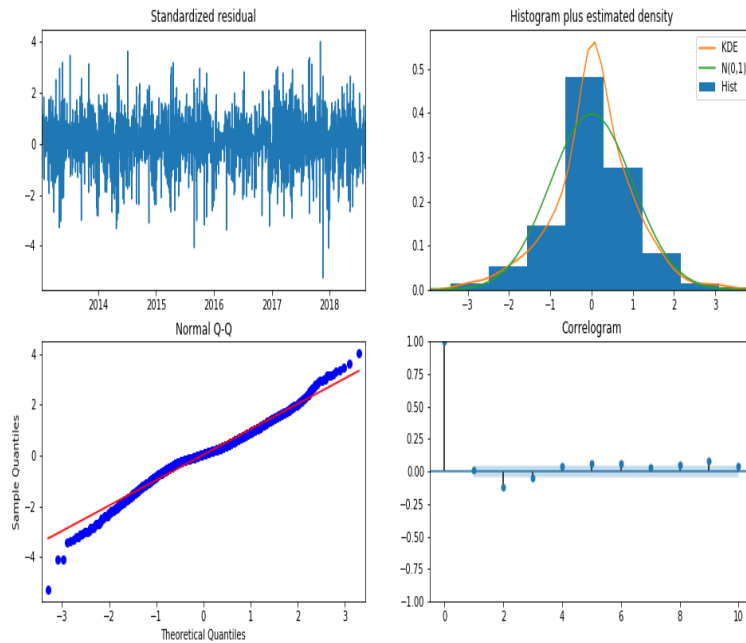


Figura 5.6: Diagnóstico del modelo *SARIMA*(1, 0, 0)x(1, 0, 1, 7)

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

Observando la gráfica presentada podemos confiar en los resultados obtenidos ya que los residuales obtenidos no están correlacionados y se distribuyen normalmente con media = 0. Esto lo podemos observar poniendo atención en los siguientes puntos:

- La gráfica *QQ* se ajusta linealmente a lo largo de la muestra con distribución normal, es decir, todos los puntos caen sobre la línea roja.
- La gráfica *KDE* muestra que los residuales son consistentes con una distribución normal con media = 0.
- El correlograma muestra una baja autocorrelación para los residuales cuando los rezagos son mayores a 0.

Podemos confirmar el desempeño del modelo con el valor del *MAPE* obtenido con el conjunto de datos de validación de 14.8880 % el cual demostró ser el mejor modelo de los tres propuestos.

Modelo de maximización de ingresos

El modelo de recomendaciones de precio toma como entrada la demanda pronosticada por el modelo de pronóstico de demanda. A partir de ahí, se define un problema de maximización sujeto a restricciones y se arroja una matriz que contiene los precios para un tipo de cuarto dependiendo del inventario disponible dentro de la propiedad.

A continuación se muestra un extracto del resultado:

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS

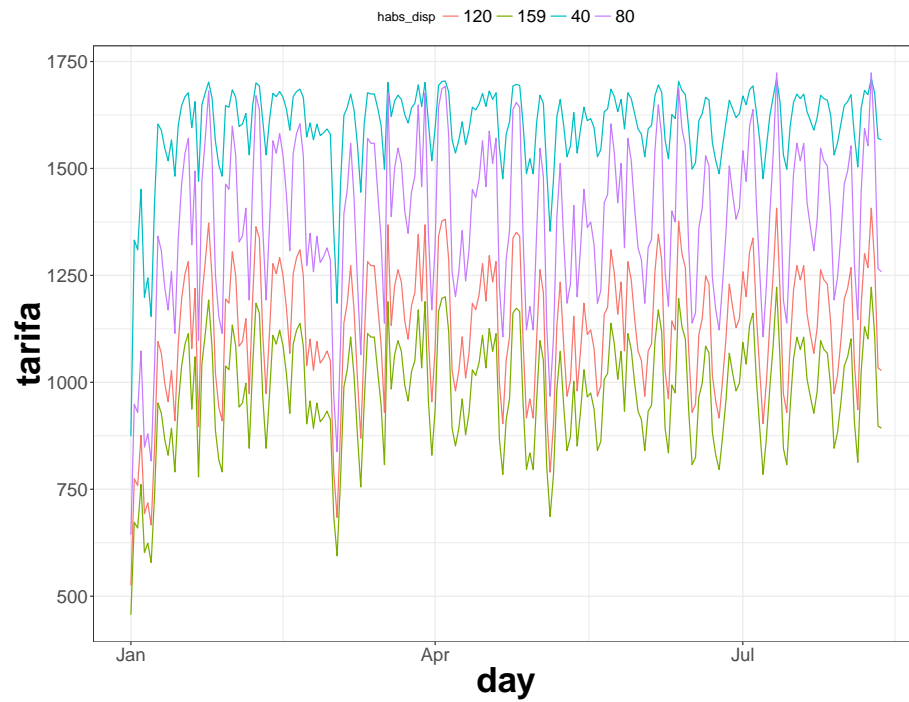
Dia	40.00	80.00	120.00	159.00
07/07/18	1476.07	1106.34	903.32	784.76
08/07/18	1552.06	1229.63	1003.99	872.21
09/07/18	1623.82	1394.27	1138.41	988.99
10/07/18	1667.92	1535.57	1253.79	1089.22
11/07/18	1710.59	1723.36	1407.12	1222.43
12/07/18	1654.45	1487.95	1214.90	1055.44
13/07/18	1531.62	1192.47	973.65	845.85
14/07/18	1498.23	1138.41	929.51	807.51
15/07/18	1598.19	1328.15	1084.43	942.09
16/07/18	1654.45	1487.95	1214.90	1055.44
17/07/18	1674.06	1558.84	1272.79	1105.73
18/07/18	1663.07	1517.89	1239.35	1076.68
19/07/18	1674.06	1558.84	1272.79	1105.73
20/07/18	1632.79	1419.85	1159.31	1007.14
21/07/18	1611.63	1361.61	1111.75	965.83
22/07/18	1589.45	1307.66	1067.70	927.56
23/07/18	1619.08	1381.30	1127.82	979.79

Tabla 5.3: Matriz de asignacion de precio por inventario disponible

La manera en la que se debe interpretar esta matriz es la siguiente: Para el día 7 de Julio de 2018, las habitaciones sencillas de este hotel deben de tener un precio de \$784.76 MXN si el inventario disponible esta entre 121 y 159 habitaciones disponibles. Si se tienen entre 81 y 120 habitaciones disponibles, el precio debe de ser de \$903.32 MXN y así sucesivamente. De esta manera podemos asignar un lote de habitaciones a diferentes rangos de precios siendo las últimas habitaciones disponibles las que tengan el precio más alto.

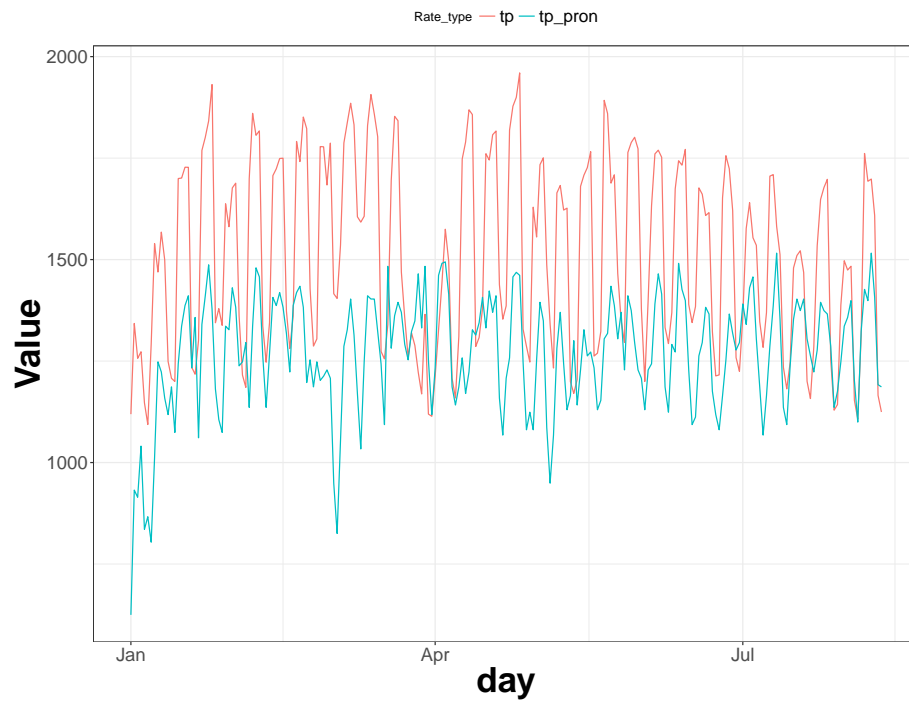
A continuación se presenta una gráfica con los resultados obtenidos por el modelo de maximización de ingresos:

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS



Para medir el desempeño del modelo de maximización de ingresos, se tomaron los resultados y se calcularon las tarifas promedio a partir de los datos obtenidos y se compararon contra las tarifas promedio reportadas por la propiedad para el año 2018.

CAPÍTULO 5: INTERPRETACIÓN Y RESULTADOS



Si obtenemos la diferencia de tarifas promedio, notamos que si se utilizara el resultado del modelo de optimización tendríamos un aumento en los ingresos de aproximadamente 57555.70 MXN. Sin embargo, para poder validar este resultado se propone implementar este modelo dentro de la propiedad y comprobar sus resultados en la operación.

Capítulo 6

Conclusiones

Apéndice A

Implementación de modelo de pronóstico de demanda utilizando un modelo lineal generalizado con liga Poisson

```
library(plyr)
HIS_CEINS<-read_csv("datos/EST_PAS_UNIC_CEINS.csv")
RES_CEINS <- read.csv("datos/CEINS_RESERVAS.csv")
#Juntamos todas las tablas en un solo dataset
#datos <- rbind(HIS_CEINS, RES_CEINS)
datos <- HIS_CEINS
#Modelo de regresion Poisson para obtener
#parametros de modelo predictivo
modelo.1<-function(base,h){
  hotel<-subset(base,prop_code==h)
  dia<-sort(unique(hotel$date_in))
```

APÉNDICE A: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO LINEAL GENERALIZADO CON LIGA POISSON

```

ndia<-length(dia)
beta0<-rep(0,ndia)
beta1<-rep(0,ndia)
variacion<-rep(0,ndia)
for(i in 1:ndia){
  eleccion <- ddply(subset(hotel,as.character(date_in)==
                        as.character(dia[i])),
                    select = c('antelacion','nights')),
                  .(antelacion),summarise,
                  nights = sum(nights))
  eleccion <- eleccion[with(eleccion,order(-antelacion)), ]
  eleccion$nights <- cumsum(eleccion$nights)
  eleccion <- eleccion[eleccion$antelacion<=60,]
  mod <- glm(nights~antelacion, family = "poisson",
             data = eleccion)
  coeficientes <- coef(mod)
  beta0[i] <- coeficientes[1]
  beta1[i]<- coeficientes[2]
  variacion[i] <- sqrt(mod$deviance)
}
yresp <- data.frame(hotel=rep(h,ndia),dia,beta0,beta1,variacion)
return(yresp)
}

# Prepara datos 2017 de entrada del modelo
res <- ddply(datos,.(prop_code,date_create,date_in),
             summarise, nights=sum(nights))
res$prop_code<-toupper(res$prop_code)
res$antelacion <- as.numeric(res$date_in - res$date_create)
res <- subset(res,format(res$date_in, "%Y")=="2017")
yresp.1 <- modelo.1(res,"CEINS")
yresp<-yresp.1

```

APÉNDICE A: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO LINEAL GENERALIZADO CON LIGA POISSON

```
### Preparacion de datos para modelo predictivo (2018)
series2018 <- subset(yresp,format(yresp$dia, "%Y")=="2017")
series2018$dia <- as.Date(series2018$dia)+365
names(series2018) <- c("hotel","dia","AAbeta0",
                      "AAbeta1","AAvariacion")

### Series a la fecha de corte en la extraccion

#Reservas
RES_CEXXX<-RES_CEINS
#
res<-RES_CEXXX
res <- ddply(res,.(prop_code,date_create,date_in),
             summarise, nights=sum(nights))
res$prop_code<-toupper(res$prop_code)
res<-res %>% filter(date_create!="0000-00-00 00:00:00") %>%
  mutate(date_create=as.Date(date_create),
         date_in=as.Date(date_in))
res$antelacion <- as.numeric(res$date_in - res$date_create)
series2018al11ago <- subset(series2018,as.numeric(dia)<
                           as.numeric(as.Date("2018-08-11")))
res2018al11ago <- subset(res,as.numeric(date_in)<
                       as.numeric(as.Date("2018-08-11")))
res2018al11ago <- subset(res2018al11ago,
                       format(res2018al11ago$date_in, "%Y")=="2018")

### Aplicacion del modelo a la fecha de corte por extraccion
param.1 <- modelo.1(res2018al11ago,"CEINS")

param2018 <- param.1
series2018al11ago <- join(series2018al11ago,param2018)

### Generacion de predictores de modelo predictivo
series2018al11ago$diasem <- as.factor(weekdays
```

APÉNDICE A: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO LINEAL GENERALIZADO CON LIGA POISSON

```

                                (series2018al11ago$dia))
series2018al11ago$mes <- as.factor(format(
  series2018al11ago$dia, "%b"))
series2018al11ago <- join(series2018al11ago,TDC)
series2018al11ago$eventos <- rep(0,nrow(series2018al11ago))
series2018al11ago$eventos <- as.factor(ifelse(
  as.numeric(series2018al11ago$dia)
  <=as.numeric(as.Date("2018-08-11")) &
  as.numeric(series2018al11ago$dia)>=
  as.numeric(as.Date("2018-03-27")),
  1,0))
#series2018al11ago <-join(series2018al11ago,eventos)
series2018al11ago$eventos<-as.factor(series2018al11ago$eventos)
PO_PT <- indicadores %>% select(Hotel,fecha,po,pt)

names(PO_PT) <- c("hotel","dia","PO","PT")
series2018al11ago <- join(series2018al11ago,PO_PT)

### Generacion de modelos predictivos por hotel
#CEINS
datos2018al11agoCEINS <- subset(series2018al11ago,hotel=="CEINS")
mod.CEINS.beta0 <- lm(beta0~AAbeta0+diasem+eventos+tdc+PO+PT,
  data=datos2018al11agoCEINS)
datos2018al11agoCEINS$pred.beta0 <-predict(mod.CEINS.beta0,
  type='response')
datos2018al11agoCEINS$AAbeta1[is.na(
  datos2018al11agoCEINS$AAbeta1)]<-0
mod.CEINS.beta1 <- lm(beta1~AAbeta1+diasem+eventos+tdc+PO+PT,
  data=datos2018al11agoCEINS)
datos2018al11agoCEINS$pred.beta1<-predict(mod.CEINS.beta1,
  type='response')

### Aplica los modelos a la serie completa 2018
series2018 <- join(series2018,TDC)

```

APÉNDICE A: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO LINEAL GENERALIZADO CON LIGA POISSON

```
series2018$AAbeta1[is.na(series2018$AAbeta1)]<-0
series2018 <- join(series2018,P0_PT)
series2018$diasem <- as.factor(weekdays(series2018$dia))
series2018$mes <- as.factor(format(series2018$dia, "%b"))

series2018$eventos1 <- ifelse(as.numeric(series2018$dia)<=
                             as.numeric(as.Date("2018-04-04")) &
                             as.numeric(series2018$dia)>=
                             as.numeric(as.Date("2018-03-27")),
                             1,0)

series2018$eventos <- series2018$eventos1
series2018$eventos <- as.factor(series2018$eventos)
series2018$pred.beta0 <- ifelse(series2018$hotel=="CEINS",
                                predict(mod.CEINS.beta0,
                                         newdata=series2018,
                                         type="response"),0)
series2018$pred.beta1 <- ifelse(series2018$hotel=="CEINS",
                                predict(mod.CEINS.beta1,
                                         newdata=series2018,
                                         type="response"),0)
```

Apéndice B

Implementación de modelo de pronóstico de demanda utilizando una regresión de Ridge

```
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 12, 10

indicadores = pd.read_excel("data/indicadores_CEINS.xls")
indicadores = indicadores[['fecha','occ']]

def regresa_ds(data>window=1,dropnan=True):
    for x in range(1>window+1):
        column = 't-'+str(window-x)
```


APÉNDICE B: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UNA REGRESIÓN DE RIDGE

```
        data[column]=data['occ'].shift(window-x)
    if dropnan:
        data.dropna(inplace=True)
    data=data.drop('occ',1)
    return data

lags = 400
indicadores = regresa_ds(indicadores,lags,1)

import datetime
indicadores['fecha']=pd.to_datetime(indicadores['fecha'],format='%Y-%m-%d')
indicadores['dow']=indicadores['fecha'].dt.strftime('%A')
indicadores['year']=indicadores['fecha'].dt.strftime('%Y')
indicadores['month']=indicadores['fecha'].dt.strftime('%m')
indicadores['day']=indicadores['fecha'].dt.strftime('%d')
indicadores['y']=indicadores['t-0']
df_dow = pd.get_dummies(indicadores['dow'])
indicadores = pd.concat([indicadores,df_dow],axis=1)
indicadores=indicadores.drop('dow',1)
indicadores=indicadores.drop('t-0',1)
indicadores.to_csv('data/indicadores.csv', sep=',')

train_size = int(len(indicadores)*0.66)
train, test = indicadores[0:train_size], indicadores[train_size:len(indicadores)]

from sklearn.linear_model import Ridge
def ridge_regression(data, predictors, alpha, models_to_plot={}):
    #Fit the model
    ridgereg = Ridge(alpha=alpha,normalize=True)
    ridgereg.fit(data[predictors],data['y'])
    y_pred = ridgereg.predict(data[predictors])

    #Check if a plot is to be made for the entered alpha
    if alpha in models_to_plot:
```

APÉNDICE B: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UNA REGRESIÓN DE RIDGE

```

plt.subplot(models_to_plot[alpha])
plt.tight_layout()
plt.plot(data['fecha'],y_pred)
plt.plot(data['fecha'],data['y'],'.')
plt.title('Plot for alpha: %.3g'%alpha)

#Return the result in pre-defined format
rss = sum((y_pred-data['y'])**2)
ret = [rss]
ret.extend([ridgereg.intercept_])
ret.extend(ridgereg.coef_)
return ret, y_pred

predictors=['t-1']
predictors.extend(['t-%d'%i for i in range(2,lags)])
predictors = predictors +
['year','month','day','Monday','Tuesday','Wednesday',
'Thursday','Friday','Saturday','Sunday']
alpha_ridge = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, .1, 5, 10, 20]

col = ['rss','intercept'] + ['coef_x_%d'%i for i in range(1,len(predictors)+1)]
ind = ['alpha_%.2g'%alpha_ridge[i] for i in range(0,10)]
coef_matrix_ridge = pd.DataFrame(index=ind, columns=col)
y_pred = [None] * 10
models_to_plot = {1e-15:231, 1e-10:232, 1e-4:233, 1e-3:234, .1:235, 5:236}
for i in range(10):
    coef_matrix_ridge.iloc[i,],y_pred[i] = ridge_regression(train,
predictors, alpha_ridge[i], models_to_plot)

alpha = 1e-15
predictors=['t-1']
predictors.extend(['t-%d'%i for i in range(2,lags)])
predictors = predictors + ['year','month','day','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
ridgereg = Ridge(alpha=alpha,normalize=True)

```

APÉNDICE B: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UNA REGRESIÓN DE RIDGE

```
ridgereg.fit(train[predictors],train['y'])
y_pred_test = ridgereg.predict(test[predictors])
MAPE = (sum(abs((test['y']-y_pred_test)/(test['y']))) / len(test['y'])) * 100
print(MAPE)

import datetime
from datetime import timedelta
pronostico_raw = pd.read_csv("data/Pronostico_CEINS.csv")
pronostico_raw = pronostico_raw[['fecha','occ']]
pronostico_raw['fecha']=pd.to_datetime(pronostico_raw['fecha'],format='%Y-%m-%d')

fecha_pron = datetime.date(2018, 8,14)
fecha_inicio = fecha_pron

predictors=['t-1']
predictors.extend(['t-%d'%i for i in range(2,lags)])
predictors = predictors + ['year','month','day','Monday',
'Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']

window=40
i=0

while i < window:
    pronostico = pronostico_raw.copy()
    pronostico = regresa_ds(pronostico,lags,1)
    pronostico['fecha']=pd.to_datetime(pronostico['fecha'],format='%Y-%m-%d')
    pronostico['dow']=pronostico['fecha'].dt.strftime('%A')
    pronostico['year']=pronostico['fecha'].dt.strftime('%Y')
    pronostico['month']=pronostico['fecha'].dt.strftime('%m')
    pronostico['day']=pronostico['fecha'].dt.strftime('%d')
    pronostico['y']=pronostico['t-0']
    df_dow = pd.get_dummies(pronostico['dow'])
    pronostico = pd.concat([pronostico,df_dow],axis=1)
```

APÉNDICE B: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UNA REGRESIÓN DE RIDGE

```
pronostico=pronostico.drop('dow',1)
pronostico=pronostico.drop('t-0',1)
pronostico = pronostico[(pronostico.fecha >=fecha_pron)]
aux=pronostico.loc[pronostico['fecha'] == fecha_pron]
aux=aux[predictors]
y_pred_test = ridgereg.predict(aux)
pronostico_raw.loc[pronostico_raw['fecha'] ==
fecha_pron,'occ']=y_pred_test
fecha_pron=fecha_pron+timedelta(days=1)
i = i+1

fecha_fin = fecha_pron-timedelta(days=1)

pronostico_raw = pronostico_raw[(pronostico_raw.fecha >=fecha_inicio)&
(pronostico_raw.fecha <=fecha_fin)]
pronostico_raw['Hab_Ocup']=round(pronostico_raw['occ']*159)
pronostico_raw.to_csv('data/pronostico.csv', sep=',')
```

Apéndice C

Implementación de modelo de pronóstico de demanda utilizando un modelo de análisis de series de tiempo SARIMA

```
import warnings
import itertools
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 12, 10

df = pd.read_csv("data/Indicadores_CEINS.csv")
```

APÉNDICE C: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO DE ANÁLISIS DE SERIES DE TIEMPO SARIMA

```
ocupacion = df.loc[df['fecha'] >= '2013-01-01']
ocupacion['fecha'].min(), ocupacion['fecha'].max()
ocupacion.head()

cols = ['Hotel', 'habs_disp', 'tp', 'po', 'pt', 'te']
ocupacion = ocupacion.drop(cols, axis=1)
ocupacion = ocupacion.sort_values('fecha')
ocupacion.isnull().sum()
ocupacion.head()

ocupacion = ocupacion.groupby('fecha')['occ'].sum().reset_index()

ocupacion['fecha']=pd.to_datetime(ocupacion['fecha'])
ocupacion = ocupacion.set_index('fecha')

y = ocupacion

p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 60) for x in list(itertools.product(p, d, q))]
print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))

lowest_aic = None
lowest_parm = None
lowest_param_seasonal = None
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(y,
            order=param,
```

APÉNDICE C: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO DE ANÁLISIS DE SERIES DE TIEMPO SARIMA

```

        seasonal_order=param_seasonal,
        enforce_stationarity=False,
        enforce_invertibility=False)
    results = mod.fit()
    current_aic = results.aic
    if (lowest_aic == None):
        lowest_aic = results.aic
    if (current_aic <= lowest_aic):
        lowest_aic = current_aic
        lowest_parm = param
        lowest_param_seasonal = param_seasonal
    print('SARIMA{ }x{ } - AIC:{ }'.format(param, param_seasonal, results.aic))
    #print('ARIMA{ }x{ }7 - AIC:{ }'.format(param, param_seasonal, results.aic))
except:
    continue

print('The best model is:
SARIMA{ }x{ } - AIC:{ }'.format(lowest_parm,
lowest_param_seasonal, lowest_aic))

mod = sm.tsa.statespace.SARIMAX(y,
                                order=lowest_parm,
                                seasonal_order=lowest_param_seasonal,
                                enforce_stationarity=False,
                                enforce_invertibility=False)

results = mod.fit()
print(results.summary())

pred = results.get_prediction(start=pd.to_datetime('2017-01-01'), dynamic=False)
pred_ci = pred.conf_int()
ax = y['2014:'].plot(label='observed')
pred.predicted_mean.plot(ax=ax,
label='Pronóstico un paso adelante',
alpha=.7, figsize=(14, 7))

```

APÉNDICE C: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO DE ANÁLISIS DE SERIES DE TIEMPO SARIMA

```
ax.fill_between(pred_ci.index,
                 pred_ci.iloc[:, 0],
                 pred_ci.iloc[:, 1], color='k', alpha=.2)
ax.set_xlabel('Date')
ax.set_ylabel('Ocupación')
plt.legend()
plt.savefig('Figures/ARIMA_pred.png')
plt.show()

real=y['2017':]
MAPE = (sum(abs((real['occ']-pred.predicted_mean)))/
real['occ'])/len(real['occ']))*100
print(MAPE)

y2=y[2018:]
pred_uc = results.get_forecast(steps=30)
pred_ci = pred_uc.conf_int()
ax = y2.plot(label='observed', figsize=(14, 7))
pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
ax.set_xlabel('Date')
ax.set_ylabel('occ')
plt.legend()
plt.savefig('Figures/ArimaPred.png',dpi=100)
plt.show()

arima_pred = pred_uc.predicted_mean
arima_pred.to_csv('data/arima_pred.csv', sep=',')
```


APÉNDICE C: IMPLEMENTACIÓN DE MODELO DE PRONÓSTICO DE DEMANDA
UTILIZANDO UN MODELO DE ANÁLISIS DE SERIES DE TIEMPO SARIMA

Apéndice D

Implementación de modelo de optimización de ingresos

```
import pandas as pd
import numpy as np
import scipy
import math as mt
import pickle as pkl
import os
import psycpg2
import matplotlib
import matplotlib.pyplot as plt
from numpy.random import normal
import calendar
from scipy.optimize import curve_fit
plt.rcParams['figure.figsize'] = (16,8)
import warnings
warnings.filterwarnings('ignore')
import plotly.plotly as py
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly.offline import download_plotlyjs,
```

APÉNDICE D: IMPLEMENTACIÓN DE MODELO DE OPTIMIZACIÓN DE INGRESOS

```
init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
from datetime import datetime
from datetime import timedelta

# Cargamos los resultados de la regresion poisson para pronosticar ocupacion
data = pd.read_csv("data/forecasted_demand.csv")
date_start = min(data.dia.values)
date_end = max(data.dia.values)
date_end=datetime.strptime(date_end, '%Y-%m-%d')
date_end= date_end + timedelta(days=1)
nb_days = (pd.to_datetime(date_end) - pd.to_datetime(date_start)).days
forecasted_demand = data.cn_predic.values
forecasted_demand=pd.DataFrame(forecasted_demand,
                               index=pd.date_range(start=date_start,
                                                    end=date_end,closed='left'),
                               columns=['occupancy'])
forecasted_demand.occupancy=data.cn_predic.values

def demand_price_elasticity(price, nominal_demand, elasticity=-2.0, nominal_price=120.0):

import scipy.optimize as optimize

def objective(p_t, nominal_demand=np.array([50,40,30,20]),
              elasticity=-2.0, nominal_price=1200.0):
    return (-1.0 * np.sum( p_t * demand_price_elasticity(p_t,
                                                            nominal_demand=nominal_demand,
                                                            elasticity=elasticity,
                                                            nominal_price=nominal_price) )) / 100

def constraint_1(p_t):
    return p_t
```

APÉNDICE D: IMPLEMENTACIÓN DE MODELO DE OPTIMIZACIÓN DE INGRESOS

```
def constraint_2(p_t, capacity=20, forecasted_demand=35.0,
                elasticity=-2.0, nominal_price=1200.0):
    return capacity - demand_price_elasticity(p_t,
                                                nominal_demand=forecasted_demand,
                                                elasticity=elasticity,
                                                nominal_price=nominal_price)

capacities = [20.0, 40.0, 60.0, 80.0]

optimization_results = {}
for capacity in capacities:
    nominal_price = 1200.0
    nominal_demand = forecasted_demand['occupancy'].values
    elasticity = -2.0

    p_start = 1250.0 * np.ones(len(nominal_demand))

    bounds = tuple((100.0, 2000.0) for p in p_start)

    constraints = ({'type': 'ineq', 'fun': lambda x: constraint_1(x)},
                  {'type': 'ineq', 'fun': lambda x, capacity=capacity,
                    forecasted_demand=nominal_demand,
                    elasticity=elasticity,
                    nominal_price=
                    nominal_price: constraint_2(x,
                    capacity=capacity,
                    forecasted_demand=nominal_demand,
                    nominal_price=nominal_price)})

    opt_results = optimize.minimize(objective, p_start,
                                    args=(nominal_demand,
                                          elasticity,
                                          nominal_price),
                                    method='SLSQP', bounds=bounds,
```

APÉNDICE D: IMPLEMENTACIÓN DE MODELO DE OPTIMIZACIÓN DE INGRESOS

```
constraints=constraints)

optimization_results[capacity] = opt_results

time_array = np.linspace(1,len(nominal_demand),len(nominal_demand))
rate_df = pd.DataFrame(index=time_array)

for capacity in optimization_results.keys():
    rate_df = pd.concat([rate_df,
                        pd.DataFrame(optimization_results[capacity]['x'],
                                    columns=['{}'.format(capacity)],
                                    index=time_array)],axis=1)

rate_df.index.name = 'Day'
datelist = pd.date_range(start=date_start, end=date_end, closed='left').tolist()
rate_df.index = [ x.date() for x in datelist]

rate_df.to_csv('data/rates.csv')
```

Bibliografía

- [1] El-Gayar N, Hendawi AT, Zakhary A, El-Shishiny H. *A proposed decision support model for hotel room revenue management*. ICGST Int J Artif Intell Mach Learn 2008.
- [2] Heba Abdel Aziz, Mohamed Saleh, Mohamed H. Rasmy, Hisham ElShishiny *Dynamic room pricing model for hotel revenue management systems*. Faculty of Computers and Information, Cairo University 2011.
- [3] Gabriel Bitran, Rene Caldentey *An Overview of Pricing Models for Revenue Management*. Sloan School of Management, MIT, Cambridge, Massachusetts 2011.
- [4] McGill, J. and van Ryzin, G *Revenue management: Research overview and Prospects*. Transportation Science 1999.
- [5] Cross, R. *Revenue Management: Hard-Core Tactics for Market Domination*. New York, NY: Broadway Books 1997.
- [6] Marriott, Jr., J. and Cross, R. *Room at the revenue inn. In Book of Management Wisdom: Classic Writings by Legendary Managers*. Peter Krass, New York, NY: Wiley 2000.
- [7] Weatherford, L. R., Kimes, S. E. *A comparison of forecasting methods for hotel revenue management*. Cornell University 2003.
- [8] M. J. Beckmann and F. Bobkowski *Airline Demand: An Analysis of Some Frequency Distributions*. Naval Res. Logistics 1958.

BIBLIOGRAFÍA

- [9] Athanasius Zakharya, Amir F. Atiyab, Hisham El-Shishiny and Neamat E. Gayara *Forecasting hotel arrivals and occupancy using Monte Carlo simulation* Faculty of Computers and Information, Cairo University 2009.
- [10] Lee, A. *Airline Reservations Forecasting* Erehwon, NC: Prentice-Hall 1990.
- [11] Flides, R, Ord, K. *Forecasting competitions: their role in improving forecasting practice and research* Oxford: Blackwell 2002.
- [12] Ming Tse, Tony S., Tung Poon Yiu *Analyzing the use of an advance booking curve in forecasting hotel reservations* Taylor and Francis 2015.
- [13] Wickham, R. R. *Evaluation of forecasting techniques for short term demand of air transportation* MIT Thesis: Flight Transportation Lab 1995.
- [14] Weatherford, L. R. *Forecasting issues in revenue management* Springs INFORMS Conference 1998.
- [15] Caicedo-Torres, W., Payares, F. *A machine learning model for occupancy rates and demand forecasting in the hospitality industry* Universidad Tecnológica de Bolívar 2016.
- [16] Tierney, Heather L.R, Bing Pan *A poisson regresion examination of the relationship between website traffic and search engine queries* Netnomics 2012.
- [17] Pan,B., Yang, Y *Forecasting destination weekly hotel occupancy with big data* Journal of Travel Research 2017.
- [18] Aziz, et al. *Dynamic Pricing for Hotel Revenue Management Using Price Multipliers* Faculty of Computers and Information, Cairo University 2011.
- [19] Reaside,R., Windle D. *Quantitive aspects of yield management* London: Cassell 2000.
- [20] Kin Keng Lai, Wan Lung Ng *A stochastic approach to hotel revenue optimization* University of Hong Kong 2003.