Grado en Estadística

TRABAJO FIN DE GRADO

Técnicas estadísticas en predicción de la demanda en el sector comercio

Tutor: José Luis Pino Mejías

Alumna: Andrea Mesta Carmona

Sevilla, Junio de 2021

Índice general

Resumen

En la actualidad, es imprescindible para cualquier organización contar con un sistema efectivo de análisis de datos, ya que el conocer y anticiparse a la demanda de productos le supone incrementar la probabilidad de alcanzar sus objetivos.
El presente trabajo se centra en el estudio y aplicación de técnicas estadísticas para datos de conteo y de Machine Learning, con el fin de conseguir una predicción de la demanda de productos. Para ello, se pretende dar una visión global de algunas de las distintas técnicas y algoritmos existentes para el modelado de datos de recuento. Además, al tratar este tipo de datos, se presenta el problema del exceso de ceros, estudiándose modelos que, siguiendo distintas estrategias, abordan dicho problema.

Abstract

Currently, it's essential for every company to have an effective system of data analysing. In order to get to know and anticipate to products demand, you can guarantee its goals to succeed. This end of degree project focuses on the research and application of statistical techniques for count data and Machine Learning to predict demand. The aim is to give a global vision of some of the techniques and algorithms available for data modeling. In addition, by working out this type of data, there could be a problem with the excess of zeros. The project will treat different models that following diverse strategies could solve that issue.

Introducción

Conocer la previsión de las ventas y la demanda ayuda a los responsables de las empresas a tomar mejores decisiones en las actividades de planificación, producción, suministro y marketing, por lo que es crucial para satisfacer sus necesidades. Por ello, la Analítica Empresarial se ha convertido en un apoyo indispensable para cualquier tipo de actividad comercial, ya que una previsión de ventas y demanda precisas es esencial para los Procedimientos de Ventas y Operaciones (S&OP). Un área en la que la previsión de la demanda es fundamental es la de los Bienes de Consumo de Alta Rotación (FMCG), en la que disponer del adecuado nivel de inventario de cada producto es un factor clave para la supervivencia de las empresas minoristas.

Para que los Procedimientos de Ventas y Operaciones de una organización proporcionen previsiones fiables, es necesario tener una previsión de la demanda adecuada, precisa y eficiente; esto no es fácil de conseguir debido a que el patrón de la demanda de los clientes dependerá, entre otros aspectos, del clima, de los periodos vacacionales, de los patrones estacionales, de las situaciones económicas,...
Además, se dispone de una cantidad de datos inmensa que resulta imposible de ser analizada y aún menos de obtener predicciones, sin utilizar software específico. En este trabajo se presenta un análisis para la predicción de la demanda empleando técnicas estadísticas y algoritmos de Machine Learning.

Como objetivos propuestos se tienen los siguientes:

- En primer lugar, se pretende proporcionar una visión global acerca de porqué es tan importante la modelización de ventas y la obtención de predicciones. Para ello, se introduce el concepto de Inteligencia de negocios.
- En segundo lugar, se pretende explicar, de forma teórica, técnicas de modelado y predicción aplicables a datos de conteo, ya que a lo que se aspira es a la predicción de ventas; y conocer sus etapas de realización. Para ello, se estudian tanto técnicas estadísticas como de Machine Learning.
- Por último, se aplican todas las técnicas citadas a un producto de una base de datos de ventas anonimizados de una cadena de supermercados, con el fin de modelar y obtener predicciones.

Para llevar a cabo dichos objetivos, el presente trabajo se ha estructurado en dos capítulos. En el primero, se habla sobre la Inteligencia de negocios, que incluye un conjunto de técnicas, estrategias y métodos capaces de recopilar, almacenar y analizar los datos de las actividades u operaciones de un negocio, con el fin de obtener la mayor cantidad de información y conocimiento posibles y así poder optimizar su rendimiento. Más adelante, se detallan teóricamente algunas de las técnicas que pueden emplearse para conseguir la previsión de la demanda.

Las técnicas a utilizar, pueden clasificarse a partir de las hipótesis realizadas sobre los datos en técnicas algorítmicas (Machine Learning) y técnicas de aprendizaje estadístico. Las técnicas estadísticas descritas se centran en la modelización y predicción de ventas como un modelo de regresión de Poisson y como un modelo de regresión Binomial negativa, al ser dos de los métodos más útiles para modelar datos de conteo; y como modelo de serie temporal. Además, también se estudiarán dos de los problemas más habituales que se dan en los modelos de regresión, la sobredispersión e infradispersión de los datos y las modificaciones producidas por los ceros. Para ello se exponen los modelos inflados y los modelos en dos partes.
Las técnicas algorítmicas detalladas corresponden al aprendizaje supervisado, debido a que se trabaja con un conjunto de datos dados, dividido en conjunto de entrenamiento y conjunto test.

El segundo y último capítulo, está dedicado a un caso práctico, donde se aplican las técnicas explicadas anteriormente a un producto de una base de datos. Esto se lleva a cabo mediante el software estadístico R.

A continuación, se extraen conclusiones generales acerca de los objetivos descritos y sobre las técnicas empleadas, realizando, además, un resumen de los valores de las predicciones en cada modelo para una semana determinada.

Finalmente, se realiza una recopilación bibliográfica sobre aquellos textos (libros, artículos, manuales,. . .) usados para la realización del trabajo.

Capítulo 1

Técnicas para la previsión de la demanda

1.1. Inteligencia de negocios

La mayoría de las organizaciones generan, almacenan y modifican una gran cantidad de datos de cualquier actividad que se registre en ellas, es decir, tienen almacenada su historia en una base datos a través de aplicaciones de gestión de datos cada vez más complicadas y obsoletas. De estas bases de datos se puede extraer conocimiento útil y novedoso y así contribuir a la ayuda de la toma de decisiones. El poder competitivo que puede tener una empresa se basa en la calidad y cantidad de información que sea capaz de usar para mejorar su eficiencia. Gracias a la implementación de la Inteligencia de negocios (BI) se cuenta con una visión integral de todos los datos de la organización, con el objetivo de poder generar el conocimiento necesario tanto para escoger la alternativa que sea más apropiada para el éxito de la empresa y adaptarse rápidamente a los cambios de la demanda, como para reducir posibles malas decisiones. Es decir, BI ofrece a las empresas un sistema de apoyo para conseguir sus metas.

La inteligencia de negocios consta de un conjunto de técnicas, metodologías y aplicaciones o herramientas que ayudan a recopilar, procesar, analizar los datos, exprimir toda la información que esconden éstos, así como descubrir posibles tendencias y patrones, y generar el conocimiento que se toma como referencia para la toma de decisiones.

Seguidamente se detallan algunas de la herramientas que existen dentro de BI.

- **Minería de datos:** disciplina complementaria a la Inteligencia de Negocios que se encarga de recoger, depurar, modelar grandes volúmenes de datos y encontrar ciertos patrones o tendencias previamente desconocidos en ellos. Éstos pueden ser cargardos posteriormente en el Data Warehouse.
- **Data Warehouse:** proceso de almacenar datos de distintas fuentes de información. Éstos deben estar depurados, transformados y bien estructurados para poder almacenarlos en una base de datos, con el fin de depurar información, la cual es utilizada posteriormente para el análisis de negocio.
- **OLAP:** Procesamiento Analítico en Línea (On-Line Analytical Processing) que se basa en el análisis de grandes cantidades de datos, organizándolos en subconjuntos

de datos con una estructura multidimensional y así agilizar la consulta de éstos, proporcionando capacidad de cálculo, consulta o pronóstico.

- **Balanced Scorecard:** o Cuadro de Mando Integral trata de alinear y conseguir un correcto equilibrio entre los elementos de la estrategia global, como el propósito u objetivo que se tiene; y los elementos operativos de la empresa y seguir su evolución. Es decir, según una serie de indicadores se evalúa el desempeño de cada una de las iniciativas que toma la empresa para conseguir resultados satisfactorios.

BI no solo es capaz de analizar los datos existentes, para aportar una visión actual del negocio, o de describir escenarios pasados, para conocer su historia, sino también es capaz de realizar predicciones futuras. Existen varios modelos, basados en estadísticas inferenciales, que ayudan a obtener esta información:

- **Predictivo:** analiza la información actual e histórica y realiza predicciones acerca del futuro.
- **Descriptivo:** clasifica en grupos según la relación entre clientes, es decir, según la relación del conjunto de información.
- **Decisión:** describe todas las posibles combinaciones que pueden tener las variables (alternativas) y así poder predecir los resultados de estas decisiones.

1.2. Técnicas estadísticas

1.2.1. Regresión (Modelo Lineal General)

La modelización estadística es un modelo matemático capaz de describir la relación de dependencia entre una variable respuesta u objetivo (dependiente) y una o más variables explicativas (independientes). Este modelo permite el estudio, el análisis y la comprensión de la variable respuesta a partir de las variables explicativas, con el objetivo de explicar su comportamiento y poder así predecir su valor futuro, o simplemente conocerla.

La demanda de un producto depende de ciertos factores como el precio, la temporada del año, la renta del país o las preferencias del consumidor, por lo que varias variables independientes van a influir en dicha variable objetivo.
El modelo de regresión lineal múltiple permite conocer el efecto simultáneo que tienen varias variables independientes sobre la dependiente, modelizando la media de esta variable respuesta, cuya distribución debe seguir una ley normal y presentar homocedasticidad, es decir, igual varianzas. Por tanto, para la formulación del modelo se debe tener en cuenta la naturaleza de las variables que están involucradas y la relación entre ellas.
Cuando se trata de datos de conteo, ya que lo que se quiere es predecir la demanda, se utilizan modelos para datos de conteo. Éstos son una generalización de los modelos lineales, puesto que para poder aplicar un modelo de regresión lineal, la variable objetivo debe ser cuantitativa y continua y, el número de ventas es cuantitativo pero discreto (número de productos demandados en un periodo de tiempo). Además, los modelos lineales tienen como susposición central que la varianza de la variable objetivo debe ser constante. En los datos de recuento, cuando la variable respuesta es un número entero y, a menudo, hay muchos ceros en el marco de datos, la varianza puede aumentar linealmente con la media (Crawley, 2012). Los modelos lineales generalizados consideran una función de la media de modo que esta si dependa linealmente de las variables explicativas.

Las hipótesis a considerar en un MLG serían:

- Los MLG consideran otras distribuciones para sus errores, no solamente la distribución normal, única aceptada por el modelo lineal clásico. Además de no cumplir el supuesto de linealidad.
- En ambos modelos, las observaciones se suponen idependientes. Sin embargo, en el MLG no se requiere la homogeneidad de varianzas, es decir, el modelo no tiene porqué ser homocedástico.
- La esperanza de la variable objetivo, μ_i, a diferencia de los modelos lineales, ya no está relacionada directamente con las variables explicativas, sino que se hace a través de una función link:

$$E(Y) = \mu = g^{-1}(X^t\beta)$$

donde,

– Y es la variable objetivo formada por un conjunto de observaciones, y_i, donde i=1,. . . ,N,

– E(Y) es el valor esperado de Y,

– $(X^t\beta)$ es el predictor lineal, combinación lineal de parámetros desconocidos, que posteriormente se debe estimar,

– $\beta = (\beta_0, \beta_1, ..., \beta_p)$ donde cada β representa el efecto de la correspondiente variable predictora,

– g es la función enlace,

– X es una matriz formada por todos los posibles valores que toman las variables explicativas.

En un modelo lineal general se distinguen tres componentes básicas:

- **Componente aleatoria:** identifica a la variable respuesta y a su función de probabilidad, la cual pertenece a la familia exponencial.
  Este trabajo se centra en las distribuciones de Poisson y Binomial negativa, ya que estos modelos de regresión permiten analizar variables de conteo frente a las demás variables explicativas.

Para que la distribución de la variable objetivo, Y, pertenezca a la familia exponencial debe presentar la siguiente estructura, expresada en forma canónica o natural:

$$f(y;\theta,\phi) = a(y,\phi)exp\left\{\frac{y\theta - k(\theta)}{\phi}\right\}$$

donde,

– f es la función de probabilidad, al ser, en este caso, una variable discreta (datos de conteo) y f caracterizada por los parámetros θ y ϕ. En el caso de que fuera una variable continua f representaría la función de densidad.

– θ el parámetro canónico

– $\phi > 0$ el parámetro de escala o dispersión.

– $k(\theta)$ es una función conocida de θ, denominada función cumulante.

– $a(y, \phi)$ es la constante normalizadora.

– El soporte no depende ni de θ ni de ϕ.

La media de Y es función del parámetro canónico, teniéndose que:

$$E(Y) = \mu = \frac{\partial}{\partial\theta}k(\theta)$$

$$Var(Y) = \sigma^2 = \phi\frac{\partial^2}{\partial\theta^2}K(\theta) = \phi\frac{\partial}{\partial\theta}\left(\frac{\partial}{\partial\theta}k(\theta)\right) = \phi\frac{\partial}{\partial\theta}\mu > 0$$

Por lo que se tiene que la media, μ es una función estrictamente creciente de θ y por consiguiente, μ y θ están relacionadas mediante funciones biyectivas.

Luego,

$$Var(Y) = \phi Var(\mu)$$

siendo $Var(\mu) = \frac{\partial}{\partial\theta}\mu$ la **función varianza**.

- **Componente sistemática:** incluye a las variables predictoras o explicativas utilizadas en la función predictor lineal junto con un conjunto de parámetros de la población bajo estudio, que se debe estimar. Estas variables predictoras,$X_j = X_1, X_2, ..., X_P$, se relacionan mediante:

$$\eta = X'\beta$$

donde,

– η es el predictor lineal

– $X' = (1, X_1, X_2, ..., X_p)$

– $\beta = (\beta_0, \beta_1, ..., \beta_p)'$

- **Función enlace:** o función link, es una función del valor esperado de la variable objetivo, E(Y), como una combinación lineal de las variables explicativas. Esta función enlace permite que el modelo lineal esté relacionado con la variable respuesta.

Es posible que exista más de una función enlace aplicable a un problema de regresión, por lo que se debe elegir aquella que facilite la interpretación del modelo óptimo obtenido.

La función enlace canónica o natural es la que permite relacionar el parámetro canónico con el predictor lineal, siendo esta, una función propia de cada elemento de la familia exponencial:

$$\theta_i = \theta(\mu_i) = \eta_i = X_i^t\beta$$

$$g(\mu_i) = \theta(\mu_i)$$

Teniendo el predictor lineal:

$$\eta = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p,$$

la función link, **g**, viene dada por:

$$g(\mu) = \eta$$

siendo g una función **monótona**, asegurando que la relación entre μ y η sea **biyectiva** y **diferenciable**, requerido para la estimación.

A $g(\mu) = \eta$ se le denomina **Link canónico**.

La construcción del modelo se realiza de forma que éste explique la variable respuesta con el menor error posible. Para ello se deben seguir una serie de etapas:

- **Especificación el modelo teórico:** se determintan las variables de interés y la relación entre ellas. Se quiere un modelo que describa de la forma más simple pero con el mínimo error posible.
  En el caso de tener variables explicativas de tipo cualitativa, es necesario introducir variables *"dummys"*.
- **Estimación de parámetros:** una vez que se tiene el modelo o modelos, se estima el valor de los coeficientes del predictor lineal para cada uno de ellos, obteniendo $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p)^t$ y $\hat{\phi}$, examinado a partir del conjunto de datos observados. Tanto el método de Mínimos Cuadrados como el de Máxima Verosimilitud son dos de los métodos más estudiados y comunes para obtener dicha estimación. En esta etapa se determina si el modelo teórico o modelos teóricos propuestos explican de forma aproximada los datos, además de realizar una comparativa a pares entre los modelos, evaluando para la precisión de esas estimaciones, con el fin de seleccionar el mejor. Dicha comparativa se puede realizar mediante el criterio de información de Akaike, AIC, medida de calidad relativa en un modelo estadístico. Dado un conjuto de modelos candidatos, se selecciona el de menor AIC aunque, cuando el modelo o los modelos no son buenos, este criterio no tiene valor.
- **Adecuación y evaluación del modelo:** se comprueban todos los supuestos que caracterizan al modelo, además de examinar todas las observaciones individuales, los datos influyentes y anómalos. Lo que se quiere obtener es un modelo con el menor número de términos posibles en el predictor lineal y que éstos ajusten bien el conjunto de datos, para así poder obtener interpretaciones claras. Para ello, una vez estimados los parámetros y admitido una combinación satisfactoria de la distribución de la componente aleatoria y la función enlace, se deben eliminar aquellas variables que no expliquen el mayor porcentaje de variabilidad de la variable objetivo, si lo que se pretende es predecir el comportamiento de esta. Hay que valorar la discrepandia que existe entre los valores ajustados y los observados, ya que lo que se quiere es seleccionar el modelo que mejor se ajuste al conjuto de datos. Esto se puede determinar de distintas formas:

1. Mediante coeficiente de determinación R^2, que indica qué tan cerca están los datos de la línea de regresión ajustada, una medida de bondad de ajuste.
2. También se puede calcular el valor de los residuos, que miden la diferencia entre una observación y su valor ajustado. El test de Ljung-Box sobre los residuos, contrasta la existencia de autocorrelación en los residuos, siendo H_0 : los datos se distribuyen

de forma independiente. Interesa que no sea significativo (o sea, debe ser mayor que 0.05).

3. Mediante $RMSE$, raíz cuadrada del error cuadrático medio, se mide la cantidad de error que hay entre dos conjuntos de datos, es decir, compara el valor ajustado con el observado.
4. Otra medida de bondad de ajuste es el estadístico Desviación D, que mide distancia entre el logaritmo de la función de verosimilitud del modelo saturado (modelo con igual número de parámetros que de observaciones) y el modelo que se está estudiando. Para comprobar si es adecuado, se compara el valor de la desviación con el percentil de alguna función de probabilidad. Si el modelo es adecuado, el estadístico de desviación se distribuye asintóticamente según una χ^2_{n-p} con n p grados de libertad (McCullagh & Nelder, 1989). Si el valor de D es pequeño, indica que para un número menor de parámetros, el ajuste es tan bueno como cuando se ajusta el modelo saturado.

- **Interpretación del modelo:** en relación a los valores obtenidos. Comprendiendo sus implicaciones con respecto a la variable respuesta. Es importante recordar la función link utilizada para que la interpretación del modelo sea correcta. Además, esta interpretación de los parámetros se realiza en función del factor de cambio en el valor esperado para un incremento unitario de las variables explicativas.

Finalmente se acepta o se rechaza el modelo, repitiendo el proceso en el caso de no obtener uno adecuado.

1.2.1.1. Modelo de Regresión de Poisson

El modelo de Regresión de Poisson es uno de los más usuales cuando se trata de datos de conteo. La media, μ, es el único parámetro a estimar en la distribución de Poisson. Esta distribución además, se caracteriza porque su esperanza y su varianza coinciden, por lo que puede ocurrir que al realizar un estudio con datos reales, éste resulte insatisfactorio al existir un problema de sobredispersión en los datos, es decir, que ambos valores difieran significativamente.

El presente trabajo tiene como objetivo hacer un estudio sobre la demanda de cierto producto de un determinado lugar, por lo que los datos de los que se disponen corresponden al número de ventas diarias de ese producto (y). Existirán días en los que no habrá ventas, y otros, en cambio, sí que habrá. Si el número medio de ventas diarias es μ, la probabilidad de observar x ventas por día viene dada por:

$$P(Y = y) = \frac{e^{-\mu}\mu^y}{y!}$$

donde y = 0,1,2, ... y parámetro $\mu > 0$

Es decir, en este modelo se asume que la variable respuesta $Y \sim P(\mu)$ con esta función de probabilidad. Por tanto:

Componente aleatoria:

$$f(y;\mu) = exp\{yln(\mu) - \mu - ln\Gamma(y+1)\}$$

donde y = 0,1,2, ... y el parámetro $\mu > 0$

- Parámetro canónico $\theta = ln(\mu)$
- La función cumulante $k(\theta) = \mu$

Función enlace

$$g(\mu_i) = \theta(\mu_i) = ln(\mu_i) = X_i^t\beta = \eta_i$$

donde η_i el **predictor lineal**

1.2.1.2. Modelo de Regresión Binomial Negativa

El modelo de Regresión Binomial Negativa es el modelo paramétrico estándar cuando se tratan datos de conteo que presentan sobredispersión, es decir, no se supone la propiedad de equidispersión (media = varianza).

Existen distintos modelos binomiales negativos, que dependen del tipo de problema de fondo que se está tratando. Existen distintas formas de derivar el modelo pero se va a particularizar en la derivación del modelo como un modelo lineal generalizado.

La distribución Binomial negativa para datos de conteo estudia la probabilidad de observar un número determinado de fracasos (ninguna venta), antes del r-ésimo éxito (venta de r unidades) en una serie de experimentos Bernoulli independientes. El éxito r, debe ser un entero positivo.

Esta distribución pertenece a la familia exponencial siempre y cuando el parámetro de dispersión ϕ, sea introducido en la distribución como una constante, y así poder usar esta distribución como componente aleatoria del modelo lineal generalizado. Se debe tener especial cuidado en la selección de la de función enlace ya que se obtienen diferentes modelos. Los enlaces más usuales son el enlace canónico y el enlace logarítmico, el cuál permite hacer una comparación con el modelo de regresión de Poisson (Hardin & Hilbe, 2012).

Si se asume que la variable respuesta $Y \sim BN(r, p)$ la función de probabilidad viene dada por:

$$P(Y = y) = \binom{y + r - 1}{r - 1} p^r(1 - p)^y = \frac{\Gamma(y + r)}{y!\Gamma(r)} p^r(1 - p)^y$$

donde y = 0,1,2,..., la función $\Gamma(v) = \int_0^{+\infty} x^{v-1}e^{-x}dx$, r = 0,1,2,... y 0 < p < 1.

Derivando el modelo como un modelo lineal generalizado, se tiene que:

Componente aleatoria:

$$f(y; r, p) = exp\left\{yln(1 - p) + rln(p) + ln\binom{y + r - 1}{r - 1}\right\}$$

donde y = 0,1,2,..., r = 0,1,2,... y 0 < p < 1.

- Parámetro canónico $\theta = ln(1 - p)$
- Función cumulante $k(\theta) = -rln(p) = -rln(1 - e^\theta)$
- Parámetro de dispersión $\phi = 1$

Función enlace

$$g(\mu_i) = \theta(\mu_i) = ln\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right) = X_i^t\beta = \eta_i$$

donde ϕ_i es el **predictor lineal**

1.2.2. Exceso de ceros

Cuando se estudian datos de conteo, se aprecian frecuentemente observaciones cuyo valor es cero; por tanto, modelar estos datos con las distribuciones de Poisson o Binomial negativa, lleva a la conclusión de que esta cantidad de ceros no es consistente con dichos modelos, debido a que por algún motivo, existen más observaciones que toman dicho valor de las que cabría esperar de acuerdo a estas distribuciones.

Existen dos tipos de valores nulos, los falsos ceros y los ceros auténticos. A continuación, se presenta un ejemplo que ayuda a distinguirlos:

Ejemplo: En el estudio del número de ventas de un determinado producto de un supermercado, aquellos días en los que no existen ventas puede ser debido a dos motivos:

- El supermercado está cerrado. *Falsos ceros*
- El supermercado está abierto y no ha habido ninguna venta de dicho producto. *Ceros aunténticos*

Esta existencia de falsos ceros puede provocar una abundancia de ceros en la base de datos. Dos de los modelos más usuales para lidiar este tipo de situaciones son:

1. **Modelo inflado con ceros:** estos modelos se utilizan cuando se considera que, por algún motivo, existe una gran cantidad de falsos ceros en los datos.
2. **Modelo en dos partes (Hurdle models):** estos modelos se utilizan cuando se supone que la mayoría de los ceros son auténticos.

1.2.2.1. Modelos inflados con ceros

Éstos se incluyen en los Modelos lineales generalizados mezclados o mixtos (mixture models) y suponen que los ceros provienen de dos procesos distintos: el binomial (falsos ceros) y el proceso que se ha supuesto como Poisson, Binomial negativa,... (ceros auténticos y demás observaciones positivas). De esta forma se tiene:

$$P(Y_i = 0|y_i = 0) = g + (1-g)f(0)$$
$$P(Y_i = y_i|y_i > 0) = (1-g)f(y_i)$$

donde:

– g está definida por un proceso de decisión binario y representa la probabilidad de los ceros falsos.

– f(0) es una distribución para datos de conteo, como la Poisson, Binomial negativa,... Representa la probabilidad de observar un cero en aquellas observaciones que no pertenecen a los falsos ceros.

1.2.2.2. Modelos en dos partes

Otra alternativa que recoge de forma adecuada esta abundancia de valores nulos es el modelo en dos partes. Este método tiene como idea principal determinar si el resultado es cero o no mediante una decisión binaria y a continuación, una vez pasado esta primera decisión, determinar la probabilidad de los valores mayores que cero. Este modelo en dos partes se dividen en:

1. Modelo de decisión binaria (generado por una distribución g)
2. Modelo truncado en cero (generado por una distribución f)

De forma que:

$$P(Y_i = y_i | y_i = 0) = g(0)$$
$$P(Y_i = y_i | y_i > 0) = (1 - g(0)) \frac{f(y_i)}{1 - f(0)}$$

donde:

– $1 - g(0)$ es la probabilidad de no ser el valor cero, pasar la primera decisión.

– $f(y_i)/(1 - f(0))$ es la probabilidad de tomar un valor y_i, con $y_i > 0$

1.2.3. Series Temporales

Los modelos explicados anteriormente no tienen en cuenta la estructura temporal de los datos por lo que aplicar técnicas específicas podría ser otra opción.
Para datos cuantitativos, un modelo de predicción común es el modelo de series temporales.

Se conoce como serie temporal a una sucesión de datos ordenados en el tiempo, correspodientes a una misma variable o característica. Estas observaciones, normalmente se toman en intervalos regulares de tiempo, es decir, presentan el mismo periodo.

Cuando se tiene una serie temporal, es muy importante elegir un modelo probabilístico, el cuál se ajuste correctamente a las observaciones. El objetivo principal en el análisis de series temporales es determinar qué proceso estocástico ha sido capaz de generar la serie bajo estudio. Esto es posible utilizando la teoría de los procesos estocásticos y gracias a ello se podrá caracterizar el comportamiento de la serie y predecir en el futuro.
Un proceso estocástico es una familia de variables aleatorias, las cuales están relacionadas entre sí y siguen una ley de distribución conjunta, por lo que gran parte de la información de estas distribuciones conjuntas se puede describir en términos de medias, variaciones y covarianzas.

Los modelos de serie temporal suponen que la combinación de patrones es frecuente con el tiempo, por lo tanto, mediante el análisis de una serie temporal se podrá describir tanto el comportamiento que la serie ha tenido en el pasado como predecir sus valores en un futuro no muy lejano.

En un análisis descriptivo de series temporales se distinguen cuatro etapas, necesarias para elegir el modelo. Éste va a proporcionar información descriptiva o va a facilitar la estimación de los parámetros con los que se ajustan mejor los datos que se tienen y por consiguiente, se va a poder obtener predicciones.

- **Representación gráfica:** mediante la exploración del gráfico se puede tener una idea del comportamiento de la serie, detectando, por ejemplo, posibles tendencias crecientes o decrecientes, y ofrediendo información acerca de qué métodos serán los más adecuados para aplicar.
- **Modelización:** el objetivo en esta etapa es buscar un modelo que se ajuste de la mejor forma posible a los datos de la serie, por lo que aquí se elabora una representación simplificada de las características más importantes de esta.
- **Validación del modelo:** es fundamental el análisis de los residuos, los cuales deben tener media o varianzas constantes. En esta etapa es necesario conocer si el modelo que se tiene es el adecuado de cara a efectuar predicciones.
- **Predicción:** dada una nueva observación, estimar el valor futuro que tendrá la variable respuesta. Esta predicción puede ser puntual o por intervalos de confianza. Dependiendo del tipo de tendencia y de si presenta o no estacionalidad, se podrán utilizar distintos alisados para conseguir el objetivo de esta etapa.

El análisis clásico de las series temporales supone que cualquier serie de tiempo se integra de cuatro componentes responsables de la variación de la variable bajo estudio con respecto al tiempo.

Estas componentes son:

- **Tendencia secular:** tendencia persistente a largo plazo. Los factores que inluyen en este tipo de tendencia pueden ser debidos a cambios en los hábitos del consumidor o variaciones demográficas.
- **Componente estacional:** movimientos regulares de la serie cuya periodicidad es inferior a un año. Recoge las oscilaciones de la serie que se repiten de manera periódica en un año (las estaciones), una semana (los fines de semana), un día (las horas puntas) o en cualquier otro periodo.
- **Variación cíclica:** esta variación está caracterizada por movimientos recuerrentes en torno a la tendencia, duran más de un año y ocurre incluso después de eliminar las variaciones estacionales o irregulares. Además, a diferencia de las variaciones estacionales, tienen período y amplitud variables.
- **Componente irregular:** incluye las variaciones de la serie que no pueden ser expicadas por las demás componentes. Al no incluir ningún compotarmiento sistemático o regular no será posible su predicción.

Este enfoce clásico de series temporales supone además, la existencia de tres tipos de series:

- **Modelo aditivo:** supone que la serie está compuesta por la suma de la tendencia, la componente estacional y la irregular.

$X_t = T_t + E_t + I_t$

- **Modelo multiplicativo:** supone que la serie está compuesta por el producto de la tendencia, la componente estacional y la irregular.

$X_t = T_t \times E_t \times I_t$

- **Modelo mixto:** supone que la serie está compuesta por la suma y producto, con distintas variantes, de la tendencia, la componente estacional y la irregular.

Una vez detectado el modelo, se debe descomponer la serie en sus distintas componentes. Gracias a esto, se podrán ver individualmente y se podrá realizar el posterior análisis y pronóstico sin la influencia del ruido o la estacionalidad.
Por un lado, entre los métodos de estimación de la tendencia destaca el método de media móvil que trata de indentidficar a estas mediante una transformación lineal a partir del conjunto de datos. Por otro lado, el análisis de la estacionalidad se realiza con el propósito de desestacionalizar la serie para así hacer comparables los datos correspodientes a estaciones diferentes; y de predecir, ya que se necesita conocer esta componente para el momento de la predicción. Se supone que el efecto de la estacionalidad consiste en aumentar o disminuir la tendencia mediante una cantidad constante, por lo que para la descomposición de la estacionalidad se usan índices llamados IVE (índices de variación estacional), expresando con ellos las variaciones porcentuales de las diferentes estaciones del año respecto a la media anual. La serie desestacionalizada se obtiene restándole a las observaciones su correspondiente IVE.

Para que los procesos estocásticos proporcionen una predicción estable, se les imponen la condición de estacionariedad, lo que supone que sus propiedades estadísticas deben ser invariantes ante una traslación del tiempo.
Todo proceso lineal es estacionario, luego, cuando la media y la varianza de la serie temporal se mantiene estable en el tiempo, se modelará este conjunto de observaciones a través de distintos modelos como son el modelo autorregresico AR(p), el modelo de medias móviles MA(q) o mediante el modelo autorregresivo de medias móviles ARMA(p,q).

Cuando una serie temporal no presenta media y/o varianza constante, entonces se tiene una serie de tiempo no estacionaria.

Uno de los modelos clásicos utilizados para el pronóstico de series temporales es el modelo integrado autorregresivo de medias móviles, ARIMA(p,d,q). Estos modelos suponen que al aplicarle al proceso estocástico la d-ésima diferencia, seguirá un ARMA(p+d,q), que es un proceso estacionario. A d, número de diferencias que hay que tomar para que el proceso sea estacionario se le conoce como orden de integración del proceso.

1.2.3.1. Metodología de Box-Jenkins

En el análisis de series de tiempo, la metodología de Box-Jenkins, se aplica a los modelos autorregresivos de media móvil (ARMA) o a los modelos autorregresivos integrados de media móvil (ARIMA), para encontrar el mejor ajuste de una serie temporal y así generar pronósticos más acertados. El método original utiliza un enfoque de modelado iterativo en tres fases o estapas:

- Identificación: donde se representan los datos y las funciones de autocorrelación (FAS) y autocorrelación parcial (FAP), para así estabilizar la serie en el caso de que no exista la estacionariedad. Si los datos no se distribuyen horizontalmente alrededor de una media constante o si la FAS o FAP no caen a cero rapidamente, entonces la serie es no estacionaria y por tanto, será necesario su transformación. En esta etapa también se identifican la tendencia y estacionalidad de la serie.
- Estimación o test: Tanto la estimación de máxima verosimilitud o mínimos cuadrados no lineales son dos de los métodos más comunes para estimar los parámetros de la serie que mejor se ajustan al modelo seleccionado.

- Diagnóstico de los modelos y selección del mejor: como hipótesis de partida, para que un proceso sea estacionario, los residuos deben ser independientes y tanto la media como la varianza deben ser constantes en el tiempo. Por tanto, para el análisis de los residuos se usa el contraste de Ljung-Box, y para la selección del modelo se puede usar el criterio AIC, criterios de los que ya se hablaron en el modelo de regresión lineal general.

Una vez seleccionado el mejor modelo que se ajuste al conjunto de datos, las predicciones se pueden realizar utilizando los parámetros estimados.

1.2.3.2. Análisis espectral

La cantidad de ventas de un determinado producto depende de ciertos criterios como pueden ser el día de la semana o mes. De esta forma y considerando el año como tiempo de referencia, es importante el estudio de estas componentes periódicas que, tanto para datos semanales (de periodo 7) como para datos mensuales (de periodo 12), se consideran estacionales. Este tipo de series se conoce como series multiestacionales y es posible modelarla mediante procesos estocásticos estacionarios o no etacionarios; o mediante el análisis espectral.

Un enfoque alternativo al análisis clásico es el análisis espectral. Este análisis permite el estudio de la componente estacional, siendo por tanto, una herramienta adecuada para sacar a luz las componentes periódicas de la serie, ya que todo proceso estocástico puede ser representado mediante transformaciones lineales de funciones trigonométricas o, mediante funciones exponenciales complejas, aunque ponderados por coeficientes aleatorios.

Este enfoque supone que cualquier serie puede ser transformada, mediante transformaciones de Fourier, en ciclos formados con senos u cosenos, basándose en el análisis de la frecuencia, que es el recíproco del período del ciclo. Por tanto, la idea básica del análisis espectral es que un proceso estacionario puede ser explicado como la suma de movimientos de seno y coseno de diferente frecuencia y amplitud.
Para describir el comportamiento de la serie es necesario determinar los ciclos de las diferentes frecuencias que predominan, pudiendo ser éstos de corto o largo plazo. Al final, lo que se consigue es descomponer a la serie en la totalidad de frecuencias existentes.

A esta representación se le conoce como representación espectral de un proceso estocástico, siendo la densidad espectral la varianza de los coeficientes de dicha representación. Por tanto, una estimación de esta densidad espectral puede obtenerse mediante la transformada de Fourier.

Mediante la representación del periodograma se obtiene una estimación de la densidad espectral y muestra la varianza para cada una de las frecuencias de Fourier. Se puede representar por frecuencia o por periodo. Además, en el eje vertical se representan la suma de los cuadrados del seno y coseno, lo que indica que los datos de mayor valor son las frecuencias más representativas de la serie.

1.3. Machine Learning

> "Machine Learning es la ciencia que permite que las computadoras aprendan y actúen como lo hacen los humanos, mejorando su aprendizaje a lo largo del

> tiempo de una forma autónoma, alimentándolas con datos e información en forma de observaciones e interacciones con el mundo real." — Dan Faggella

Machine Learning (ML), subcampo de las ciencias de la computación y técnica del ámbito de la Inteligencia Artificial (IA), se basa en algoritmos de aprendizaje. Los algoritmos de ML construyen modelos basados en los datos de muestra (datos de entrenamiento), siendo éstos modelos no lineales más complejos que los lineales, capaces de abordar distintas incertidumbres como situaciones económicas o clima, además de ofrecer predicciones más precisas y oportunas mediante algoritmos capaces de identificar patrones complejos y predecir comportamientos futuros.

Para poder llevar a cabo un problema de ML, se debe seguir una serie de pasos.

1.3.1. Preprocesamiento de los datos y elección de los modelos

En primer lugar, se realiza un pre-procesamiento de los datos, ya que a menudo estos pueden presentarse en fomatos no válidos para ser procesados por el modelo. Algunos de los posibles procesos a tener en cuenta, son: el tratamiento de los valores perdidos, la transformación de las variables, la codificación de las variables o la normalización.

Se necesita un modelo que consiga la mayor precisión posible, por lo que en segundo lugar, se debe identificar la variable objetivo y el conjunto de características o variables predictoras. Dado que puede ocurrir que exista correlación entre las características se debe aplicar técnicas para reducir su dimensión.
Existen varios métodos para optimizar el modelo y conseguir una mejora de éste, dando como resultado uno más fiable y eficaz.
Los tres enfoques más comunes para conseguir dicho objetivo son:

- Feature Engineering es como se denomina en Machine Learning al proceso de transformar los datos brutos en datos utilizables en los modelos predictivos.
- Optimización de Hiperparámetros.
- Uso de diferentes algoritmos de ML para crear nuevos modelos.

1.3.1.1. Feature Engineering

El rendimiento de un modelo se incrementa reduciendo la carga de datos. En la etapa de Feature Engineering se reduce el número de características si es necesario, identificando aquellas que son importantes para la predicción de la variable objetivo y eliminando las irrelevantes. Llevarlo a cabo es posible mediante diversas técnicas entre las que se encuentra el Análisis de Componentes Principales (PCA).

1.3.1.2. Optimizar Hiperparámetros

Otra etapa en el proceso de optimizar el rendimiento de los distintos modelos es ajustando sus hiperparámetros de forma que proporcionen una mayor precisión después de su aprendizaje o entrenamiento. Se conoce como hiperparámetros a determinados parámetros "internos" de los distintos algoritmos de Machine Learning; como por ejemplo, en árboles de decisión existen hiperparámetros tales como número de nodos, números de hojas, profundidad máxima de nodos...

Los parámetros del modelo son distintos a los hiperparámetros. Para ajustar los hiperparámetros se debe entender el efecto de cada parámetro, ya que estos difieren según

el algoritmo utilizado. Es conveniente generar y comparar varios modelos con distintas combinaciones de parámetros para así encontrar la mejor combinación de los mismos. Para ello, puede utilizarse la búsqueda en cuadrícula o la búsqueda aleatoria.

Al ajustar los hiperparámetros puede que se produzca un sobreajuste y por tanto, se necesite utilizar la validación cruzada; aunque cuando se trata de Big Data, esta etapa es de difícil implementación .

1.3.1.3. Generar modelos alternativos con diferentes algoritmos

La predicción suele variar dependiendo del algoritmo utilizado, por lo que optimizar un único modelo no siempre es la mejor solución. Para obtener el modelo de mejor rendimiento, el modelo optimizado se debería probar con otros algoritmos.

A continuación se detallan algunos de los algoritmos de ML.

Random Forest

Formalizado por Breiman, se utiliza tanto para problemas de regresión como de clasificación y se basa en la creación de múltiples árboles de decisión a partir de los datos de partida.
Este algoritmo selecciona de manera aleatoria distintos conjuntos de variables, que posteriormente serán utilizados para la construcción de cada árbol individual.

Un árbol de decisión consiste en una partición del espacio, definido por las variables predictoras, en distintas regiones. La creación del árbol parte del nodo raiz, formado por todos los datos del conjunto de entrenamiento y estructurados por las distintas variables. Las particiones del árbol se hacen en función del mejor atributo que lo divida y de unos valores, ambos elegidos por el algoritmo. A cada punto del árbol donde se divide el espacio predictor se le conoce como nodo interno. Si éste no se divide a su vez en otros nodos, entonces se está ante un nodo terminal. Los nodos terminales se dan bien porque solo exista un dato o porque no exista diferencia entre valores. A los segmentos de los árboles que conectan los nodos se les denomina ramas. Su finalidad es encontrar aquella partición o caja del espacio predictor que minimice la suma de los residuos. De esta forma, este algoritmo repite el proceso hasta encontrar el mejor predictor y punto de corte.

Random Forest, a diferencia de los árboles, busca la mejor característica entre un subconjunto aleatorio de características al dividir un nodo, presenta un sesgo prácticamente nulo y existe una mayor dificultad de que aparezan algún tipo de sobreajuste en sus hiperparámetros, por la gran cantidad de árboles creados.

El proceso que sigue Random Forest es el siguiente:

1. Se divide al conjunto de observaciones en conjunto de datos de entrenamiento y conjunto test.
2. Se especifica un número de variables de entrada siendo este menor al original, de manera que para cada nodo, sean seleccionadas aleatoriamente entre todas. La mejor división de estas variables explicativas es usada para ramificar el árbol. Este número de variables especificado se mantiene constante durante todo el proceso de creación del árbol.A diferencia de los árboles de regresión, aquí no hay proceso de poda, cada árbol crece hasta su máxima extensión.

3. Para realizar una predicción, se elige, entre todos los árboles creados, mediante la media de los valores de predicción por todos los árboles del modelo o mediante el nodo más votado.

Gradient-boosting

Esta técnica de aprendizaje automático también se basa en árboles de decisión, pero a diferencia de bosques aleatorios, los árboles son entrenados de forma secuencial, de forma que cada nuevo árbol trata de mejorar los errores, ajustando los residuos, de los árboles anteriores.

Este algoritmo sigue el siguiente procedimiento:

1. Sobre los datos de entrenamiento, se ajusta un primer modelo con el cuál se predice la variable objetivo, calculándose posteriormente los residuos.
2. Se ajusta un nuevo modelo que trata de corregir los errores del modelo anterior intentando predecir sus residuos.
3. De forma conjunta, se calculan los residuos de ambos modelos y se ajusta un tercer modelo que intenta corregir los errores de los modelos anteriores.
4. Esto se repite M veces, de forma que cada nuevo predictor minimice el error.

Redes neuronales artificiales

Uno de los tipos de algoritmos más comunes del Machine Learning son las Redes Neuronales Artificiales. Las Redes Neuronales Artificiales (RNA) son sistemas informáticos inspirados en el funcionamiento del cerebro humano de forma que lo puedan imitar. Son capaces de automatizar funciones que en un pincipio solo podían ser realizadas por personas, es decir, adoptan el conocimiento y la percepción de estas a través de ejemplos. Las RNA están formadas por un conjunto de unidades o neuronas artificiales, conectadas entre sí, con el fin de transmitirse señales, a través de unos enlaces de comunicación. Cada uno de estos enlaces tiene asociado un peso en los que se encuentra el conocimiento que tiene la RNA acerca de un determinado problema. Estas neuronas reciben unos datos de entrada y generan unos de salida. A cada neurona le llegan múltiples entradas que van siendo filtradas gracias al peso de esta, que pueden variar, y así poder conocer la importancia y el efecto sobre el procesamiento de la neurona.
Este algoritmo es un modelo matemático basado en una estructura de grafo dirigido, cuyos nodos son neuronas artificiales. Su finalidad es modelar la variable objetivo como una función no lineal, a través de las variables de entradas, de las que extrae combinaciones lineales de las características más importantes de cada neurona.
Las neuronas que componen la red neuronal están organizadas de forma jerárquica formando capas o niveles, de manera que cada neurona de un mismo nivel tiene como entrada de información la misma fuente y como salida el mismo destino. Existen tres tipos de capas:

1. Capa de entrada: recibe información del exterior.
2. Capa oculta: tanto sus entradas como salidas están dentro del sistema, no tienen contacto con el exterior. En estas capas se optimizan los pesos de las variables de entradas, mediante el método de retropropagación, con el fin de obtener el mejor modelo predictivo.
3. Capa de salida: envía respuesta al exterior.

Para poder aplicar este algoritmo, uno de los procedimientos más impotantes es ajustar los datos a una escala común, es decir, normalizar los datos y, en el caso de variables categóricas, reescalarlos, para así evitar el efecto de la escala de las variables. Esto hará posible la comparación entre los valores predichos y los reales. Posteriormente, se deshace el cambio para representar las predicciones.

1.3.2. Entrenar el modelo

Para cada modelo de aprendizaje supervisado los datos se dividen en conjunto de entrenamiento (donde se entrena el modelo, aprendiendo de estos resultados conocidos) y de prueba (donde se prueba y compara la predicción de la variable objetivo con los valores reales), al disponer de la variable histórica etiquetada para entrenar el modelo, es decir, se conoce cada entrada del conjunto de entrenamiento y sus resultados de salida. En el caso de tener datos que solo contienen entradas de la variable objetivo se hablaría de un problema ML no supervisado.

Se debe seleccionar el mejor algoritmo de ML, para ello, se utilizan diversos algoritmos para entrenar un número múltiple de modelos que serán comparados, normalmente, mediante el Error Cuadrático Medio (MSE), que mide la diferencia media cuadrática entre los valores estimados y los reales.

1.3.3. Evaluar y seleccionar el modelo

Siempre se debe evaluar el modelo que es seleccionado y ajustado previamente al conjunto de entrenamiento. Mediante el conjunto de prueba se estima el rendiemiento del modelo en los datos nuevos, como por ejemplo, haciendo una estimación del error de generalización del modelo, siendo esta la capacidad del modelo para hacer predicciones utilizando nuevos datos.
Por último, si el valor obtenido según la métrica utilizada resulta satisfactorio, se tendrá el modelo listo para realizar predicciones sobre valores futuros.

Capítulo 2

Aplicación a un caso prático

2.1. Modelización y predicción de ventas

En este capítulo se lleva a cabo la aplicación práctica de técnicas estadísticas para datos de conteo y algoritmos de aprendizaje supervisado, con el fin de predecir la demanda de un producto. Para ello se toma como variable dependiente u objetivo la cantidad diaria vendida de éste.

2.1.1. Descripción de los datos

La base de datos de la que se dispone contiene la serie temporal diaria de cierto producto de un supermercado de Las Palmas de Gran Canarias, desde el 2 de Enero de 2014 hasta el 26 de Agosto de 2016, por lo que se cuenta con un total de 973 observaciones. Se tienen recogidas, entre otras, distintas variables climatológicas, pero la existencia de un elevado número de valores perdidos en estas sugiere realizar las pruebas sin esas variables, al tener que eliminar observaciones y por consiguiente, perder la estructura temporal de los datos. Por ello, en estas pruebas, la modelización de las ventas se hace en función de estas variables predictoras: Precio diario, día de la semana y mes.
Al no tener información sobre el calendario de apertura en festivos de dicho negocio, se ha creado una nueva covariable, *DiaLaborable*, en base al calendario laboral de 2014-2016, lo que ofrecerá información sobre si influyen o no en las predicciones.

En primer lugar, se realiza un breve resumen descriptivo de los datos recogidos en el fichero "9004Item15.RData".

```
load("9004Item15.RData")
datos=df_item_agr_ampliado[,c(1:3,16:17)] #Selección de variables
```

Se crea la covariable *DiaLaborable*, de modo que:

$$DiaLaborable = \begin{cases} 1 & si\ es\ laboral \\ 0 & si\ no\ es\ laboral \end{cases}$$

Se ha tomado como referencia, para la creación de dicha variable, el calendario laboral de las Palmas de Gran Canarias de los años 2014, 2015 y 2016.

```
# Se crea DiaLaborable
datos$DiaLaborable<-ifelse(datos$Fecha=="2014-01-06"  |
datos$Fecha=="2014-03-04" | datos$Fecha=="2014-04-18" |
datos$Fecha=="2014-06-24" | datos$Fecha=="2014-12-25" |
datos$Fecha=="2015-01-06" | datos$Fecha=="2015-11-02" |
datos$Fecha=="2015-12-25" | datos$Fecha=="2016-01-06" ,0,1)
```

Se realiza un análisis exploratorio de los datos para conocer la estructura y tipo de éstos:

```
str(datos) # Estructura de los datos
```

```
## 'data.frame':    973 obs. of  6 variables:
##  $ Fecha            : Date, format: "2014-01-02" "2014-01-03" ...
##  $ CantidadDiaria   : num  103 55 30 0 0 0 34 75 56 42 ...
##  $ PrecioMedioDiario: num  1.95 1.95 1.95 1.95 1.95 1.95 1.95 1.95 1.95 1.95 ...
##  $ diasem           : int  5 6 7 1 2 3 4 5 6 7 ...
##  $ mes              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DiaLaborable     : num  1 1 1 1 0 1 1 1 1 1 ...
```

```
summary(datos) # Resumen de los valores que toma cada variable
```

```
##      Fecha            CantidadDiaria  PrecioMedioDiario     diasem
##  Min.   :2014-01-02   Min.   :  0.0   Min.   :0.490     Min.   :1
##  1st Qu.:2014-09-02   1st Qu.: 14.0   1st Qu.:1.870     1st Qu.:2
##  Median :2015-05-03   Median : 31.0   Median :1.870     Median :4
##  Mean   :2015-05-03   Mean   : 42.9   Mean   :1.774     Mean   :4
##  3rd Qu.:2016-01-01   3rd Qu.: 48.0   3rd Qu.:1.870     3rd Qu.:6
##  Max.   :2016-08-31   Max.   :735.0   Max.   :1.950     Max.   :7
##       mes          DiaLaborable
##  Min.   : 1.000   Min.   :0.0000
##  1st Qu.: 3.000   1st Qu.:1.0000
##  Median : 6.000   Median :1.0000
##  Mean   : 6.028   Mean   :0.9908
##  3rd Qu.: 9.000   3rd Qu.:1.0000
##  Max.   :12.000   Max.   :1.0000
```

Tanto *diasem* como *mes* requieren una descripción adicional en función de los valores que toman. Esto se realiza más adelante.

A continuación, se muestra una representación gráfica de las ventas diarias recogidas por la base de datos.

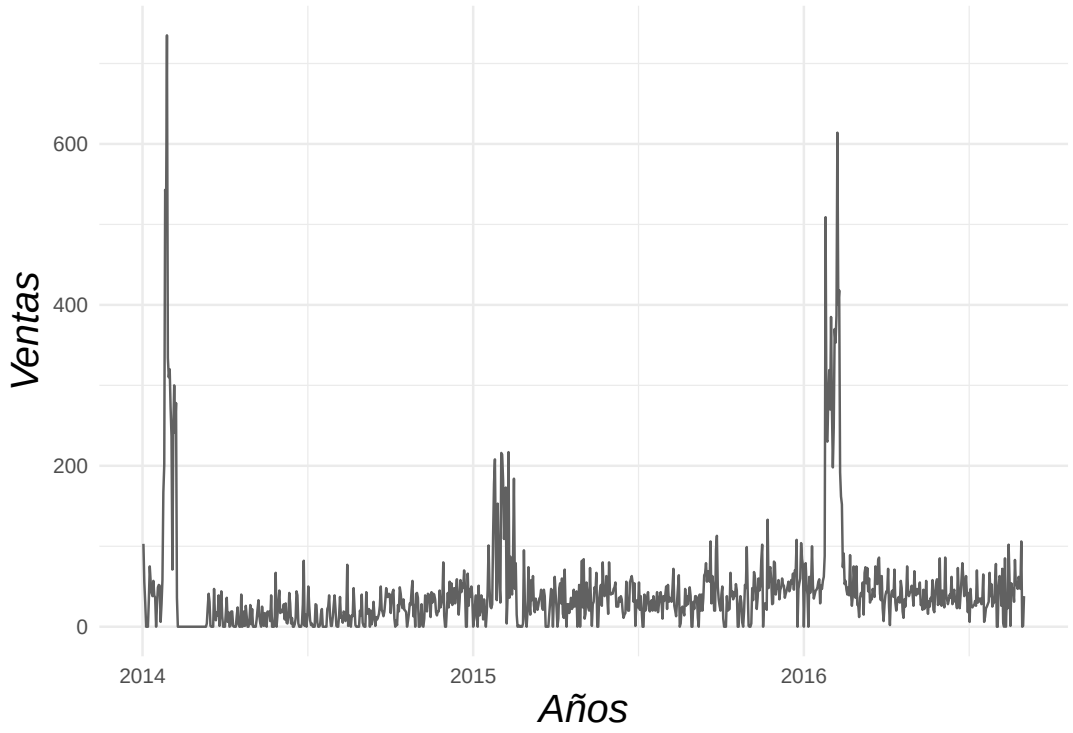


Figura 2.1: Evolución de la cantidad diaria de ventas

Se observa que, en general, estas ventas se mantienen relativamente estables durante todo el año, salvo en los primeros meses donde se percibe un incremento.
Con respecto a dicho incremento, se aprecia que en 2014, existen días el los que no se han producido ventas. Esto puede ser porque se haya agotado dicho producto debido a un exceso de demanda en unos días determinados. Por el contrario, en 2015, a pesar de este incremento, las ventas se mantuvieron más uniformes. Por último, en 2016, a pesar de la gran cantidad de producto vendido, no se quedaron sin stock.

Se crea una nueva covariable *Carnaval* debido a que este aumento de ventas puede ser provocado por éste, ya que dicha festividad se celebra en Las Palmas de Gran Canarias durante los primeros meses del año. Para la creación de la covariable se tiene en cuenta el calendario de festivos de Las Palmas para los años 2014-2016. De modo que:

$$Carnaval = \begin{cases} 1 & si\ es\ carnaval \\ 0 & si\ no\ es\ carnaval \end{cases}$$

Basándose en el calendario de dicha festividad:

```
datos$Carnaval=c(rep(0,44),rep(1,22),rep(0,328),rep(1,22),rep(0,341),
                 rep(1,24),rep(0,192))
datos$Carnaval=factor(datos$Carnaval)
str(datos$Carnaval)
```

```
##  Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Este incremento de ventas provoca una sobredispersión en los datos que se puede ver reflejado en el valor de la varianza muestral, siendo éste excesivamente más elevado que la media:

```
## [1] 4181.715
```

La media muestral de la cantidad diaria vendida:

```
## [1] 42.90339
```

Otra forma de ver la evolución de las ventas es representando esta respecto a los días de la semana *(Figura 2.2)* y respecto al mes *(Figura 2.3)*.

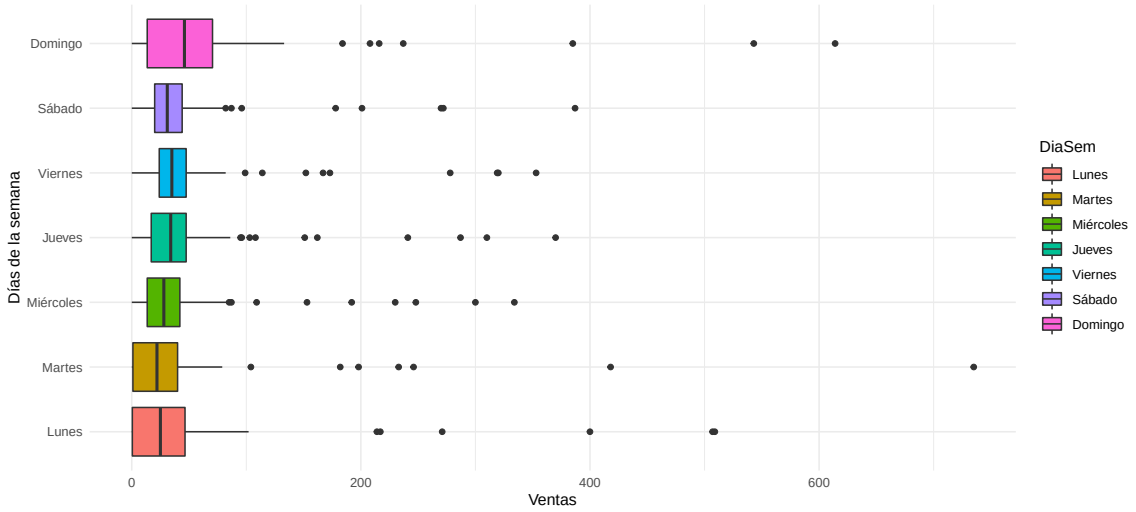


Figura 2.2: Ventas con respecto al día

Según la media de ventas, la gran mayoría se reparte entre los domingos, lunes y viernes. Esto puede ser porque los viernes y domingos se dispone de mayor tiempo libre y porque los lunes se prefiere realizar la compra de toda la semana.

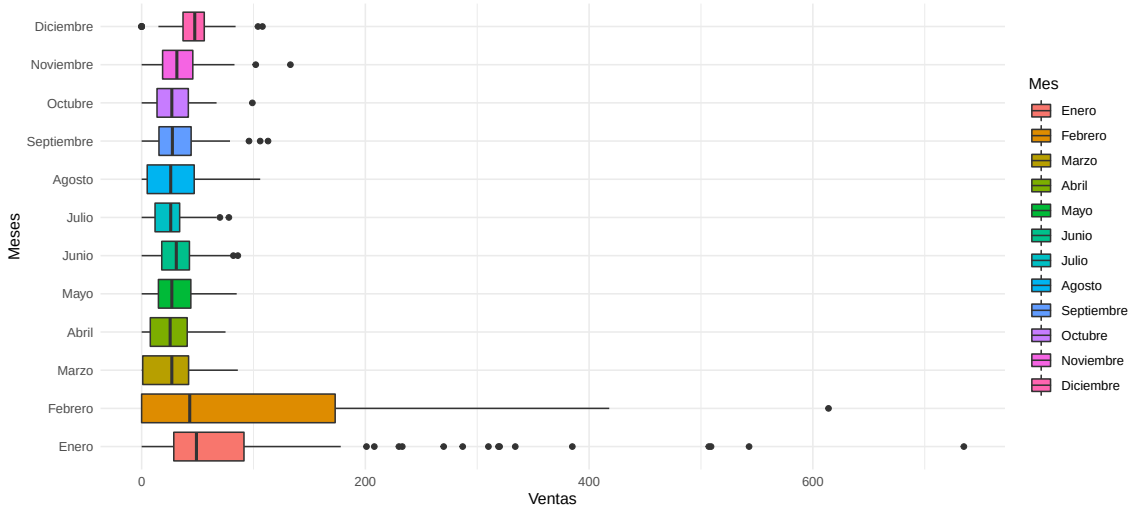


Figura 2.3: Ventas con respecto al mes

Como ya se comentó anteriormente, es en los meses de enero/febrero donde se aprecia una subida de ventas, con mayor importancia en el mes de febrero, lo que puede hacer sospechar que los valores outliers de la *(Figura 2.2)* sean debido a este aunmento de cantidades vendidas.

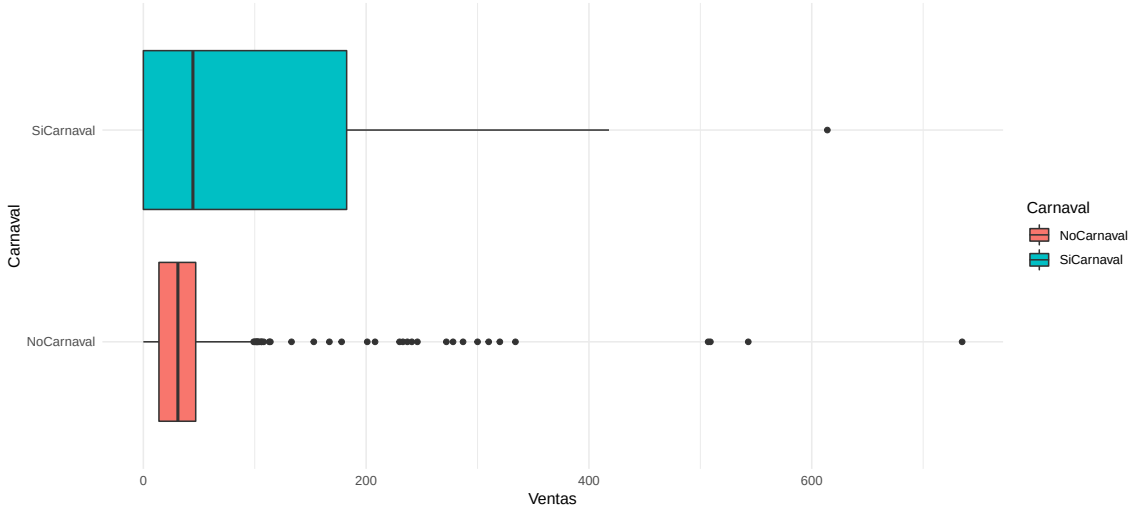


Figura 2.4: Ventas con respecto a la festividad Carnaval

Por último, en esta gráfica se concluye que efectivamente, durante el Carnaval se vende mucho más producto que cuando no lo es.

Como objetivo de estudio, se han considerado diversos modelos predictivos, cuyo rendimiento se ha medido de forma numérica y gráfica. Dado que la variable objetivo toma valores enteros, todas las predicciones han sido redondeadas al entero más cercano.

Para comparar los distintos modelos, se van a dividir los datos en conjunto de entrenamiento y conjunto test, reservando para este último, las últimas 30 observaciones donde se podrán comparar las predicciones y obtener el mejor modelo.

```
# PARTICION EN ENTRENAMIENTO Y TEST
n=nrow(datos) # Total observaciones
nt=30
indient=1:(n-nt) # Datos entrenamiento
inditest=(n-nt+1):n # Datos test
```

2.1.2. Modelo de regresión de Poisson

Como primer modelo predictivo se utiliza el modelo de regresión de Poisson. Éste se aplica en primer lugar a los datos de entrenamiento. *PrecioMedioDiario*, *DiaSem*, *Mes*, *DiaLaborable* y *Carnaval* han sido las variables explicativas que se han seleccionado.

Para poder aplicar los modelos de regresión, las variables categóricas deben ser definidas como variables *dummy*:

```
datos$Mes=factor(datos$Mes)
datos$DiaSem=factor(datos$DiaSem)
datos$DiaLaborable=factor(datos$DiaLaborable)
datos$Carnaval=factor(datos$Carnaval)
```

Se aplica la regresión de Poisson al modelo con la interacción *DiaSem* y *Mes* y sin interacción, comparando ambos posteriormente.

Modelo con interacción

```
# Modelo con interacción
modeloP1=glm(CantidadDiaria ~  PrecioMedioDiario + DiaSem*Mes +
               DiaLaborable + Carnaval,family = "poisson",
               data =datos[indient,])
summary(modeloP1)
```

```
##
## Call:
## glm(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem * Mes +
##     DiaLaborable + Carnaval, family = "poisson", data = datos[indient,
##     ])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -17.2038   -3.7578   -0.2436    2.3571   28.5752
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -9.16826   83.96631  -0.109 0.913052
## PrecioMedioDiario              -1.30664    0.01905 -68.601  < 2e-16 ***
## DiaSemMartes                   -0.17255    0.03898  -4.427 9.55e-06 ***
## DiaSemMiércoles                -0.36535    0.04117  -8.874  < 2e-16 ***
## DiaSemJueves                   -0.40061    0.03979 -10.069  < 2e-16 ***
## DiaSemViernes                  -0.50535    0.03828 -13.201  < 2e-16 ***
## DiaSemSábado                   -0.53487    0.04013 -13.329  < 2e-16 ***
## DiaSemDomingo                   0.05235    0.03588   1.459 0.144586
## MesFebrero                     -0.89711    0.04045 -22.176  < 2e-16 ***
## MesMarzo                       -1.98674    0.07018 -28.311  < 2e-16 ***
## MesAbril                       -1.29500    0.06717 -19.280  < 2e-16 ***
## MesMayo                        -1.18656    0.06160 -19.262  < 2e-16 ***
## MesJunio                       -1.41727    0.06433 -22.031  < 2e-16 ***
## MesJulio                       -1.45186    0.07092 -20.473  < 2e-16 ***
## MesAgosto                      -1.75931    0.08785 -20.026  < 2e-16 ***
## MesSeptiembre                  -1.18134    0.07143 -16.538  < 2e-16 ***
## MesOctubre                     -1.33549    0.08035 -16.622  < 2e-16 ***
## MesNoviembre                   -0.65090    0.06032 -10.790  < 2e-16 ***
## MesDiciembre                   -0.62234    0.05700 -10.918  < 2e-16 ***
## DiaLaborableSiLaborable        16.03821   83.96630   0.191 0.848520
## CarnavalSiCarnaval              0.09977    0.01946   5.127 2.95e-07 ***
## DiaSemMartes:MesFebrero         0.25432    0.05541   4.590 4.44e-06 ***
## DiaSemMiércoles:MesFebrero      0.15022    0.05836   2.574 0.010055 *
## DiaSemJueves:MesFebrero         0.31018    0.05643   5.496 3.87e-08 ***
## DiaSemViernes:MesFebrero        0.49449    0.05558   8.897  < 2e-16 ***
## DiaSemSábado:MesFebrero         0.33628    0.05835   5.763 8.27e-09 ***
## DiaSemDomingo:MesFebrero        0.17744    0.05177   3.428 0.000609 ***
## DiaSemMartes:MesMarzo           0.81204    0.09170   8.855  < 2e-16 ***
## DiaSemMiércoles:MesMarzo        0.82361    0.09332   8.826  < 2e-16 ***
```

```
## DiaSemJueves:MesMarzo          1.09057    0.09320  11.702  < 2e-16 ***
## DiaSemViernes:MesMarzo         1.39046    0.08903  15.619  < 2e-16 ***
## DiaSemSábado:MesMarzo          1.07881    0.09129  11.818  < 2e-16 ***
## DiaSemDomingo:MesMarzo         0.58215    0.08800   6.615 3.71e-11 ***
## DiaSemMartes:MesAbril         -0.04776    0.09778  -0.488 0.625207
## DiaSemMiércoles:MesAbril       0.54364    0.09004   6.038 1.56e-09 ***
## DiaSemJueves:MesAbril          0.69886    0.08869   7.880 3.27e-15 ***
## DiaSemViernes:MesAbril         0.80234    0.08917   8.998  < 2e-16 ***
## DiaSemSábado:MesAbril          0.53174    0.09375   5.672 1.41e-08 ***
## DiaSemDomingo:MesAbril         0.46591    0.08509   5.475 4.37e-08 ***
## DiaSemMartes:MesMayo          -0.14477    0.09322  -1.553 0.120426
## DiaSemMiércoles:MesMayo        0.65489    0.08499   7.706 1.30e-14 ***
## DiaSemJueves:MesMayo           0.58353    0.08478   6.883 5.85e-12 ***
## DiaSemViernes:MesMayo          0.89439    0.07985  11.200  < 2e-16 ***
## DiaSemSábado:MesMayo           0.37829    0.08870   4.265 2.00e-05 ***
## DiaSemDomingo:MesMayo          0.47728    0.07701   6.198 5.73e-10 ***
## DiaSemMartes:MesJunio          0.42993    0.08835   4.866 1.14e-06 ***
## DiaSemMiércoles:MesJunio       0.62702    0.08887   7.055 1.72e-12 ***
## DiaSemJueves:MesJunio          0.83323    0.08568   9.725  < 2e-16 ***
## DiaSemViernes:MesJunio         1.25146    0.08243  15.182  < 2e-16 ***
## DiaSemSábado:MesJunio          1.08808    0.08575  12.690  < 2e-16 ***
## DiaSemDomingo:MesJunio         0.56656    0.08136   6.964 3.31e-12 ***
## DiaSemMartes:MesJulio         -0.10438    0.10467  -0.997 0.318636
## DiaSemMiércoles:MesJulio       0.19403    0.10139   1.914 0.055647 .
## DiaSemJueves:MesJulio          0.79322    0.09111   8.706  < 2e-16 ***
## DiaSemViernes:MesJulio         0.92948    0.09004  10.323  < 2e-16 ***
## DiaSemSábado:MesJulio          0.77290    0.09467   8.165 3.23e-16 ***
## DiaSemDomingo:MesJulio         0.61862    0.08690   7.119 1.09e-12 ***
## DiaSemMartes:MesAgosto         0.34587    0.12555   2.755 0.005873 **
## DiaSemMiércoles:MesAgosto      0.53866    0.12625   4.267 1.98e-05 ***
## DiaSemJueves:MesAgosto         0.84689    0.11867   7.136 9.58e-13 ***
## DiaSemViernes:MesAgosto        1.43391    0.10704  13.395  < 2e-16 ***
## DiaSemSábado:MesAgosto         1.05812    0.11256   9.400  < 2e-16 ***
## DiaSemDomingo:MesAgosto        0.57996    0.10919   5.312 1.09e-07 ***
## DiaSemMartes:MesSeptiembre     0.38997    0.09478   4.115 3.88e-05 ***
## DiaSemMiércoles:MesSeptiembre  0.30283    0.10311   2.937 0.003315 **
## DiaSemJueves:MesSeptiembre     0.37928    0.10434   3.635 0.000278 ***
## DiaSemViernes:MesSeptiembre    0.66965    0.09963   6.721 1.80e-11 ***
## DiaSemSábado:MesSeptiembre     1.05088    0.09405  11.173  < 2e-16 ***
## DiaSemDomingo:MesSeptiembre    0.64888    0.08962   7.240 4.48e-13 ***
## DiaSemMartes:MesOctubre       -0.21881    0.12492  -1.752 0.079841 .
## DiaSemMiércoles:MesOctubre     0.28657    0.11332   2.529 0.011443 *
## DiaSemJueves:MesOctubre        0.67686    0.10350   6.539 6.17e-11 ***
## DiaSemViernes:MesOctubre       1.06819    0.09851  10.843  < 2e-16 ***
## DiaSemSábado:MesOctubre        0.98318    0.10256   9.587  < 2e-16 ***
## DiaSemDomingo:MesOctubre       0.33785    0.10400   3.249 0.001160 **
## DiaSemMartes:MesNoviembre     -0.34498    0.09588  -3.598 0.000320 ***
## DiaSemMiércoles:MesNoviembre   0.13068    0.09046   1.445 0.148576
```

```
## DiaSemJueves:MesNoviembre      0.22571    0.08866   2.546 0.010907 *
## DiaSemViernes:MesNoviembre     0.05295    0.09400   0.563 0.573185
## DiaSemSábado:MesNoviembre      0.33346    0.08713   3.827 0.000130 ***
## DiaSemDomingo:MesNoviembre     0.03064    0.07914   0.387 0.698580
## DiaSemMartes:MesDiciembre      0.21638    0.07827   2.764 0.005702 **
## DiaSemMiércoles:MesDiciembre   0.32700    0.08053   4.061 4.89e-05 ***
## DiaSemJueves:MesDiciembre      0.68867    0.07840   8.785  < 2e-16 ***
## DiaSemViernes:MesDiciembre     0.50464    0.08441   5.978 2.25e-09 ***
## DiaSemSábado:MesDiciembre      0.53177    0.08294   6.412 1.44e-10 ***
## DiaSemDomingo:MesDiciembre     0.04291    0.07934   0.541 0.588636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 51393  on 942  degrees of freedom
## Residual deviance: 29936  on 856  degrees of freedom
## AIC: 34403
##
## Number of Fisher Scoring iterations: 10
```

Como se puede ver, no influyen los festivos en el modelo, es decir, para un p-valor de 0.05, la covariable *DiaLaborable* es no significativa, al igual que algunos coeficientes de la interacción.
Al contrario de esta, el modelo identifica claramente a la covariable *Carnaval* ya que es significativa y por tanto, influye en el modelo. Esta covariable ayuda a explicar el incremento de ventas que se tiene durante los primeros meses del año.

Por otro lado, parece que la interacción entre las variables *DiaSem* y *Mes* es significativa, es decir, la asociación entre ambas variables varía dependiendo de sus niveles.

También aparece en la salida las estimaciones de los coeficientes de regresión para cada una de las variables que se han incluido en el modelo de Regresión de Poisson.

A nivel interpretativo, se deben tener en cuenta las categorías de referencia para las variables cualitativas: DiaSem: Lunes, Mes: Enero, DiaLaborable: 0, Carnaval: 0 y la interacción Lunes:Enero.

Además, se puede evaluar tanto el ajuste del modelo a los datos como la complejidad de éste gracias al AIC, índice que cuánto más pequeño sea, mejor será el ajuste. En este modelo se tiene un AIC de 34403, un valor excesivamente alto, por lo que se puede concluir que no es un buen ajuste.

Por último, se obtienen las predicciones mediante la función *predict* y se contrastan estos resultados tanto gráfica como numéricamente:

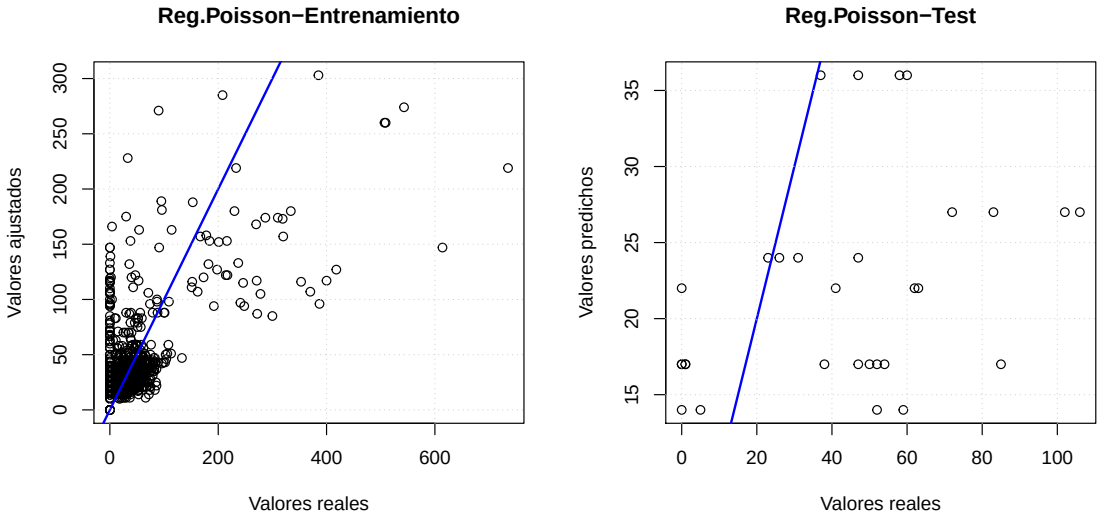


Figura 2.5: Modelo de Regresión de Poisson. Modelo con interacción.

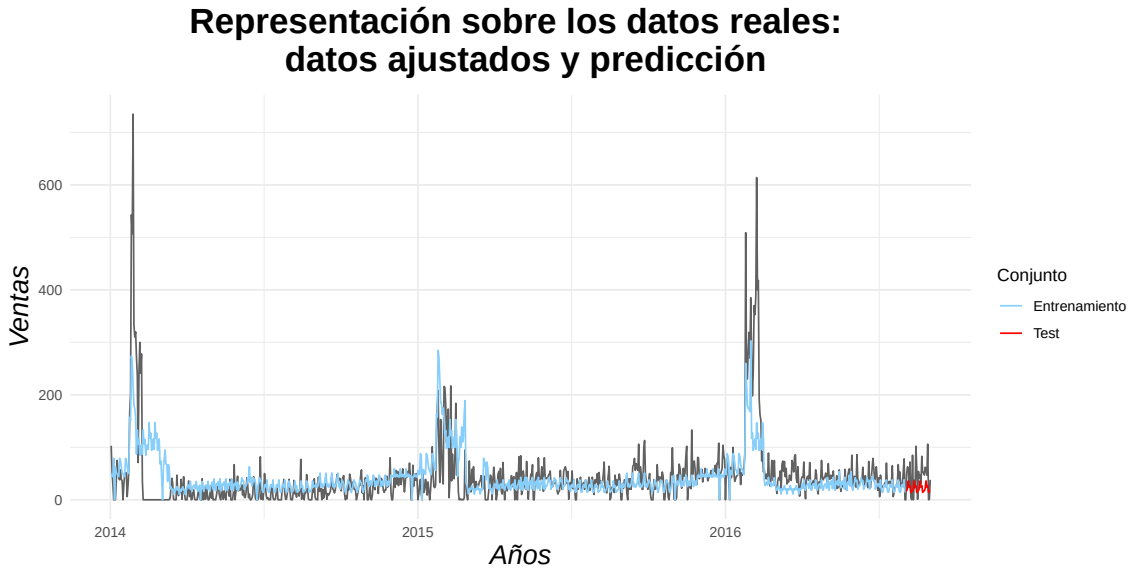


Figura 2.6: Modelo de Regresión de Poisson. Modelo con interacción.

```
##                              LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.0000000 50.93735 0.3970711
## Conjunto_test          0.3356464 35.30911 0.1332997
```

Tanto en la primera como en la segunda gráfica se representan los valores observados respecto a los estimados o ajustados, para los datos del conjunto de entrenamiento y para los del conjunto test, respectivamente. Para que un modelo sea bueno, todos los puntos estarían sobre la línea de regresión ajustada, es decir, a mayor varianza explicada por el modelo de regresión, más cerca estarán dichos puntos de la línea de regresión ajustada. Este modelo explica el 39'71 % de la variabilidad, para los datos de entrenamiento; y tan solo el 13'32 % para los datos del conjunto test, y como se observa, los datos no se ajustan bien a la recta. Esto puede ser debido a la poca cantidad de valores que se tienen en el conjunto test.

El p-valor obtenido para el conjunto test es de 0.336, que a diferencia del obtenido para el conjunto de entrenamiento, este acepta que los datos se distribuyen de forma independiente, lo que supone la existencia de correlación entre las variables. Además, ha disminuido el valor de $RMSE = 35.3091112$ aunque la calidad del modelo para replicar los resultados es mucho menor, como ya se ha comentado.

La representación sobre los datos reales muestra que ambos conjuntos oscilan sobre el rango de dichos datos. Además, el conjunto de entrenamiento recoge el gran aumento de ventas de 2014/2015. En general, ambos conjuntos imitan el comportamiento de los valores reales, aunque podría mejorar.

Se prueba con un modelo sin interacción.

Modelo sin interacción

Como se tuvo que la covariable *DiaLaborable* resultaba ser no significativa, se realiza el nuevo modelo sin interacción, sin este predictor.

```
modeloP2=glm(CantidadDiaria ~  PrecioMedioDiario + DiaSem  + Mes +
             Carnaval,family = "poisson", data = datos[indient,])
summary(modeloP2)
```

```
##
## Call:
## glm(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem + Mes +
##     Carnaval, family = "poisson", data = datos[indient, ])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -18.0745   -4.1752   -0.3023    2.4217   31.3442
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         6.52272    0.03154 206.802  < 2e-16 ***
## PrecioMedioDiario  -1.30914    0.01880 -69.647  < 2e-16 ***
## DiaSemMartes       -0.04243    0.01949  -2.177  0.02950 *
## DiaSemMiércoles    -0.08054    0.01959  -4.112 3.93e-05 ***
## DiaSemJueves        0.05434    0.01895   2.867  0.00414 **
## DiaSemViernes       0.10526    0.01860   5.659 1.52e-08 ***
## DiaSemSábado       -0.01874    0.01929  -0.972  0.33120
## DiaSemDomingo       0.34011    0.01776  19.153  < 2e-16 ***
## MesFebrero         -0.59324    0.01938 -30.615  < 2e-16 ***
## MesMarzo           -1.15074    0.02245 -51.264  < 2e-16 ***
## MesAbril           -0.83501    0.02476 -33.724  < 2e-16 ***
## MesMayo            -0.72766    0.02323 -31.322  < 2e-16 ***
## MesJunio           -0.71001    0.02290 -31.002  < 2e-16 ***
## MesJulio           -0.92425    0.02487 -37.165  < 2e-16 ***
## MesAgosto          -1.01227    0.02965 -34.138  < 2e-16 ***
## MesSeptiembre      -0.63797    0.02633 -24.229  < 2e-16 ***
## MesOctubre         -0.80671    0.02761 -29.217  < 2e-16 ***
```

```
## MesNoviembre       -0.56771    0.02553 -22.241  < 2e-16 ***
## MesDiciembre       -0.29809    0.02315 -12.875  < 2e-16 ***
## CarnavalSiCarnaval  0.05219    0.01883   2.773  0.00556 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 51393  on 942  degrees of freedom
## Residual deviance: 32091  on 923  degrees of freedom
## AIC: 36425
##
## Number of Fisher Scoring iterations: 6
```

Al suponer que no existe interacción, se tiene un modelo con todas las variables significativas y un valor del AIC de 36425 que ha aumentado con respecto al anterior, pero al tener anteriormente un modelo de baja calidad, no se tiene en cuenta este aumento.

Para la interpretación de los coeficientes, se debe tener en cuenta las categorías de referencia para las variables cualitativas: DiaSem: Lunes, Mes: Enero y Carnaval: 0

A nivel interpretativo, se tiene que un aumento del *PrecioMedioDiario* produce una disminución del valor esperado del número de ventas. Con respecto a las variables explicativas definidas, solo *CarnavalSi*, produce un aumento en el valor esperado de dichas ventas.

Como antes, se obtienen las predicciones, mediante la función *predict* y se contrastan estos resultados gráfica y numéricamente:

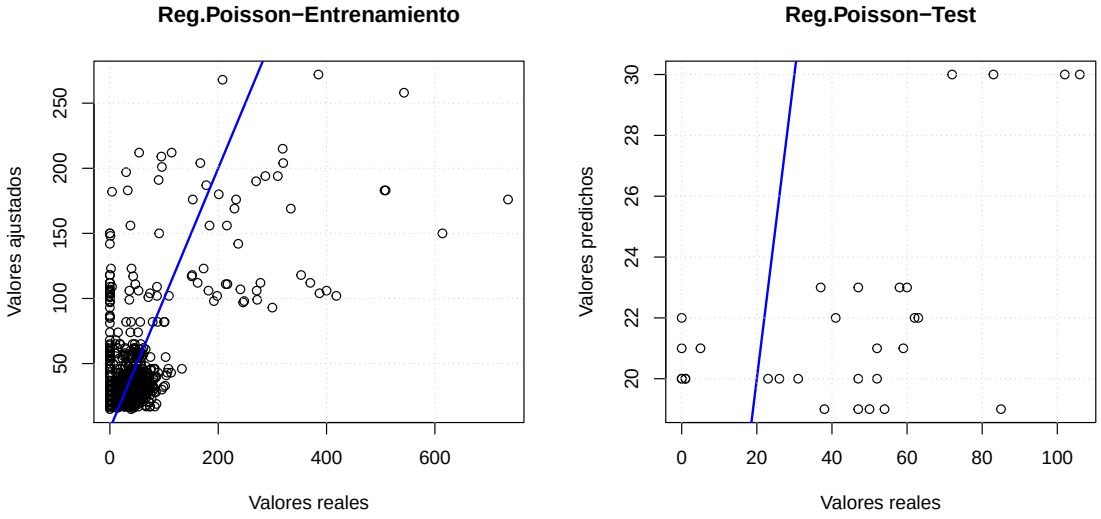


Figura 2.7: Modelo de Regresión de Poisson. Modelo sin interacción.

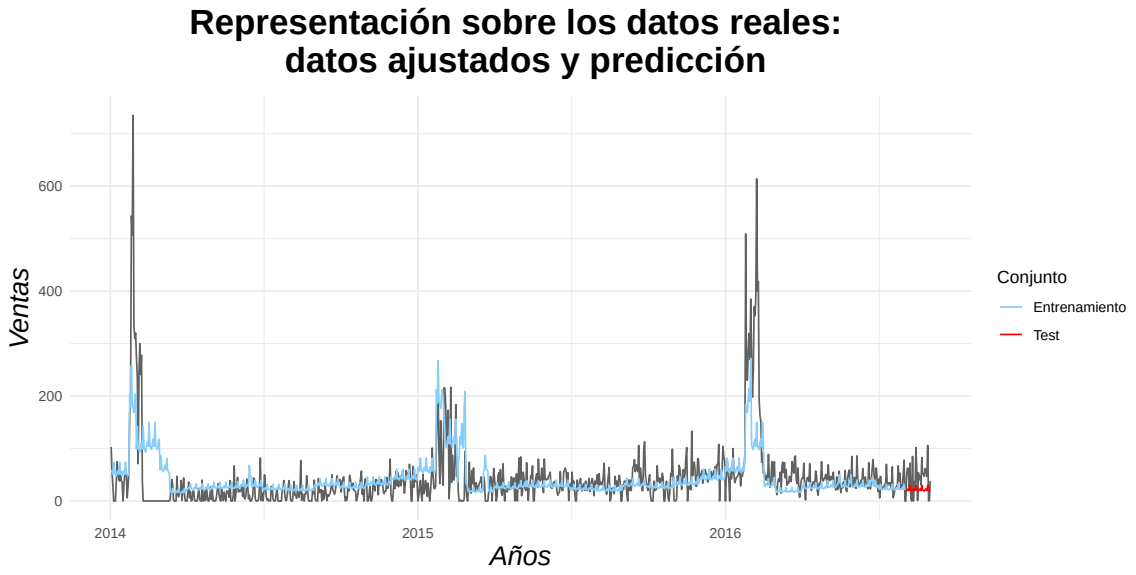


Figura 2.8: Modelo de Regresión de Poisson. Modelo sin interacción.

```
##                             LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.0000000 52.57640 0.3564272
## Conjunto_test          0.3283883 35.23966 0.3695807
```

Este modelo obtiene mejores predicciones en comparativa con el anterior, al tener un valor de $R^2 = 0.3696$, para el conjunto test. A pesar de esto, la variabilidad explicada no es demasiado alta y se aprecia, tanto en los ajustes de la recta de regresión como en la representación con los datos reales, que sigue sin mostrar el gran volumen de ventas de 2016, ya que al igual que el modelo anterior, para 2014 las muestra pero más uniformes.

Se prueba con un modelo de regresión Binomial negativa.

2.1.3. Modelo de regresión Binomial negativa

Ahora se va a modelizar este conjunto de datos de acuerdo a una binomial negativa como miembro de la familia de modelos lineales generalizados. Se va a utilizar la función glm.nb del paquete *MASS*, mediante el cual se estiman los parámetros de dicho modelo. Las predicciones se van a obtener igual que en el modelo de poisson, se ajustan los datos al conjunto de entrenamiento y posteriormente se observará si el modelo puede llegar a conseguir buenas predicciones en el conjunto test. Las variables predictivas a tener en cuenta, van a seguir siendo las mismas: *PrecioMedioDiario*, *dia*, *Mes*, *DiaLaborable* y *Carnaval*

```
library("MASS")
modeloBN1=glm.nb(CantidadDiaria~PrecioMedioDiario + DiaSem + Mes +
        DiaLaborable + Carnaval, data = datos[indient,])
summary(modeloBN1)
```

```
##
## Call:
## glm.nb(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem +
##     Mes + DiaLaborable + Carnaval, data = datos[indient, ], init.theta = 0.8182081
##     link = log)
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.94233  -0.63317  -0.02656   0.33708   2.22304
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -2.280e+01  5.571e+04   0.000 0.999673
## PrecioMedioDiario        -1.152e+00  1.875e-01  -6.145 8.01e-10 ***
## DiaSemMartes             -2.651e-02  1.381e-01  -0.192 0.847823
## DiaSemMiércoles           3.949e-02  1.376e-01   0.287 0.774046
## DiaSemJueves              1.867e-01  1.373e-01   1.360 0.173823
## DiaSemViernes             3.073e-01  1.374e-01   2.236 0.025378 *
## DiaSemSábado              1.456e-01  1.371e-01   1.062 0.288094
## DiaSemDomingo             4.504e-01  1.367e-01   3.294 0.000989 ***
## MesFebrero               -2.362e-01  2.162e-01  -1.093 0.274600
## MesMarzo                 -7.025e-01  1.669e-01  -4.210 2.56e-05 ***
## MesAbril                 -7.492e-01  1.731e-01  -4.329 1.50e-05 ***
## MesMayo                  -6.510e-01  1.705e-01  -3.819 0.000134 ***
## MesJunio                 -5.830e-01  1.711e-01  -3.407 0.000657 ***
## MesJulio                 -8.519e-01  1.705e-01  -4.997 5.83e-07 ***
## MesAgosto                -9.291e-01  1.887e-01  -4.923 8.51e-07 ***
## MesSeptiembre            -5.398e-01  1.908e-01  -2.829 0.004666 **
## MesOctubre               -7.300e-01  1.891e-01  -3.860 0.000113 ***
## MesNoviembre             -4.426e-01  1.915e-01  -2.311 0.020816 *
## MesDiciembre             -1.424e-01  1.904e-01  -0.748 0.454619
## DiaLaborableSiLaborable   2.883e+01  5.571e+04   0.001 0.999587
## CarnavalSiCarnaval       -5.468e-02  2.163e-01  -0.253 0.800439
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.8182) family taken to be 1)
##
##     Null deviance: 1468.0  on 942  degrees of freedom
## Residual deviance: 1137.1  on 922  degrees of freedom
## AIC: 8623.1
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.8182
##           Std. Err.:  0.0396
##
##  2 x log-likelihood:  -8579.1370
```

Además de la estimación de los parámetros, esta función ofrece una estimación previa del parámetro de dispersión θ mediante el método de máxima verosimilitud, pero es importante saber que el valor de $\hat{\theta}$ no es 0'8182 sino el de su inversa, es decir, 1.222. También ofrece, al igual que el modelo de regresión de Poisson, estadísticos adicionales

como el estadístico desviación o el criterio AIC.

Según este modelo, se pueden suponer nulos los coeficientes de regresión para las variables *DíaLaboral* y *Carnaval* al no ser significativos. Esto se observa en el valor que toma cada pvalor, 0.999 y 0.800 respectivamente. Luego, ni los días laborables ni el carnaval van a influir en la cantidad de ventas y por tanto, se pueden eliminar del modelo.

Para la interpretación de los coeficientes, al igual que en el modelo de Poisson, se deben tener en cuenta las categorías de referencia para las variables cualitativas: DiaSem: Lunes, Mes: Enero, DiaLaborable: 0 y Carnaval: 0.

Modelo sin DiaLaborable ni Carnaval

```
modeloBN2=glm.nb(CantidadDiaria~PrecioMedioDiario + DiaSem + Mes,
          data = datos[indient,])
summary(modeloBN2)
```

```
##
## Call:
## glm.nb(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem +
##     Mes, data = datos[indient, ], init.theta = 0.778229711, link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.87535  -0.64825  -0.02644   0.33063   2.24990
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        5.98773    0.33463  17.893  < 2e-16 ***
## PrecioMedioDiario -1.15653    0.18075  -6.398 1.57e-10 ***
## DiaSemMartes      -0.03453    0.14022  -0.246 0.805496
## DiaSemMiércoles    0.04721    0.14016   0.337 0.736263
## DiaSemJueves       0.19447    0.13984   1.391 0.164326
## DiaSemViernes      0.30517    0.13974   2.184 0.028978 *
## DiaSemSábado       0.15973    0.13980   1.143 0.253218
## DiaSemDomingo      0.46365    0.13956   3.322 0.000893 ***
## MesFebrero        -0.23869    0.19314  -1.236 0.216519
## MesMarzo          -0.67477    0.16859  -4.002 6.27e-05 ***
## MesAbril          -0.72252    0.17504  -4.128 3.67e-05 ***
## MesMayo           -0.61300    0.17288  -3.546 0.000391 ***
## MesJunio          -0.55587    0.17319  -3.210 0.001329 **
## MesJulio          -0.81434    0.17290  -4.710 2.48e-06 ***
## MesAgosto         -0.89101    0.19174  -4.647 3.37e-06 ***
## MesSeptiembre     -0.50140    0.19390  -2.586 0.009712 **
## MesOctubre        -0.69236    0.19216  -3.603 0.000315 ***
## MesNoviembre      -0.42178    0.19374  -2.177 0.029478 *
## MesDiciembre      -0.13615    0.19166  -0.710 0.477499
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for Negative Binomial(0.7782) family taken to be 1)
##
##     Null deviance: 1407.4  on 942  degrees of freedom
## Residual deviance: 1147.8  on 924  degrees of freedom
## AIC: 8676.2
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.7782
##           Std. Err.:  0.0375
##
##  2 x log-likelihood:  -8636.1880
```

En el modelo anterior, se tenía un AIC = 8623.1. En este modelo, el valor del AIC se ha incrementado, aunque mínimamente, por lo que su complejidad no se ha visto apenas afectada.

En este modelo, las categorías de referencia a tener en cuenta para las variables cualitativas son: DiaSem: Lunes y Mes: Enero.

A nivel interpretativo, el aumento del *PrecioMedioDiario* produce un descenso del número esperado de ventas. En función a las variables *dummys* solo *DiaSemViernes* y *DiaSemDomingo* aumentan el valor esperado de las ventas, aunque en menor medida que *DiaSemLunes*. Para los demás días se consideran nulos sus coeficientes, al igual que *MesFebrero* y *MesDiciembre*. El resto de meses producen un descenso del número esperado de ventas.

Como se realizó antes, el valor esperado del parámetro de dispersión θ es $1/0.7782 = 1.2850$.

El estadístico desviación D, bajo las hipótesis del modelo correcto, sigue una distribución chi-cuadrado con 924 grados de libertad y un valor de 1147'8. Para detectar sobredispersión en los datos se evalúa la siguiente relación:

$$\frac{D}{gl} = \frac{1147.8}{924} = 1.242 > 1$$

lo que indica sobredispersión en los datos.

Se procede a realizar el ajuste sobre el conjunto de entrenamiento y la predicción sobre el conjunto test, al igual que en el modelo de Poisson, mediante la función *predict*. También se pueden contrastar los resultados numéricamente:

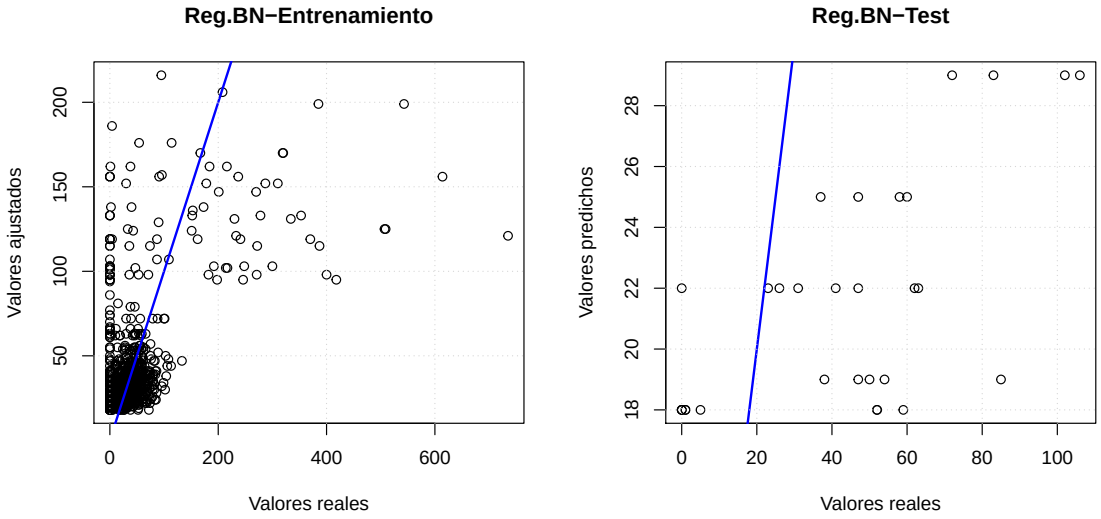


Figura 2.9: Modelo de regresión Binomial negativa. Modelo sin *DiaLaborable* ni *Carnaval*.

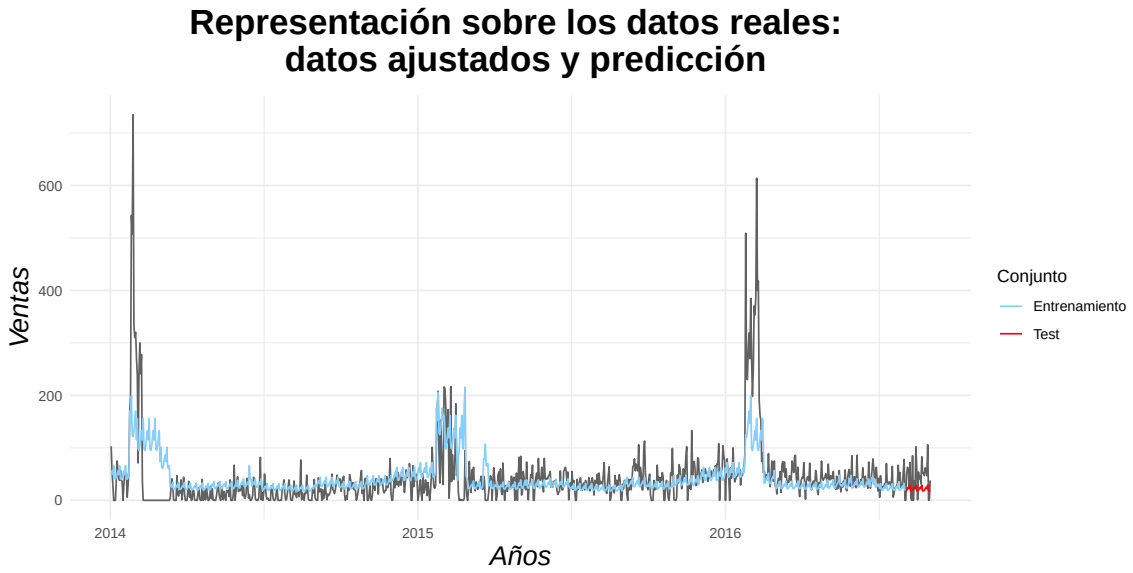


Figura 2.10: Modelo de regresión Binomial negativa. Modelo sin *DiaLaborable* ni *Carnaval*.

```
##                              LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.0000000 54.90988 0.2990190
## Conjunto_test          0.2991143 35.21979 0.4026964
```

El modelo de regresión Binomial negativa, para los datos del conjunto de entrenamiento, explica tan solo el 29.90 % de la variabilidad, mientras que para el cojunto test, explica el 40.27 %, además de haber disminuido la cantidad de error entre los dos conjunto de datos, reales y conjunto test, 35.22. Aún así, en la última gráfica se observa que los datos del conjunto test apenas muestran el comportamiento real. Esto, como se comentó en el modelo de Poisson, puede ser porque este conjuto tan solo está formado por 30 observaciones.

2.1.4. Modelo inflado con ceros y modelo en dos partes

Para tratar el tema de la abundancia de valores nulos, es decir, para intentar explicar la causa de las no ventas, se van a utilizar los predictores *PrecioMedioDiario*, *DiaSem*, *Mes* y *Carnaval*.

Inicialmente se van a suponer todos los ceros iguales, por tanto, se va a utilizar un modelo en dos partes (hurdle), mediante el cual se va a estimar la probabilidad de no ser cero. Para ello, se utiliza la libreria *pscl* y es necesario introducir dos fórmulas, la primera, que corresponde al modelo lineal generalizado que se vaya a utilizar y la segunda al Binomial.

En primer lugar se utiliza el **modelo de regresión de Poisson**:

```
library(pscl)
mhurdleP=hurdle(CantidadDiaria ~ PrecioMedioDiario + DiaSem + Mes +
                  Carnaval |PrecioMedioDiario + DiaSem + Mes +
                  Carnaval  , dist="poisson",
                  data = datos)
summary(mhurdleP)
```

```
##
## Call:
## hurdle(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem + Mes +
##     Carnaval | PrecioMedioDiario + DiaSem + Mes + Carnaval, data = datos,
##     dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -5.1658 -1.3273 -0.1041  1.1194  9.0045
##
## Count model coefficients (truncated poisson with log link):
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         7.00050    0.03146 222.495  < 2e-16 ***
## PrecioMedioDiario  -1.55067    0.01979 -78.374  < 2e-16 ***
## DiaSemMartes       -0.04329    0.01935  -2.237 0.025298 *
## DiaSemMiércoles    -0.16686    0.01925  -8.669  < 2e-16 ***
## DiaSemJueves       -0.06175    0.01876  -3.291 0.000999 ***
## DiaSemViernes      -0.07261    0.01837  -3.952 7.74e-05 ***
## DiaSemSábado       -0.19539    0.01905 -10.257  < 2e-16 ***
## DiaSemDomingo       0.23020    0.01748  13.167  < 2e-16 ***
## MesFebrero         -0.44019    0.01997 -22.046  < 2e-16 ***
## MesMarzo           -0.51110    0.02419 -21.132  < 2e-16 ***
## MesAbril           -0.57990    0.02564 -22.621  < 2e-16 ***
## MesMayo            -0.53541    0.02409 -22.229  < 2e-16 ***
## MesJunio           -0.58014    0.02384 -24.339  < 2e-16 ***
## MesJulio           -0.71505    0.02574 -27.782  < 2e-16 ***
## MesAgosto          -0.50429    0.02436 -20.703  < 2e-16 ***
## MesSeptiembre      -0.50608    0.02713 -18.652  < 2e-16 ***
## MesOctubre         -0.58075    0.02841 -20.441  < 2e-16 ***
## MesNoviembre       -0.38738    0.02636 -14.693  < 2e-16 ***
```

```
## MesDiciembre       -0.15114    0.02407  -6.280 3.38e-10 ***
## CarnavalSiCarnaval  0.18133    0.01989   9.118  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.53507    0.79574  -1.929  0.05372 .
## PrecioMedioDiario   1.95505    0.47426   4.122 3.75e-05 ***
## DiaSemMartes        0.06242    0.29351   0.213  0.83158
## DiaSemMiércoles     0.82731    0.33059   2.503  0.01233 *
## DiaSemJueves        1.06055    0.35012   3.029  0.00245 **
## DiaSemViernes       1.82567    0.42802   4.265 2.00e-05 ***
## DiaSemSábado        1.51770    0.38928   3.899 9.67e-05 ***
## DiaSemDomingo       1.04649    0.34566   3.027  0.00247 **
## MesFebrero         -0.05561    0.54303  -0.102  0.91843
## MesMarzo           -1.23313    0.47498  -2.596  0.00943 **
## MesAbril           -1.33427    0.52459  -2.543  0.01098 *
## MesMayo            -0.89019    0.54266  -1.640  0.10092
## MesJunio            0.08120    0.62865   0.129  0.89722
## MesJulio           -0.99343    0.53535  -1.856  0.06350 .
## MesAgosto          -1.23972    0.52009  -2.384  0.01714 *
## MesSeptiembre      -0.16960    0.67899  -0.250  0.80275
## MesOctubre         -1.15689    0.56843  -2.035  0.04183 *
## MesNoviembre       -0.67234    0.61528  -1.093  0.27450
## MesDiciembre       -0.38791    0.64062  -0.606  0.54483
## CarnavalSiCarnaval -0.63306    0.46509  -1.361  0.17346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 28
## Log-likelihood: -1.116e+04 on 40 Df
```

Se observa que, tanto el precio como los días de la semana son significativos.

Ahora, se supone que existen dos tipos de ceros, los falsos y los auténticos. Para ello, se aplica el modelo inflado por cero con las mismas variables y se utiliza la función *zeroinfl* que estima la probabilidad de que la variable de conteo valga 0.

```
mzeroinflP <- zeroinfl(CantidadDiaria ~  PrecioMedioDiario +
                         DiaSem + Mes + Carnaval|PrecioMedioDiario +
                          DiaSem + Mes + Carnaval,
                         data = datos, dist = "poisson")
summary(mzeroinflP)
```

```
##
## Call:
## zeroinfl(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem + Mes +
##     Carnaval | PrecioMedioDiario + DiaSem + Mes + Carnaval, data = datos,
##     dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -5.1658 -1.3273 -0.1041  1.1194  9.0045
##
## Count model coefficients (poisson with log link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          7.00050    0.03146 222.495  < 2e-16 ***
## PrecioMedioDiario   -1.55067    0.01979 -78.374  < 2e-16 ***
## DiaSemMartes        -0.04329    0.01935  -2.237 0.025298 *
## DiaSemMiércoles     -0.16686    0.01925  -8.669  < 2e-16 ***
## DiaSemJueves        -0.06175    0.01876  -3.291 0.000999 ***
## DiaSemViernes       -0.07261    0.01837  -3.952 7.74e-05 ***
## DiaSemSábado        -0.19539    0.01905 -10.257  < 2e-16 ***
## DiaSemDomingo        0.23020    0.01748  13.167  < 2e-16 ***
## MesFebrero          -0.44019    0.01997 -22.046  < 2e-16 ***
## MesMarzo            -0.51110    0.02419 -21.131  < 2e-16 ***
## MesAbril            -0.57990    0.02564 -22.621  < 2e-16 ***
## MesMayo             -0.53541    0.02409 -22.229  < 2e-16 ***
## MesJunio            -0.58014    0.02384 -24.339  < 2e-16 ***
## MesJulio            -0.71505    0.02574 -27.782  < 2e-16 ***
## MesAgosto           -0.50429    0.02436 -20.703  < 2e-16 ***
## MesSeptiembre       -0.50608    0.02713 -18.652  < 2e-16 ***
## MesOctubre          -0.58075    0.02841 -20.441  < 2e-16 ***
## MesNoviembre        -0.38738    0.02636 -14.693  < 2e-16 ***
## MesDiciembre        -0.15114    0.02407  -6.280 3.38e-10 ***
## CarnavalSiCarnaval   0.18133    0.01989   9.118  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.53507    0.79574   1.929  0.05372 .
## PrecioMedioDiario   -1.95505    0.47426  -4.122 3.75e-05 ***
## DiaSemMartes        -0.06242    0.29351  -0.213  0.83158
## DiaSemMiércoles     -0.82731    0.33059  -2.503  0.01233 *
## DiaSemJueves        -1.06055    0.35012  -3.029  0.00245 **
## DiaSemViernes       -1.82567    0.42802  -4.265 2.00e-05 ***
## DiaSemSábado        -1.51770    0.38928  -3.899 9.67e-05 ***
## DiaSemDomingo       -1.04649    0.34566  -3.027  0.00247 **
## MesFebrero           0.05561    0.54303   0.102  0.91843
## MesMarzo             1.23313    0.47498   2.596  0.00943 **
## MesAbril             1.33427    0.52459   2.543  0.01098 *
## MesMayo              0.89019    0.54266   1.640  0.10092
## MesJunio            -0.08120    0.62865  -0.129  0.89722
## MesJulio             0.99343    0.53535   1.856  0.06350 .
## MesAgosto            1.23972    0.52009   2.384  0.01714 *
## MesSeptiembre        0.16960    0.67899   0.250  0.80275
## MesOctubre           1.15689    0.56843   2.035  0.04183 *
## MesNoviembre         0.67234    0.61528   1.093  0.27450
## MesDiciembre         0.38791    0.64062   0.606  0.54483
## CarnavalSiCarnaval   0.63306    0.46509   1.361  0.17346
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 1
## Log-likelihood: -1.116e+04 on 40 Df
```

Se observa como, al igual que en el modelo en dos partes (Hurdle models), solo son significativos el precio y los días de la semana.

Con el objetivo de ver que modelo se adapta mejor a los datos se comparan los dos modelos utilizados para abordar el problema de exceso de ceros. Esto se realiza mediante el *test de Vuong*, que compara la idoneidad de los modelos:

```
vuong(mhurdleP,mzeroinflP)
```

```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##                Vuong z-statistic             H_A p-value
## Raw               -1.177423e-05 model2 > model1     0.5
## AIC-corrected     -1.177423e-05 model2 > model1     0.5
## BIC-corrected     -1.177423e-05 model2 > model1     0.5
```

Donde se tiene un $pvalor = 0.5 > 0.05$, por tanto los modelos son equivalentes.

Ahora, se realiza de nuevo todo lo anterior pero con un **modelo de regresión Binomial negativo**:

```
mhurdleBN=hurdle(CantidadDiaria ~ PrecioMedioDiario + DiaSem +
                   Mes + Carnaval| PrecioMedioDiario + DiaSem +
                   Mes + Carnaval , dist="negbin",
                   data = datos)
summary(mhurdleBN)
```

```
##
## Call:
## hurdle(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem + Mes +
##     Carnaval | PrecioMedioDiario + DiaSem + Mes + Carnaval, data = datos,
##     dist = "negbin")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.47924 -0.68176 -0.04615  0.52627  4.75492
##
## Count model coefficients (truncated negbin with log link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         6.940307   0.220732  31.442  < 2e-16 ***
## PrecioMedioDiario  -1.636661   0.127486 -12.838  < 2e-16 ***
## DiaSemMartes       -0.079648   0.091707  -0.869 0.385119
## DiaSemMiércoles    -0.055297   0.089026  -0.621 0.534516
## DiaSemJueves        0.036190   0.088266   0.410 0.681796
## DiaSemViernes       0.077417   0.087333   0.886 0.375369
```

```
## DiaSemSábado       -0.043924   0.087746  -0.501 0.616668
## DiaSemDomingo       0.334279   0.088082   3.795 0.000148 ***
## MesFebrero         -0.211198   0.145547  -1.451 0.146761
## MesMarzo           -0.331371   0.112797  -2.938 0.003306 **
## MesAbril           -0.465115   0.111877  -4.157 3.22e-05 ***
## MesMayo            -0.415321   0.109034  -3.809 0.000139 ***
## MesJunio           -0.428501   0.107878  -3.972 7.12e-05 ***
## MesJulio           -0.597223   0.109722  -5.443 5.24e-08 ***
## MesAgosto          -0.387302   0.110356  -3.510 0.000449 ***
## MesSeptiembre      -0.367824   0.119996  -3.065 0.002174 **
## MesOctubre         -0.451165   0.121873  -3.702 0.000214 ***
## MesNoviembre       -0.254731   0.120847  -2.108 0.035042 *
## MesDiciembre        0.005403   0.118796   0.045 0.963724
## CarnavalSiCarnaval  0.095070   0.160757   0.591 0.554260
## Log(theta)          0.889142   0.052901  16.808  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.53507    0.79574  -1.929  0.05372 .
## PrecioMedioDiario   1.95505    0.47426   4.122 3.75e-05 ***
## DiaSemMartes        0.06242    0.29351   0.213  0.83158
## DiaSemMiércoles     0.82731    0.33059   2.503  0.01233 *
## DiaSemJueves        1.06055    0.35012   3.029  0.00245 **
## DiaSemViernes       1.82567    0.42802   4.265 2.00e-05 ***
## DiaSemSábado        1.51770    0.38928   3.899 9.67e-05 ***
## DiaSemDomingo       1.04649    0.34566   3.027  0.00247 **
## MesFebrero         -0.05561    0.54303  -0.102  0.91843
## MesMarzo           -1.23313    0.47498  -2.596  0.00943 **
## MesAbril           -1.33427    0.52459  -2.543  0.01098 *
## MesMayo            -0.89019    0.54266  -1.640  0.10092
## MesJunio            0.08120    0.62865   0.129  0.89722
## MesJulio           -0.99343    0.53535  -1.856  0.06350 .
## MesAgosto          -1.23972    0.52009  -2.384  0.01714 *
## MesSeptiembre      -0.16960    0.67899  -0.250  0.80275
## MesOctubre         -1.15689    0.56843  -2.035  0.04183 *
## MesNoviembre       -0.67234    0.61528  -1.093  0.27450
## MesDiciembre       -0.38791    0.64062  -0.606  0.54483
## CarnavalSiCarnaval -0.63306    0.46509  -1.361  0.17346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 2.433
## Number of iterations in BFGS optimization: 30
## Log-likelihood: -4163 on 41 Df
```

Con este modelo, se vuelven a tener las mismas variables explicativas significativas, *PrecioMedioDiario* y *DiaSem*.

El modelo inflado por cero con las mismas variables:

```
 mzeroinflBN=zeroinfl(CantidadDiaria ~ PrecioMedioDiario + DiaSem +
                  Mes + Carnaval| PrecioMedioDiario + DiaSem +
                  Mes + Carnaval, data = datos, dist = "negbin")
summary(mzeroinflBN)
```

```
##
## Call:
## zeroinfl(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem + Mes +
##     Carnaval | PrecioMedioDiario + DiaSem + Mes + Carnaval, data = datos,
##     dist = "negbin")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.47944 -0.68181 -0.04606  0.52622  4.75504
##
## Count model coefficients (negbin with log link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         6.940966   0.220675  31.453  < 2e-16 ***
## PrecioMedioDiario  -1.637815   0.127484 -12.847  < 2e-16 ***
## DiaSemMartes       -0.079702   0.091714  -0.869 0.384830
## DiaSemMiércoles    -0.055291   0.089034  -0.621 0.534593
## DiaSemJueves        0.036395   0.088269   0.412 0.680102
## DiaSemViernes       0.077948   0.087331   0.893 0.372091
## DiaSemSábado       -0.043752   0.087749  -0.499 0.618057
## DiaSemDomingo       0.334454   0.088084   3.797 0.000146 ***
## MesFebrero         -0.210002   0.145510  -1.443 0.148959
## MesMarzo           -0.329329   0.112828  -2.919 0.003513 **
## MesAbril           -0.464356   0.111949  -4.148 3.35e-05 ***
## MesMayo            -0.414424   0.109100  -3.799 0.000146 ***
## MesJunio           -0.426552   0.107916  -3.953 7.73e-05 ***
## MesJulio           -0.595706   0.109774  -5.427 5.74e-08 ***
## MesAgosto          -0.385582   0.110400  -3.493 0.000478 ***
## MesSeptiembre      -0.366432   0.120045  -3.052 0.002270 **
## MesOctubre         -0.449515   0.121910  -3.687 0.000227 ***
## MesNoviembre       -0.253296   0.120894  -2.095 0.036154 *
## MesDiciembre        0.006593   0.118857   0.055 0.955764
## CarnavalSiCarnaval  0.094680   0.160718   0.589 0.555790
## Log(theta)          0.889228   0.052901  16.809  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.55139    0.79850   1.943 0.052029 .
## PrecioMedioDiario  -1.97587    0.47820  -4.132 3.60e-05 ***
## DiaSemMartes       -0.06370    0.29536  -0.216 0.829246
## DiaSemMiércoles    -0.83551    0.33384  -2.503 0.012325 *
## DiaSemJueves       -1.06738    0.35350  -3.019 0.002532 **
## DiaSemViernes      -1.83458    0.43295  -4.237 2.26e-05 ***
## DiaSemSábado       -1.53354    0.39527  -3.880 0.000105 ***
```

```
## DiaSemDomingo      -1.04736    0.34768  -3.012 0.002591 **
## MesFebrero          0.06574    0.54636   0.120 0.904223
## MesMarzo            1.25220    0.48165   2.600 0.009328 **
## MesAbril            1.34637    0.53424   2.520 0.011731 *
## MesMayo             0.89929    0.55275   1.627 0.103748
## MesJunio           -0.07232    0.63968  -0.113 0.909989
## MesJulio            1.00359    0.54541   1.840 0.065760 .
## MesAgosto           1.25999    0.52859   2.384 0.017141 *
## MesSeptiembre       0.17479    0.69262   0.252 0.800758
## MesOctubre          1.17501    0.57728   2.035 0.041808 *
## MesNoviembre        0.68899    0.62446   1.103 0.269875
## MesDiciembre        0.40468    0.64988   0.623 0.533478
## CarnavalSiCarnaval  0.63329    0.46576   1.360 0.173926
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 2.4333
## Number of iterations in BFGS optimization: 35
## Log-likelihood: -4163 on 41 Df
```

Al igual que en los anteriores, tanto el precio como los días de la semana son significativos.

Mediante el *test de Vuong*, se compara la idoneidad de los modelos:

```
vuong(mhurdleBN,mzeroinflBN)
```

```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##                Vuong z-statistic             H_A p-value
## Raw                    0.1290491 model1 > model2 0.44866
## AIC-corrected          0.1290491 model1 > model2 0.44866
## BIC-corrected          0.1290491 model1 > model2 0.44866
```

Se vuelve a aceptar la hipótesis nula, los modelos son equivalentes, al obtener un pvalor mayor que 0.05, 0.447.

Por tanto, se puede concluir que la no existencia de ventas podría estar relacionada con el alto coste del producto o con el cierre de muchos días de dicho establecimiento.

2.1.5. Modelo de series temporales

Otro de los modelos a tener en cuenta eran los modelos de series temporales. Como ya se comentó, estos modelos tienen en cuenta la estructura temporal de los datos, a diferencia de los otros. Se sabe que los datos de los que se disponen no son estacionarios y por ello, se van a utilizar los modelos ARIMA.
Se va a modelizar la serie temporal utilizando la función *auto.arima* del paquete *forecast*, que proporciona un buen modelo capaz de realizar pronósticos, ya que evalúa entre todos los posibles modelos, al mejor, considerando diversos criterios como estacionariedad, estacionalidad o diferencias, entre otras.

Esta función modeliza la serie temporal a través una función lineal de un conjunto de variables predictoras y de un término error, ajustado de forma automática mediante un modelo ARIMA apropiado.
Para poder representar las componentes estacionales, ha sido necesario utilizar la función *fourier* de R que, mediante desarrollos trigonométricos ha facilitado la obtención de variables que respresenten estas componentes estacionales.

```
library(forecast)
```

Para aplicar el modelo de series temporales se eliminan las variables *dummys*. Para ello, se crea un nuevo conjunto de datos sin dichas variables.

```
datos2=datos[,c(1:5)]
datos2$DiaSem=as.numeric(datos2$DiaSem)
datos2$Mes=as.numeric(datos2$Mes)
```

La cantidad diaria de ventas de cierto producto se tiene recogida en distintos períodos de tiempo luego, en primer lugar, se va a estudiar la perioricidad mediante un análisis espectral.
Pero antes de proceder a dicho análisis, se va a representar la media de ventas respecto a distintos períodos de tiempo.

```
mfecha=format(datos2$Fecha,"%m")
afecha=format(datos2$Fecha,"%Y")
dfecha=format(datos2$Fecha,"%d") #DIA DEL MES
```

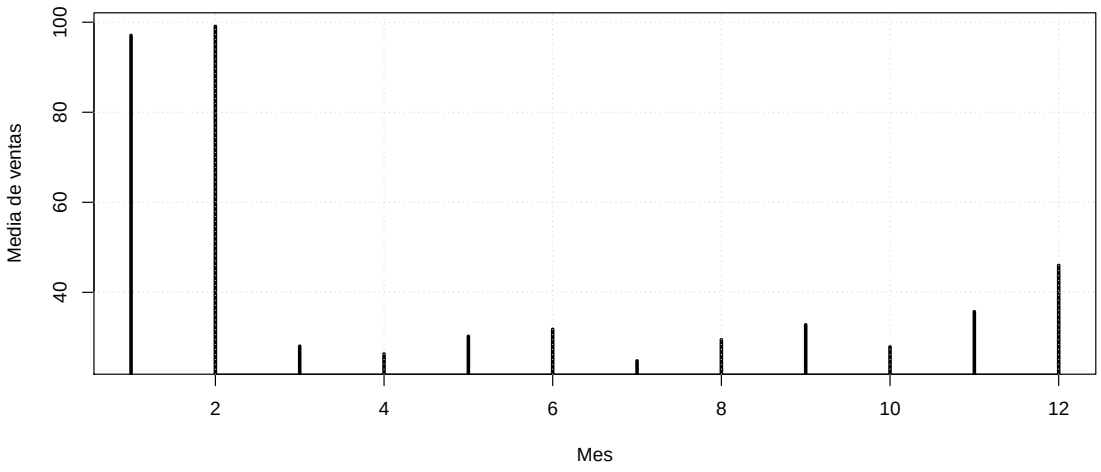


Figura 2.11: MST. Media de ventas respecto al mes

Ya se comentó la gran abundancia de ventas en enero(1) y febrero(2) ocasionada por el Carnaval.

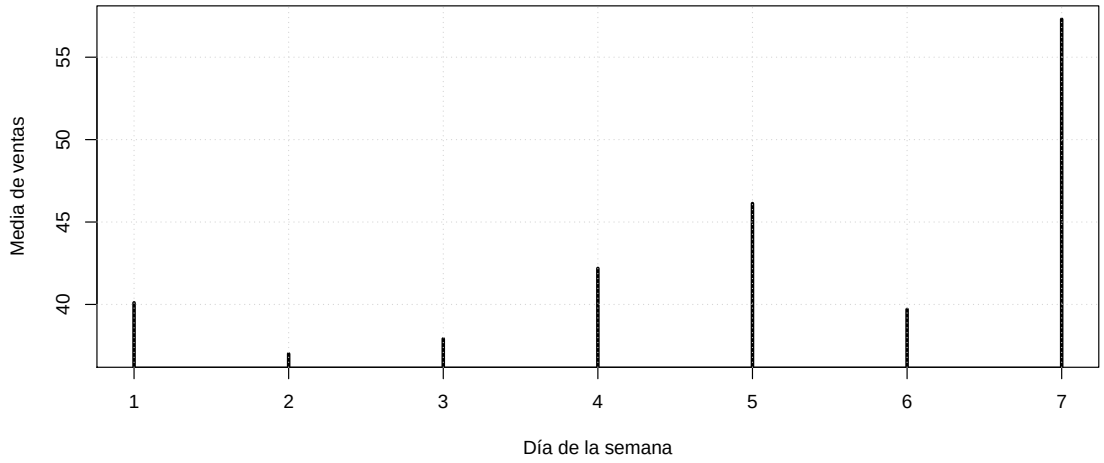


Figura 2.12: MST. Media de ventas respecto al día de la semana

Con una clara diferencia, se aprecia que los domingos(7) se da una media de ventas superior al resto de la semana, seguida del viernes(5). Ya anteriormente se supuso que esto pordría ser porque se dispone de mayor tiempo libre para ir a comprar.

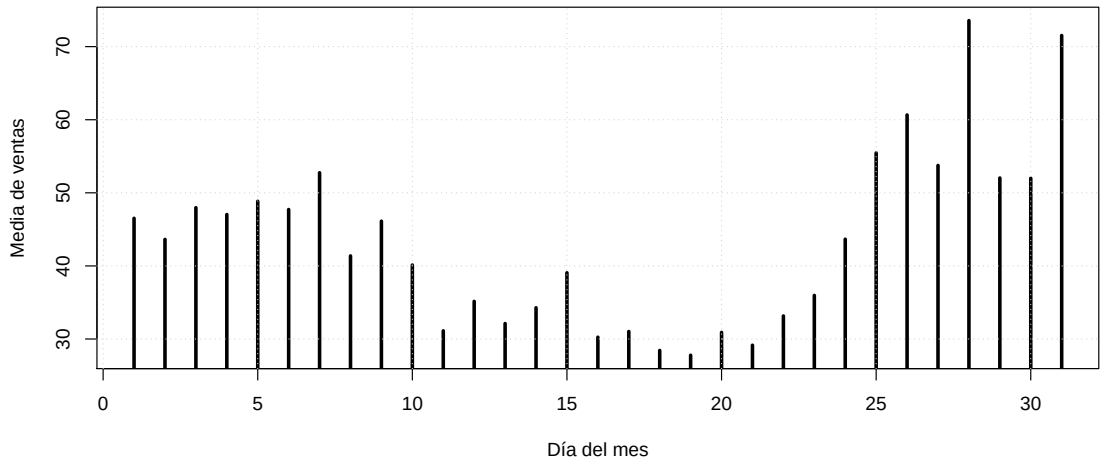


Figura 2.13: MST. Media de ventas respecto al día del mes

Se observa que, tanto a primero como a último de mes es cuando más se vende. Esto puede ser por el cobro de los salarios.

Para el estudio de la periodicidad, se va a realizar un análisis espectral utilizando *spectrum*, en el que se va a respresentar el periodograma sin procesar tanto por frecuencia como período para la variable objetivo, *CantidadDiaria*. Esta función utiliza la Transformada rápida de Fourier.

Periodograma por frecuencia

```
espectral<- spectrum(datos2$CantidadDiaria)
```

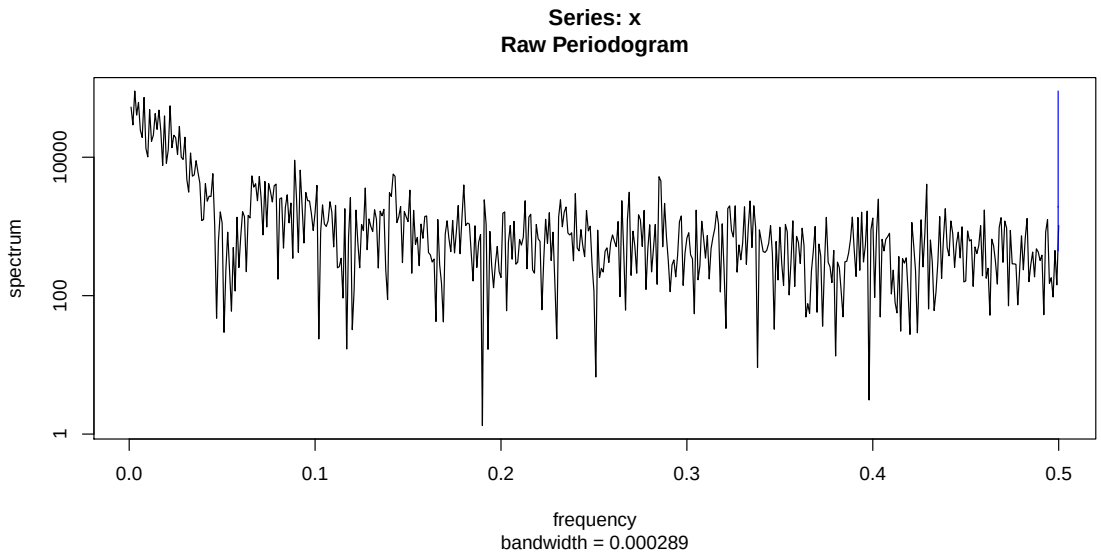


Figura 2.14: MST. Periodograma por frecuencia

La *(Figura 2.13)* representa el periodoframa por frecuencia de la variable analizada, donde se aprecian la existencia de distintos picos, siendo éstos las frecuencias más características de la serie.

Periodograma por periodo

```
periodo<- 1/espectral$freq
```

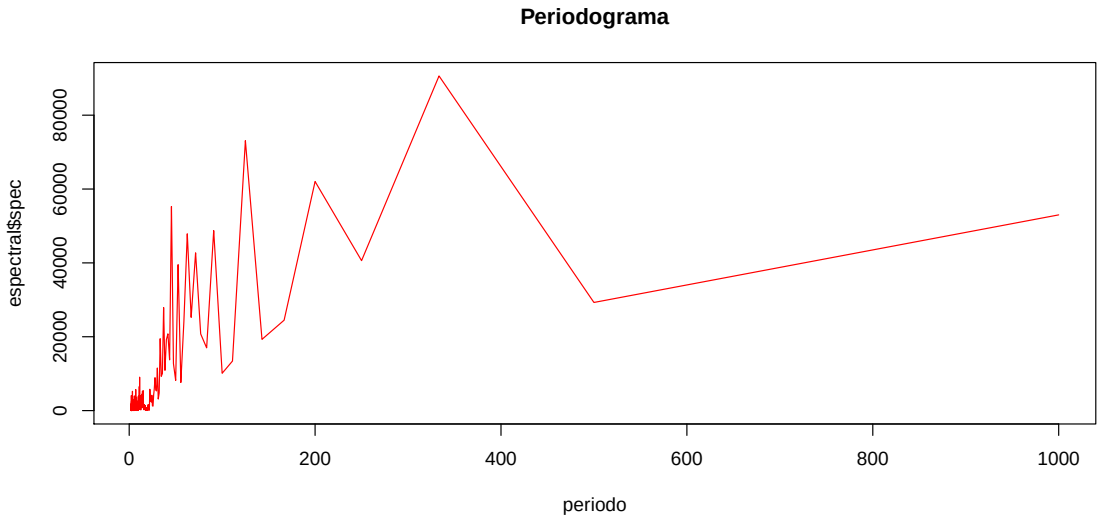


Figura 2.15: MST. Periodograma por periodo

El pico más pronunciado se da en un valor aproximado a 365, teniéndose por tanto la periodicidad anual.

Se puede considerar la periodicidad dentro de cada mes o no. Por ello, se ajustan a los datos dos tipos de modelos.

Periodicidad dentro de cada mes

Se debe calcular dicha periodicidad, por lo que se calcula la media de días que tiene un mes, que será incluida al crear la serie temporal:

```
(31+28.25+31+30+31+30+31+31+30+31+30+31)/12
```

```
## [1] 30.4375
```

Al considerar varias componentes estacionales, se crea la serie temporal mediante *msts*, donde se incluyen los períodos semanales, mensuales y anuales.

```
y1=msts(datos2$CantidadDiaria[indient],
       seasonal.periods=c(7,30.44,365.25), ts.frequency=1)
```

Se realiza la transformación de fourier a los datos, ya que esta proporciona funciones que modelizan y simplifican modelos matemáticos.

```
z <- fourier(y1, K=c(2,3,5))
zf <- fourier(y1, K=c(2,3,5), h=nt)
xregent=data.frame(z,datos2[indient,c(3)])
xregtest=data.frame(zf,datos2[inditest,c(3)])
```

El conjunto de datos presenta valores 0, esto ocasiona problemas a la hora de predecir, para ello se le suma una constante a la variable objetivo, eliminando este problema.

```
arima1=auto.arima(y1+10, seasonal=FALSE)
summary(arima1)
```

```
## Series: y1 + 10
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.3433  -0.6774
## s.e.  0.0948   0.0781
##
## sigma^2 estimated as 1591:  log likelihood=-4807.93
## AIC=9621.85   AICc=9621.88   BIC=9636.4
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.1415941 39.82024 21.64648 -35.45436 62.94914 0.9497875
##                      ACF1
## Training set -0.02247888
```

Se ha ajustado un ARIMA (1,1,1)

Para la representación se debe eliminar la constante anteriormente sumada.

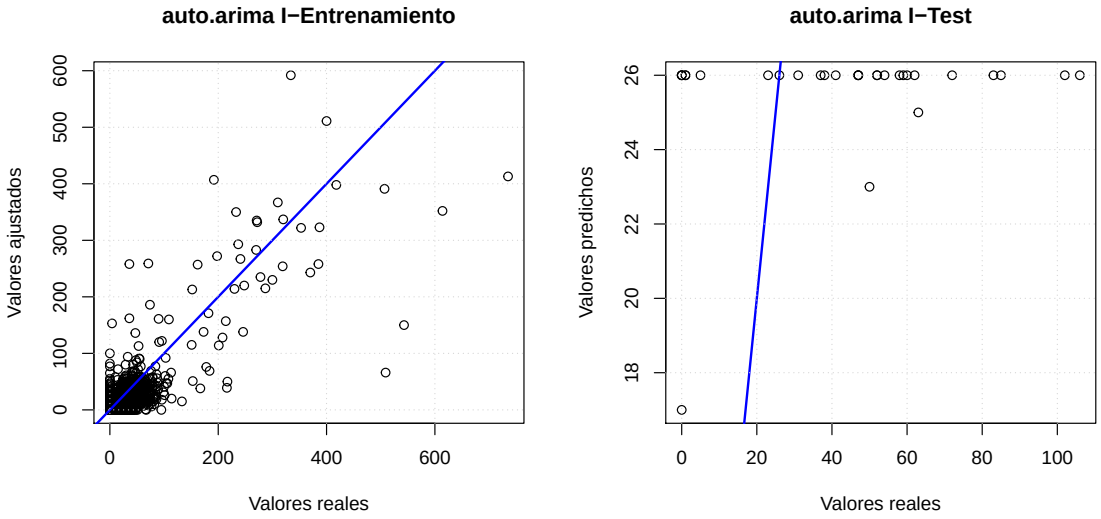


Figura 2.16: Modelo de Series temporales. Modelo con periodicidad mensual.

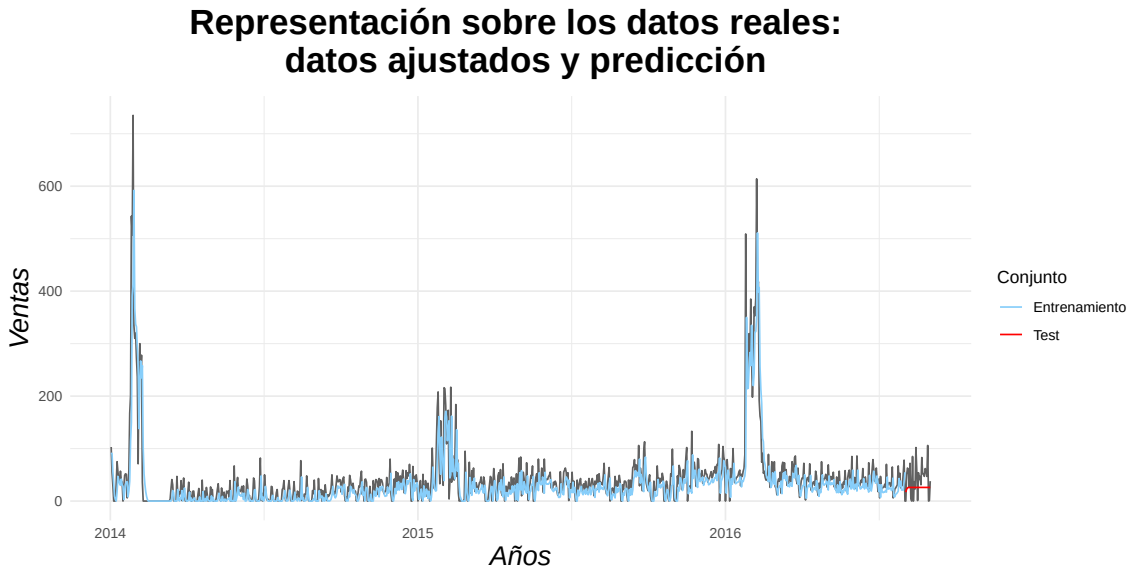


Figura 2.17: Modelo de Series temporales. Modelo con periodicidad mensual.

```
##                            LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.5230179 40.89349 0.6392678
## Conjunto_test          0.2938408 34.52101 0.0538574
```

A pesar de que para el conjunto de entrenamiento parece que este modelo, por ahora, es el que mejor ajusta los datos, teniéndose explicada el 63'93 % de la variabilidad y aceptando la hipótesis de la independencia de los datos, para el conjunto test solo explica un 5 %. Ésto se debe a que se está produciendo un sobreajuste, reflejado de forma muy clara en la última gráfica, que aunque se observa que este modelo sí que recoge exitosamente las ventas de los meses enero/febrero, realiza una mala predicción.

Sin considerar periodicidad dentro de cada mes

Se incluyen los períodos semanales y anuales.

```
y2=msts(datos2$CantidadDiaria[indient],
       seasonal.periods=c(7,365.25), ts.frequency=1)
```

La transformada de fourier para este modelo:

```
z <- fourier(y2, K=c(2,5))
zf <- fourier(y2, K=c(2,5), h=nt)
xregent=data.frame(z,datos2[indient,c(3)])
xregtest=data.frame(zf,datos2[inditest,c(3)])
```

se ajusta mediente *auto.arima*:

```
arima2=auto.arima(y2+10, seasonal=FALSE,xreg=as.matrix(xregent))
summary(arima2)
```

```
## Series: y2 + 10
## Regression with ARIMA(0,1,1) errors
##
## Coefficients:
##           ma1    drift    S1.7     C1.7    S2.7    C2.7   S1.365   C1.365
##       -0.3907  -0.0254  0.7324  -5.1615  4.2652  0.4382  -1.4054  16.1339
## s.e.   0.0349   0.7887  1.6897   1.6895  1.3228  1.3226  63.9593  65.7805
##        S2.365   C2.365   S3.365   C3.365   S4.365   C4.365   S5.365    C5.365
##        4.2733  11.6838  12.9786   6.3455  12.0460  -3.7527  10.7095  -12.5468
## s.e.  32.2598  32.6610  21.6837  21.6525  16.2992  16.2237  12.9295   12.9275
##       datos2.indient..c.3..
##                   -61.8675
## s.e.               12.1493
##
## sigma^2 estimated as 1547:  log likelihood=-4787.3
## AIC=9610.6   AICc=9611.34   BIC=9697.86
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.02376879 38.95864 21.67419 -31.0777 65.91478 0.9510033
##                    ACF1
## Training set 0.02684946
```

En este modelo, se obtiene un ARIMA(0,1,1), no existe la parte aurorregresiva.

De nuevo, se elima la constante sumada y se obtienen las predicciones:

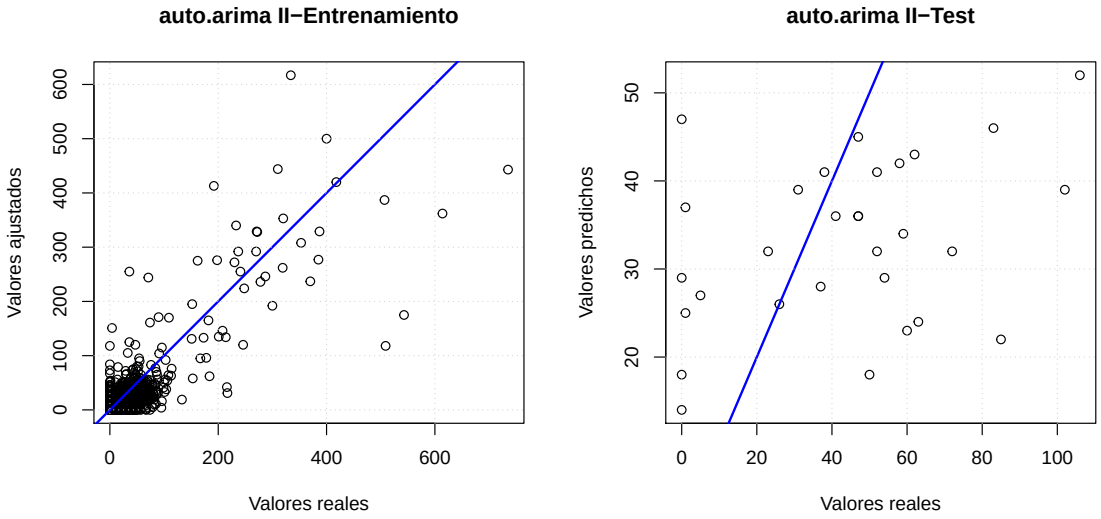


Figura 2.18: Modelo de Series temporales. Modelo sin periodicidad mensual.

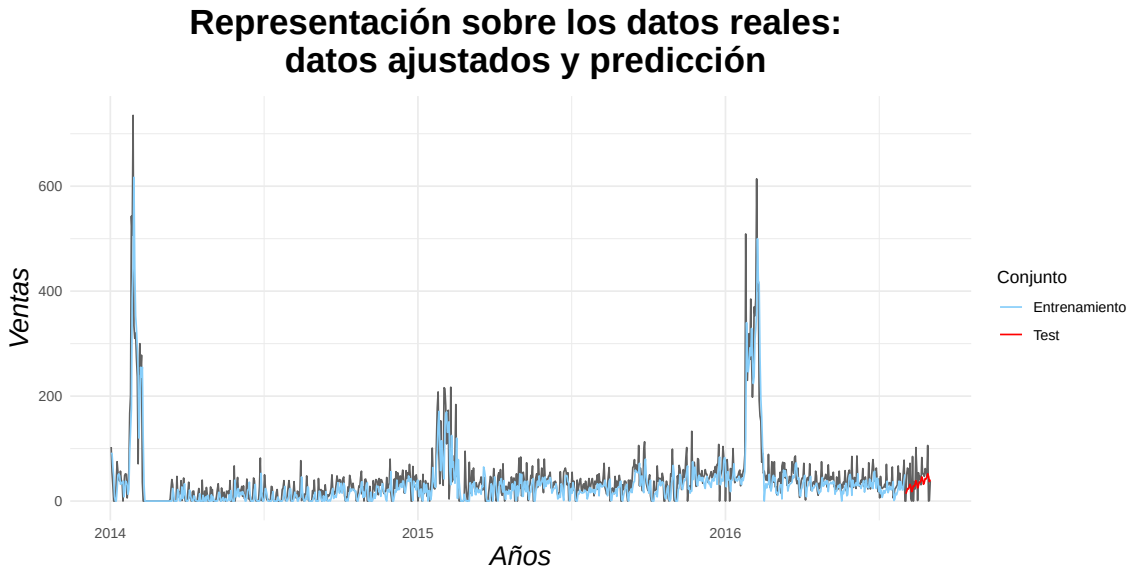


Figura 2.19: Modelo de Series temporales. Modelo sin periodicidad mensual.

```
##                            LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.4856349 39.53581 0.6623559
## Conjunto_test          0.2532062 29.75287 0.1311991
```

Se vuelve a tener un modelo el cual ajusta los datos de entrenamiento demasiado bien y por tanto, ocasiona un sobreajuste.

2.1.6. Modelo ramdonForest

Para la implementación del algoritmo árboles aleatorios se va a utilizar la librería *randomForest* que aplica el algoritmo de Breiman. Para la obtención del modelo, no se va a fijar el número de árboles y se quiere evaluar la importancia de los predictores.

```
library(randomForest)
mRF1=randomForest(CantidadDiaria ~  PrecioMedioDiario + DiaSem + Mes,
                  importance=TRUE, data = datos[indient,])
```

```
mRF1
```

```
##
## Call:
##  randomForest(formula = CantidadDiaria ~ PrecioMedioDiario + DiaSem +      Mes, da
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 2741.096
##                     % Var explained: 35.98
```

```
importance(mRF1)
```

```
##                     %IncMSE IncNodePurity
## PrecioMedioDiario  31.11697      917467.8
## DiaSem            -12.44184      213683.9
## Mes                32.00179      774464.0
```

Se han creado 500 árboles de regresión y se tiene que el número de variables seleccionadas aleatoriamente es 1.
El modelo seleccionado explica el 37 % de la variabilidad de los datos. Además, la variable *DiaSem* es la variable menos importante.

Las predicciones se van a realizar de la misma forma que en los casos anteriores.

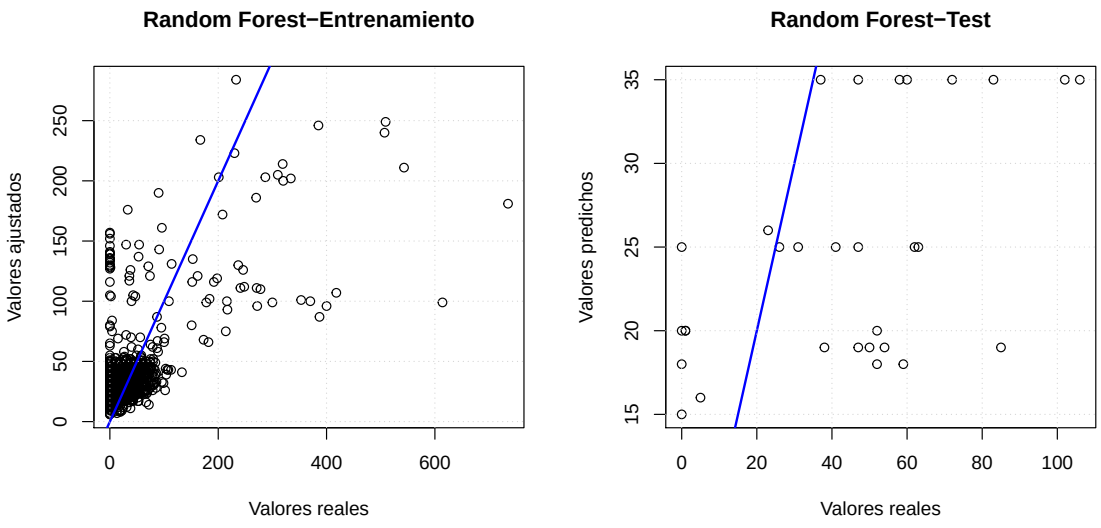


Figura 2.20: Algoritmo de RamdonForest.

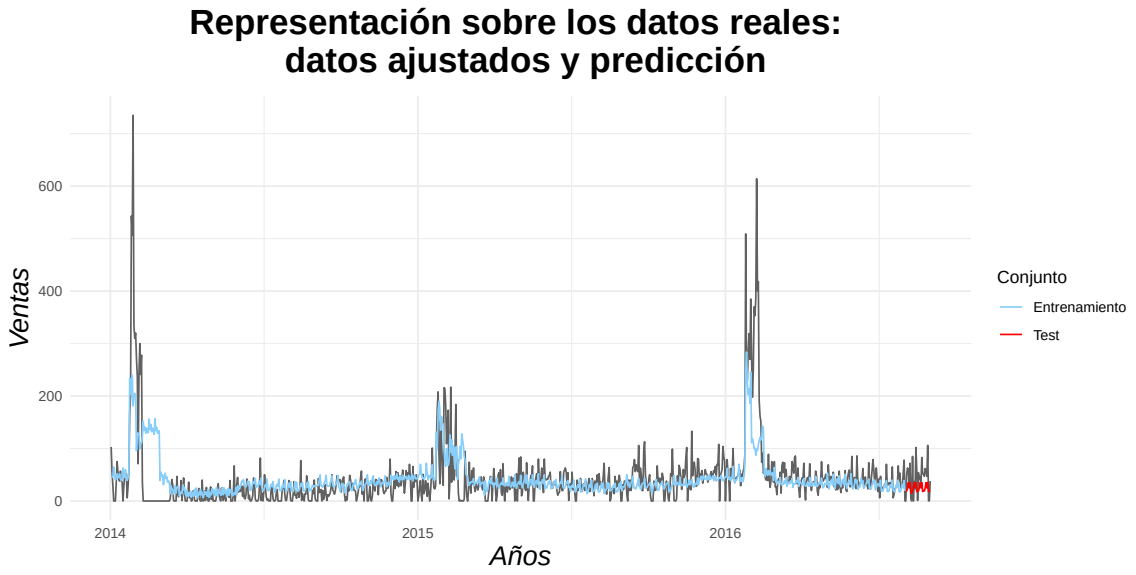


Figura 2.21: Algoritmo de RamdonForest.

```
##                             LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.0000000 52.35513 0.3649620
## Conjunto_test          0.3187755 32.69506 0.2935005
```

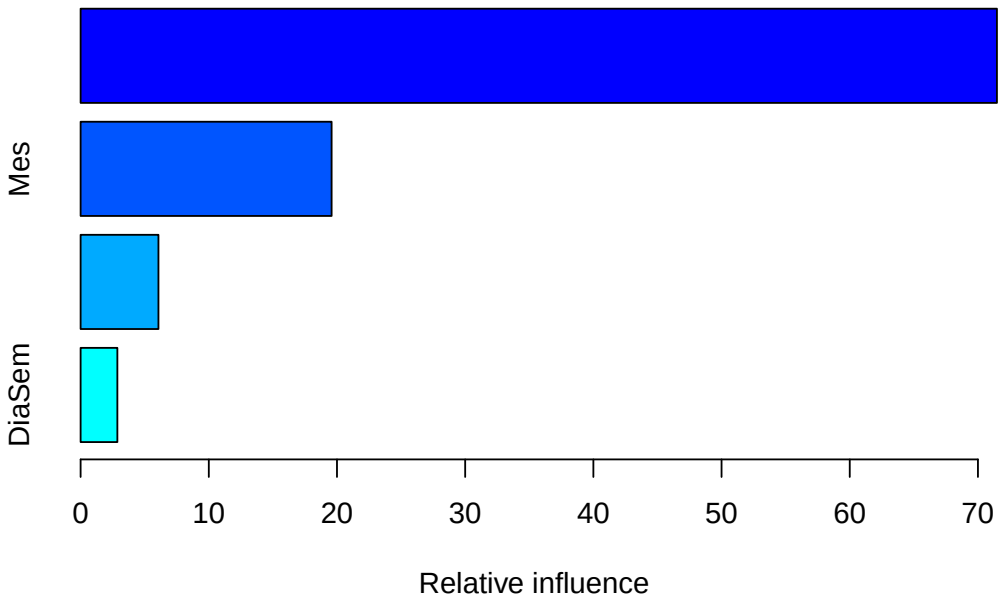
Mediante este algoritmo, se obtiene un peor comportamiento en comparativa con los modelos estudiados anteriormente. Aún así, los valores predichos oscilan sobre el rango de los observados.

2.1.7. Modelo de Gradient Boosting

Otra de las técnicas basadas en árboles es Gradient Boosting. Para su aplicación se utiliza la librería *gbm*. Por defecto, va a ajustar 100 árboles (iteraciones).

```
library(gbm)
modelogbm=gbm(CantidadDiaria ~  PrecioMedioDiario + DiaSem + Mes +
                Carnaval ,
              data = datos[indient,])
```

```
summary(modelogbm)
```




```
##                                  var   rel.inf
## PrecioMedioDiario PrecioMedioDiario 71.481446
## Mes                             Mes 19.579801
## Carnaval                   Carnaval  6.071268
## DiaSem                       DiaSem  2.867485
```

Con clara diferencia, el predictor más influyente es el precio del producto.

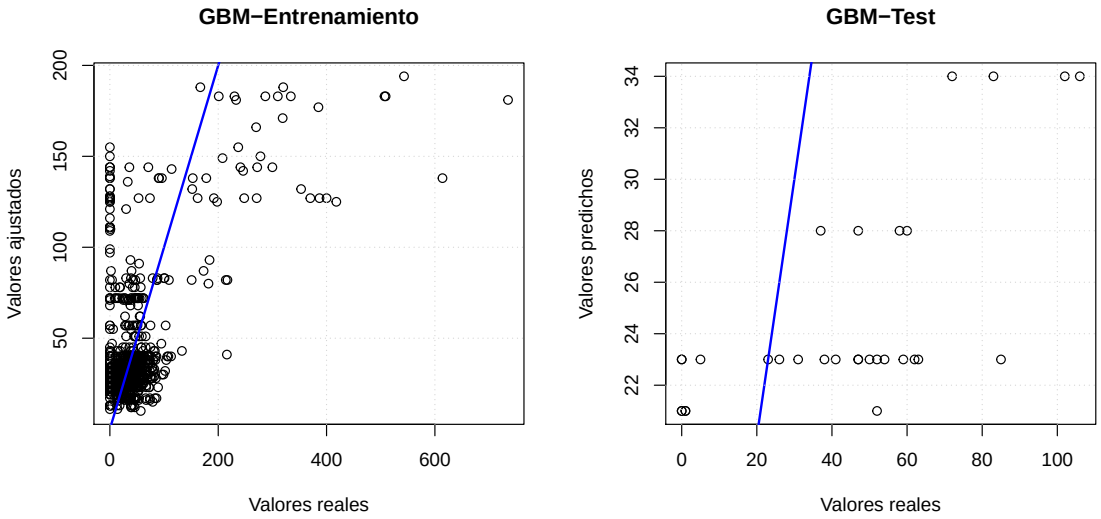


Figura 2.22: Algoritmo de Gradient Boosting.

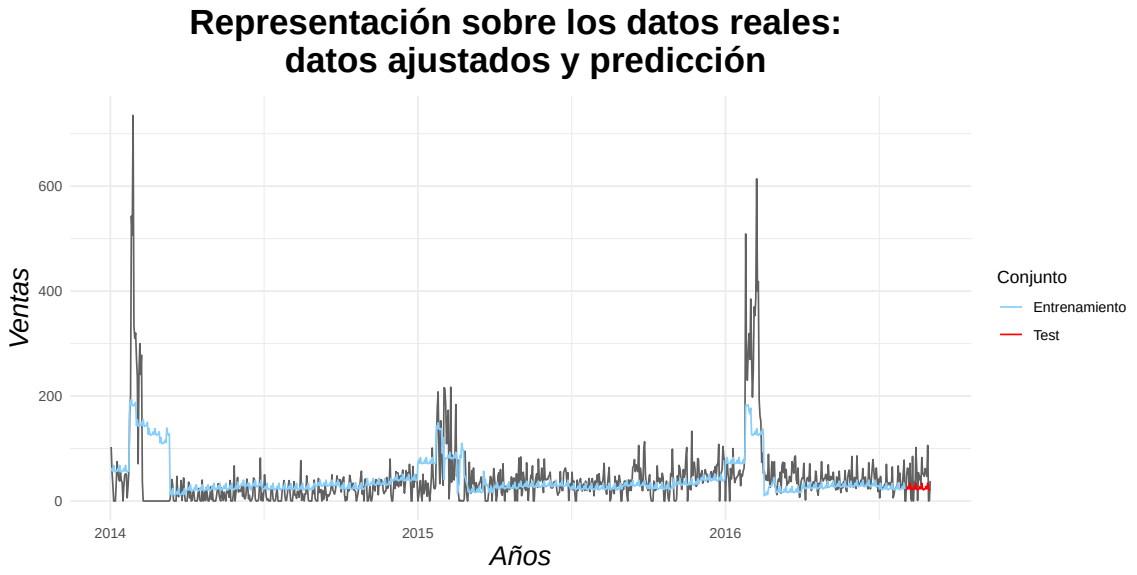


Figura 2.23: Algoritmo de Gradient Boosting.

```
##                              LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.0000000 53.51122 0.3347442
## Conjunto_test          0.3788418 32.96968 0.4787901
```

Este modelo, al igual que ocurrió en el modelo de regresión Binomial negativa, explica mayor porcentaje de variabilidad para el conjunto test, 47.87 %, que para el de entrenamiento, 33.47 % y la cantidad de error ha disminuido. Además, el incremento de ventas dado los primeros meses del años también se ve reflejado, aunque sigue sin mostrar el gran aumento de 2016.

2.1.8. Modelo de RNA

Para poder aplicar el algoritmo de redes neuronales artificiales, el paquete utilizado es *neuralnet*. El cual va a permitir la aplicación, visualización e implementación de redes neuronales.

Las variables predictoras seleccionadas en este algoritmo son *PrecioMedioDiario*, *DiaSem*, *Mes* y *Carnaval*.

```
library(neuralnet)
datos3=datos[,c(2:5,7)]
```

Antes de crear la red neuronal, se comienza tanto por la normalización como por la codificación de etiquetas de las clases nominales de los predictores. Para la normalización se va a utilizar la técnica de *normalización máximo-mínimo*, que consiste en transformar los valores de las variables al intervalo (0,1) mediante el método de escalación *min-max*. Mediante la codificación *one-hot* se crean variables dummy, es decir, se crean tantas columnas como categorías tengan las variables categóricas, de forma que se obtiene un 1 si ese datos presenta esa característica, o un 0 en caso contrario. Esto se realiza mediante la siguiente función:

```
# PREPARADO DE DATOS
```

```
library(fastDummies)
normalize<-function(x){
  if (is.numeric (x)) {
  return ( (x-min(x))/(max(x)-min(x)))}
  else dummy_cols(x)
}

datos3_norm=as.data.frame(lapply(datos3,FUN=normalize))

# Se eliminan las columnas categóricas
datos3_norm=datos3_norm[-c(3,11,24)]

train_nrm=datos3_norm[indient, ] # conjunto de entrenamiento
test_nrm=datos3_norm[inditest, ] #conjunto test
```

A continuación, se aplica el algoritmo de red neuronal a los datos de entrenamiento con una capa oculta y 5 neuronas en ella. La capa de entrada tendrán tantas neuronas como predictores, en este caso 21 y la de la salida 1, al tener una única variable dependiente. Además, se especifica que no se aplique una función diferenciable, para suavizar el resultado del producto cruzado de la covariable o neuronas y los pesos; y el umbral para los derivados parciales de la función de error como criterio de parada igual a 0.01.

```
names(train_nrm)= c("CantidadDiaria","PrecioMedioDiario",
                    "Lunes","Martes","Miércoles","Jueves",
                   "Viernes","Sábado","Domingo",
                   "Enero","Febrero","Marzo","Abril","Mayo",
                   "Junio","Julio","Agosto","Septiembre",
                   "Octubre","Noviembre","Diciembre",
                   "NoCarnaval", "SiCarnaval")

nms=names(train_nrm)
frml <- as.formula(paste("CantidadDiaria ~",
                         paste(nms[!nms %in% "CantidadDiaria"],
                         collapse = " + ")))
attach(datos3_norm)

# Modelo
modelorna=neuralnet(frml, data = train_nrm, hidden = c(4),
                    threshold=0.01, algorithm="rprop+" )

head(modelorna$result.matrix)
```

```
##                                    [,1]
## error                       2.014016e+00
## reached.threshold           9.347711e-03
## steps                       1.277000e+03
## Intercept.to.1layhid1      -1.189404e+00
## PrecioMedioDiario.to.1layhid1 -4.731633e+00
## Lunes.to.1layhid1           3.561047e-01
```

```
# Una lista con los 6 primeros pesos
head(unlist(modelorna$weights))
```

```
## [1] -1.18940370 -4.73163309  0.35610474  0.03053803 -0.84633887 -1.08480170
```

Gráficamente, la red ajustada:

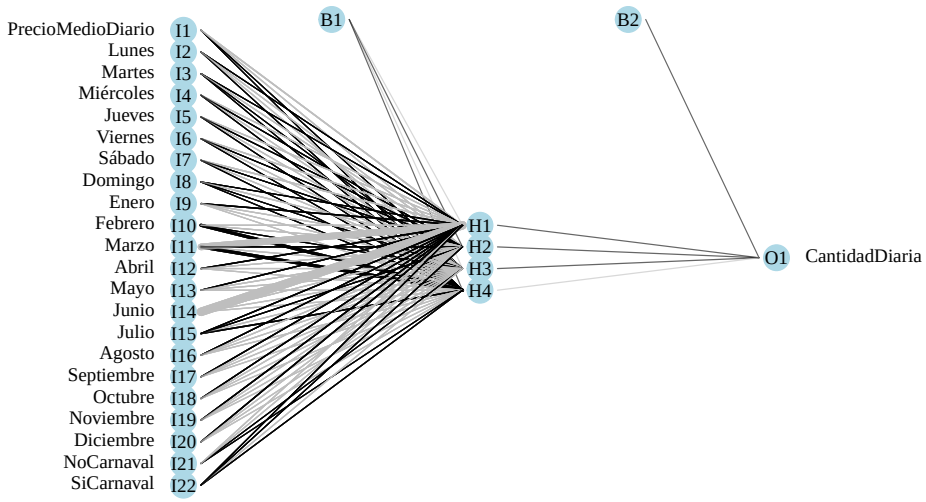


Figura 2.24: Modelo de red neuronal artifical

Ahora bien, se ajusta el modelo tanto al conjunto de entrenamiento como al de prueba para, posteriormente, realizar sus resentaciones gráficas y calcular errores.

```
modeloresult_train=compute(modelorna,train_nrm)
predic_train=modeloresult_train$net.result

modeloresult_test=compute(modelorna,test_nrm[-1])
predic_test=modeloresult_test$net.result
```

Para la predicción es necesario la desnormalización de los datos, es decir, convertirlos a su formato original . Esto se realiza mediante la siguiente función:

```
str_max=max(datos3$CantidadDiaria)
str_min=min(datos3$CantidadDiaria)

unnormalize <- function(x, min, max) {
  return( (max - min)*x + min )
}

ActualCantidad_train=unnormalize(predic_train,str_min,str_max)
ActualCantidad_test=unnormalize(predic_test,str_min,str_max)
```

Se obtienen las predicciones:

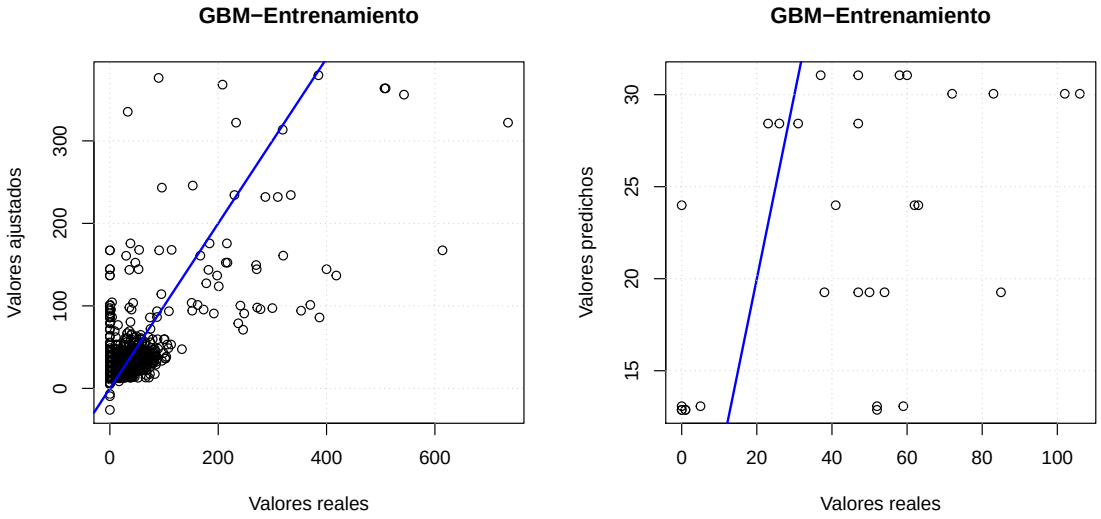


Figura 2.25: Algoritmo de Red Neuronal Artificial.

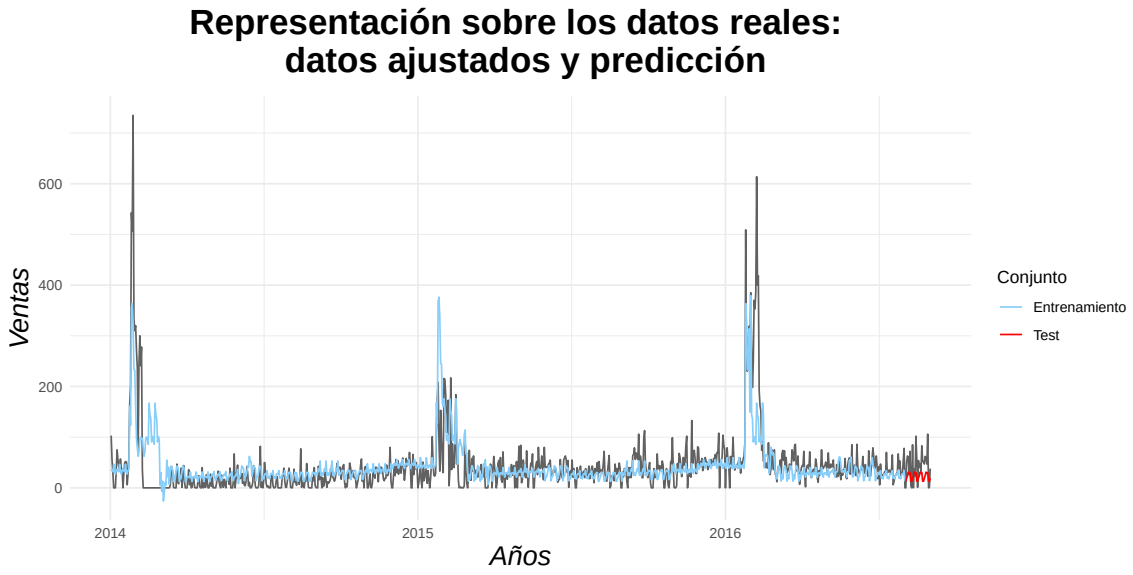


Figura 2.26: Algoritmo de Red Neuronal Artificial.

```
##                              LBtest     RMSE        R2
## Conjunto_Entrenamiento 0.0000000 48.03723 0.4610816
## Conjunto_test          0.2617346 34.24371 0.2656996
```

Sin que exista un sobreajuste, este algoritmo es el que mejor se adapta al conjunto de entrenamiento, reflejando el crecimiento de las ventas a principio de año. Además, el conjunto test también presenta buenas predicciones.

Conclusiones

Esta última sección del trabajo está dedicada a estimar en qué medida se han alcanzado las metas y los objetivos propuestos.

En primer lugar, la modelización y predicción de ventas forman uno de los pilares básicos para el triunfo de cualquier organización dedicada al comercio. Para ello, es crucial conocer y comprender las distintas fases del modelado de datos y saber interpretar sus resultados correctamente, por lo que la Analítica empresarial es, para cualquier empresa, un apoyo esencial.

En segundo lugar, cuando se trata de modelar una situación real, se presentan dintintas limitaciones y complicaciones ya que este proceso depende de muchos factores como por ejemplo, la calidad de los datos. También puede perderse información, como por ejemplo al no haber considerado los predictores *Temperatura*, al presentar demasiados valores perdidos.

Por último, en el presente trabajo, todos los modelos y algoritmos utilizados han obtenido buenas predicciones, en mayor o menor medida. Como resumen final, se van a presentar las cantidades de ventas correpondientes a la primera semana de los datos del conjunto test, desde 11/08/2016 hasta 18/08/2016.

Tabla 2.1: Cantidad diaria de ventas

Dias	ValorObservado
Miércoles	0
Jueves	37
Viernes	23
Sábado	102
Domingo	59
Lunes	1
Martes	54

Tabla 2.2: Cantidad predicha según los modelos aplicados

Dias	Poisson_I	Poisson	BN	ST_mes	ST	GB	RF	RNA
Miércoles	17	20	18	17	14	21	20	36
Jueves	17	19	19	23	18	23	19	34
Viernes	22	22	22	25	24	23	25	32

Dias	Poisson_I	Poisson	BN	ST_mes	ST	GB	RF	RNA
Sábado	36	23	25	26	23	28	35	47
Domingo	24	20	22	26	26	23	25	32
Lunes	27	30	29	26	32	34	35	27
Martes	14	21	18	26	27	23	16	35

De acuerdo a los valores pronosticados, en general, el modelo que mayor cantidad de ventas estima es el de Red neuronal, el cual predice que para esa semana, la cantidad de árticulos vendidos por día estarían entre 13 y 51.

Ahora se estima la cantidad media según todos los modelos, teniéndose:

MediaObservada
39

	MediaStock
Poisson_I	22
Poisson	22
BN	22
ST_mes	24
ST	23
GB	25
RF	25
RNA	35

Por un lado, se observa que todas las técnicas estadísticas estiman para dicha semana, aproximadamente, la misma cantidad de productos vendidos, entre 22 y 24. Por otro lado, los algoritmos de aprendizaje supervizado prevén una mayor cantidad de ventas, entre 25 y 31, siendo esta última, la cantidad estimada por RNA, como ya se ha comentado.
Por tanto, comparando con la media de artículos vendidos en esa semana (39), se aconseja el modelo RNA.

Por último, se van a predecir los valores para la semana de la que no se obtienen datos, que corresponde a la primera semana de septiembre de 2016, es decir, desde 01/09/2016 (jueves) al 07/09/2016 (miércoles). Dicha predicción se realiza mediante todas las técnicas utilizadas anteriormente.

Tabla 2.5: Cantidad predicha según el precio medio

Dia	Poisson_I	Poisson	BN	ST_mes	ST	GB	RF	RNA
Jueves	21	27	29	17	6	14	28	21
Viernes	25	29	32	23	10	19	31	31
Sábado	36	25	28	25	16	14	30	27
Domingo	43	36	37	26	15	25	34	45
Lunes	21	26	23	26	18	14	23	14
Martes	26	25	23	26	24	12	25	22
Miércoles	20	24	25	26	19	14	26	16

Tabla 2.6: Cantidad predicha según el precio máximo

Dia	Poisson_I	Poisson	BN	ST_mes	ST	GB	RF	RNA
Jueves	22	29	30	17	9	14	28	21
Viernes	27	31	34	23	13	19	31	31
Sábado	38	27	29	25	19	14	30	27
Domingo	46	39	40	26	18	25	33	45
Lunes	23	27	25	26	21	14	23	14
Martes	28	26	24	26	28	12	24	22
Miércoles	21	25	26	26	22	14	26	16

Tabla 2.7: Cantidad predicha según el precio mínimo

Dia	Poisson_I	Poisson	BN	ST_mes	ST	GB	RF	RNA
Jueves	152	199	166	17	99	43	33	373
Viernes	183	210	185	23	103	48	29	266
Sábado	260	185	160	25	110	43	26	191
Domingo	314	265	217	26	108	54	37	482
Lunes	155	189	136	26	111	43	24	495
Martes	193	181	132	26	118	41	20	460
Miércoles	146	174	143	26	113	43	19	372

Tabla 2.8: Cantidad predicha según la mediana de los precios

Dia	Poisson_I	Poisson	BN	ST_mes	ST	GB	RF	RNA
Jueves	25	32	33	17	14	30	29	29
Viernes	30	34	37	23	18	35	33	38
Sábado	42	30	32	25	24	30	38	33
Domingo	51	43	44	26	23	40	48	53
Lunes	25	31	27	26	26	30	27	25
Martes	31	29	26	26	32	28	26	30
Miércoles	24	28	29	26	27	30	26	24

Se observa como el precio del producto influye en gran medida en la venta de éste. Por ejemplo, la cantidad vendida cuando el precio es mínimo es casi seis veces mayor a cuando el precio es máximo. En general, todos predicen cantidades similares de ventas, excepto, como ya se ha comentado, cuando el precio es mínimo, que las ventas se incrementan de manera razonable.

Como posibles líneas futuras, están el estudio de las ventas cruzadas de productos similares, el estudio de impacto de promociones que afecta a conjuntos de productos y la influencia de los factores meteorológicos.

Agradecimientos

Finalmente, quiero agradecer a todos los profesores que me han acompañado durante mi formación en estos años de aprendizaje y muy especialmente a José Luis Pino, mi tutor en esta última etapa, por su dedicación, atención y esfuerzo.

También a mis padres, por creer en mí y apoyarme en todo momento. He aprendido este último año que mezclar Trabajo de Fin de Grado y pandemia no es fácil.

Y por supuesto, a mis compañeras y amigas, esas que he conocido gracias a la estadística. Sin duda, es lo mejor que me llevo de estos cuatro maravillosos años, mi equipo.

Apéndice A

Apéndice: Exploración y manipulación de datos

A.1. Lectura y resumen de datos

```
load("9004Item15.RData")
summary(df_item_agr_ampliado)
```

```
##      Fecha             CantidadDiaria  PrecioMedioDiario    Lugar
##  Min.   :2014-01-02   Min.   :  0.0   Min.   :0.490     Length:973
##  1st Qu.:2014-09-02   1st Qu.: 14.0   1st Qu.:1.870     Class :character
##  Median :2015-05-03   Median : 31.0   Median :1.870     Mode  :character
##  Mean   :2015-05-03   Mean   : 42.9   Mean   :1.774
##  3rd Qu.:2016-01-01   3rd Qu.: 48.0   3rd Qu.:1.870
##  Max.   :2016-08-31   Max.   :735.0   Max.   :1.950
##
##   Provincia              TMax            TMin            TMed
##  Length:973         Min.   :17.40   Min.   :12.30   Min.   :15.20
##  Class :character   1st Qu.:20.70   1st Qu.:16.90   1st Qu.:18.80
##  Mode  :character   Median :22.80   Median :18.70   Median :20.70
##                     Mean   :23.05   Mean   :19.07   Mean   :21.07
##                     3rd Qu.:25.00   3rd Qu.:21.30   3rd Qu.:23.10
##                     Max.   :31.80   Max.   :25.20   Max.   :28.20
##                     NA's   :209     NA's   :209     NA's   :209
##      Racha             VMax           Pre24            Pre00_06
##  Min.   :12.00   Min.   : 6.00   Min.   : 0.0000   Min.   :0.00000
##  1st Qu.:24.00   1st Qu.:12.00   1st Qu.: 0.0000   1st Qu.:0.00000
##  Median :29.00   Median :14.00   Median : 0.0000   Median :0.00000
##  Mean   :29.82   Mean   :14.55   Mean   : 0.4471   Mean   :0.07313
##  3rd Qu.:34.00   3rd Qu.:17.00   3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   :73.00   Max.   :29.00   Max.   :39.9000   Max.   :9.50000
##  NA's   :212     NA's   :212     NA's   :226       NA's   :184
##     Pre06_12          Pre12_18         Pre18_24          diasem
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.0000   Min.   :1
##  1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.:2
```

```
##  Median : 0.0000   Median : 0.000   Median : 0.0000   Median :4
##  Mean   : 0.1489   Mean   : 0.116   Mean   : 0.1099   Mean   :4
##  3rd Qu.: 0.0000   3rd Qu.: 0.000   3rd Qu.: 0.0000   3rd Qu.:6
##  Max.   :36.9000   Max.   :11.800   Max.   :30.4000   Max.   :7
##  NA's   :172       NA's   :171      NA's   :183
##       mes
##  Min.   : 1.000
##  1st Qu.: 3.000
##  Median : 6.000
##  Mean   : 6.028
##  3rd Qu.: 9.000
##  Max.   :12.000
##
```

```
table(df_item_agr_ampliado$Lugar)
```

```
##
## Las Palmas de Gran Canaria, Pl. de la Feria
##                                         839
```

```
table(df_item_agr_ampliado$Provincia)
```

```
##
## Las Palmas
##        839
```

A.2. Manipulación

```
# Reenombrar columnas
colnames(datos)=c("Fecha","CantidadDiaria","PrecioMedioDiario",
                  "DiaSem","Mes","DiaLaborable","Carnaval")

# Reenombrar y reordenar DiaSem
datos$DiaSem=factor(datos$DiaSem)
levels(datos$DiaSem)=c("Domingo","Lunes","Martes","Miércoles",
                       "Jueves","Viernes","Sábado")
levels(datos$DiaSem)
datos$DiaSem=factor( datos$DiaSem,
          levels = levels( datos$DiaSem )[ c( 2,3,4,5,6,7,1 ) ] )
levels( datos$DiaSem)

# Reenombrar y reordenar Mes
datos$Mes=factor(datos$Mes)
levels(datos$Mes)=c("Enero","Febrero","Marzo","Abril",
                    "Mayo","Junio","Julio","Agosto",
                    "Septiembre", "Octubre","Noviembre","Diciembre")


# Reenombrar y reordenar DiaLaborable
```

```
datos$DiaLaborable=factor(datos$DiaLaborable)
levels(datos$DiaLaborable)
levels(datos$DiaLaborable)=c("NoLaborable","SiLaborable")

# # Reenombrar y reordenar Carnaval
datos$Carnaval=factor(datos$Carnaval)
levels(datos$Carnaval)
levels(datos$Carnaval)=c("NoCarnaval","SiCarnaval")
```

A.3. Gráficos

```
ggplot(data=datos, aes(CantidadDiaria, DiaSem,fill=DiaSem)) +
geom_boxplot()+
labs(x= "Ventas",y = "Días de la semana") +
theme(axis.title = element_text(face="italic", colour="black",
                                size=rel(1.5))) +
theme_minimal()
```

```
 ggplot(datos, aes(CantidadDiaria, Mes, fill=Mes)) +
 geom_boxplot() +
labs(x= "Ventas",y = "Meses") +
theme(axis.title = element_text(face="italic", colour="black",
                                size=rel(1.5))) +
theme_minimal()
```

```
 ggplot(datos, aes(CantidadDiaria, Carnaval, fill=Carnaval)) +
 geom_boxplot() +
labs(x= "Ventas",y = "Carnaval") +
theme(axis.title = element_text(face="italic", colour="black",
                                size=rel(1.5))) +
theme_minimal()
```

```
plot(by(datos2$CantidadDiaria,datos2$Mes,mean),type="h",lwd=3,
    xlab="Mes",ylab="Media de ventas"); grid()
```

```
plot(by(datos2$CantidadDiaria,datos2$DiaSem,mean),type="h",lwd=3,
     xlab="Día de la semana",ylab="Media de ventas"); grid()
```

```
plot(by(datos2$CantidadDiaria,dfecha,mean),type="h",lwd=3,
     xlab="Día de la semana",ylab="Media de ventas"); grid()
```

Apéndice B

Apéndice: Análisis y predicción

B.1. Análisis espectral

```
espectral<- spectrum(datos2$CantidadDiaria)
```

```
plot(periodo,espectral$spec,main="Periodograma",type="l",col="red")
```

B.2. Red neuronal

```
library(NeuralNetTools)
par(mar = numeric(4), family = 'serif')
plotnet(modelorna, alpha = 0.6,cex_val=1, circle_cex=3, pad_x=0.7)
```

B.3. Función para predicción

```
library(ggplot2)
library(gridExtra)

Ajuste<- function(y1,pred1,y2,pred2,titulo1,titulo2)
{
  residuos1=y1-pred1
  residuos2=y2-pred2
  Predic=c(pred1,pred2)
  Fecha=datos$Fecha
  Conjunto=c(rep("Entrenamiento",943),rep("Test",30))
  datospred=data.frame(Fecha,datos["CantidadDiaria"],Predic,Conjunto)
  par(mfrow=c(1,2))
  plot1=plot(y1,pred1,main=titulo1, xlab="Valores reales",
             ylab="Valores ajustados")
  abline(a=0,b=1,col="blue",lwd=2)
  grid()
  plot2=plot(y2,pred2,main=titulo2,xlab="Valores reales",
```

```
            ylab="Valores predichos")
  abline(a=0,b=1,col="blue",lwd=2)
  grid()

  plot3=ggplot(data=datospred, aes(Fecha, CantidadDiaria)) +
    geom_line(color="gray38") +
    theme_minimal() +
    labs(x= "Años",y = "Ventas") +
    theme(axis.title = element_text(face="italic", colour="black",
                                    size=rel(1.5))) +
    geom_line(data = datospred, mapping = aes(Fecha,
                                              Predic,
                                              color= Conjunto)) +
    scale_color_manual(values=c("lightskyblue", "red")) +
    ggtitle ("Representación sobre los datos reales:
             \n datos ajustados y predicción") +
    theme (plot.title = element_text(size=rel(2),
                                     vjust=2,
                                     face="bold",
                                     color="black",
                                     hjust = 0.5))



  LBtest_ent=Box.test(residuos1,type="Ljung")$p.value
  RMSE_ent<- sqrt(mean(residuos1^2))
  R2_ent<- cor(y1,pred1)^2
  LBtest_test=Box.test(residuos2,type="Ljung")$p.value
  RMSE_test<- sqrt(mean(residuos2^2))
  R2_test<- cor(y2,pred2)^2
  graf=grid.arrange(plot3)
  LBtest=c(LBtest_ent,LBtest_test)
  RMSE=c(RMSE_ent,RMSE_test)
  R2=c(R2_ent,R2_test)
  resul=data.frame(LBtest,RMSE,R2)
  rownames(resul)=c("Conjunto_Entrenamiento","Conjunto_test")
  return(resul)

}
pred_E=function(x)
{
  pmax(0,floor(x))

}
```

B.4. Predicción

```
predient=pred_E(predict(modeloP1,type="response")) # Predicciones
preditest=pred_E(predict(modeloP1,datos[inditest,],type="response"))
Ajuste(datos[indient,"CantidadDiaria"],predient,
       datos[inditest,"CantidadDiaria"],preditest,
       "Reg.Poisson-Entrenamiento","Reg.Poisson-Test")
```

```
predient=pred_E(predict(modeloP2,type="response"))
preditest=pred_E(predict(modeloP2,datos[inditest,],type="response"))
Ajuste(datos[indient,"CantidadDiaria"],predient,
       datos[inditest,"CantidadDiaria"],preditest,
       "Reg.Poisson-Entrenamiento","Reg.Poisson-Test")
```

```
predient=pred_E(predict(modeloBN2,type="response"))
preditest=pred_E(predict(modeloBN2,datos[inditest,],type="response"))
Ajuste(datos[indient,"CantidadDiaria"],predient,
       datos[inditest,"CantidadDiaria"],preditest,
       "Reg.BN-Entrenamiento","Reg.BN-Test")
```

```
predient=pred_E(y1-arima1$residuals-10)
predic1=forecast(arima1,  h=nt,level=95)
preditest=c(pred_E(predic1$mean)-10)

Ajuste(datos2$CantidadDiaria[indient],predient,
       datos2$CantidadDiaria[inditest],preditest,
       "auto.arima I-Entrenamiento","auto.arima I-Test")
```

```
predient=pred_E(y2-arima2$residuals-10)
predic2=forecast(arima2, xreg=as.matrix(xregtest), h=nt,level=95)
preditest=c(pred_E(predic2$mean)-10)
Ajuste(datos2$CantidadDiaria[indient],predient,
       datos2$CantidadDiaria[inditest],preditest,
      "auto.arima II-Entrenamiento","auto.arima II-Test")
```

```
predient=pred_E(predict(mRF1))
preditest=pred_E(predict(mRF1,datos[inditest,]))
Ajuste(datos[indient,"CantidadDiaria"],predient,
       datos[inditest,"CantidadDiaria"],preditest,
       "Random Forest-Entrenamiento","Random Forest-Test")
```

```
predient=pred_E(predict(modelogbm,n.trees=100))
preditest=pred_E(predict(modelogbm,datos[inditest,],n.trees=100))
Ajuste(datos[indient,"CantidadDiaria"],predient,
       datos[inditest,"CantidadDiaria"],preditest,
       "GBM-Entrenamiento","GBM-Test")
```

```
Ajuste(datos3[indient,"CantidadDiaria"],ActualCantidad_train,
       datos3[inditest,"CantidadDiaria"],ActualCantidad_test,
       "GBM-Entrenamiento","GBM-Entrenamiento")
```

```
library(knitr)
Poisson_I=(pred_E(predict(modeloP1,datos[inditest,],
                                        type="response"))[1:7])
Poisson=(pred_E(predict(modeloP2,datos[inditest,],
                                          type="response"))[1:7])
BN=(pred_E(predict(modeloBN2,datos[inditest,],
                      type="response"))[1:7])
ST_periodomes=(c(pred_E(predic1$mean)-10)[1:7])
ST=(c(pred_E(predic2$mean)-10)[1:7])
RF=pred_E(predict(mRF1,datos[inditest,]))[1:7]
GB=pred_E(predict(modelogbm,datos[inditest,],n.trees=100))[1:7]
RNA=round(ActualCantidad_train[1:7])
Dias=c("Miércoles","Jueves","Viernes","Sábado","Domingo","Lunes","Martes")
ValorObservado=data.frame(Dias,datos[953:959,"CantidadDiaria"])
colnames(ValorObservado)=c("Dias","ValorObservado")
predtest=data.frame(Dias,Poisson_I,Poisson,BN,
                    ST_periodomes,ST,
                    GB,RF,RNA)
kable(
  ValorObservado, caption = "Cantidad de stock según los valores
  observados")
```

```
kable(
  ValorObservado, caption = "Cantidad de stock según los valores
  observados" )

kable(
  predtest, caption = "Cantidad de stock según los modelos aplicados"
)
```

```
MediaObservada=data.frame(round(mean(datos[953:959,"CantidadDiaria"]),0))
colnames(MediaObservada)="MediaObservada"
medmod=data.frame(colMeans(predtest[,2:9]))
medmod=round(medmod,0)
colnames(medmod)="MediaStock"
kable(MediaObservada)
kable(medmod)
```

B.4.1. Nuevos datos

```
PrecioMedioDiario_mean=round(rep(mean(datos[,"PrecioMedioDiario"]),7))
PrecioMedioDiario_max=rep(max(datos[,"PrecioMedioDiario"]),7)
PrecioMedioDiario_min=rep(min(datos[,"PrecioMedioDiario"]),7)
PrecioMedioDiario_median=rep(median(datos[,"PrecioMedioDiario"]),7)
Fecha=as.Date(c("2016-09-01","2016-09-02","2016-09-03","2016-09-04",
                "2016-09-05","2016-09-06","2016-09-07"))
CantidadDiaria=rep(NA,7)
```

```
DiaSem=factor(c("Jueves","Viernes","Sábado","Domingo","Lunes",
                "Martes","Miércoles"))
DiaSem=factor(DiaSem,
          levels = levels(DiaSem )[c(3,4,5,2,7,6,1)])
Mes=factor(rep("Septiembre",7))
DiaLaborable=factor(rep("SiLaborable",7))

Carnaval=factor(rep("NoCarnaval",7))

datospred1=data.frame(Fecha=Fecha,CantidadDiaria=CantidadDiaria,
                      PrecioMedioDiario=PrecioMedioDiario_mean,
                      DiaSem=DiaSem,Mes=Mes,DiaLaborable=DiaLaborable,
                      Carnaval=Carnaval)
datospred2=data.frame(Fecha=Fecha,CantidadDiaria=CantidadDiaria,
                      PrecioMedioDiario=PrecioMedioDiario_max,
                      DiaSem=DiaSem,Mes=Mes,DiaLaborable=DiaLaborable,
                      Carnaval=Carnaval)
datospred3=data.frame(Fecha=Fecha,CantidadDiaria=CantidadDiaria,
                      PrecioMedioDiario=PrecioMedioDiario_min,
                      DiaSem=DiaSem,Mes=Mes,DiaLaborable=DiaLaborable,
                      Carnaval=Carnaval)
datospred4=data.frame(Fecha=Fecha,CantidadDiaria=CantidadDiaria,
                      PrecioMedioDiario=PrecioMedioDiario_median,
                      DiaSem=DiaSem,Mes=Mes,DiaLaborable=DiaLaborable,
                      Carnaval=Carnaval)
```

```
newpred <- function(datos,datospred,titulo){

datos=rbind(datos, datospred)

Poisson_I=pred_E(predict(modeloP1,datos[c(974:980),],
                                      type="response"))

Poisson=pred_E(predict(modeloP2,datos[c(974:980),],
                                        type="response"))
BN=pred_E(predict(modeloBN2,datos[c(974:980),],
                    type="response"))
datos2=datos[,c(1:5)]
datos2$DiaSem=as.numeric(datos2$DiaSem)
datos2$Mes=as.numeric(datos2$Mes)
nn=7

zf1= fourier(y1, K=c(2,3,5), h=nn)
xregnew1=data.frame(zf1,datos2[c(974:980),c(3)])
predic1=forecast(arima1,  h=nn,level=95)
ST_mes=c(pred_E(predic1$mean)-10)

zf2=fourier(y2, K=c(2,5), h=nn)
```

```
xregnew2=data.frame(zf2,datos2[c(974:980),c(3)])
predic3=forecast(arima2, xreg=as.matrix(xregnew2), h=nn,level=95)
ST=c(pred_E(predic3$mean)-10)


RF=pred_E(predict(mRF1,datos[c(974:980),]))
GB=pred_E(predict(modelogbm,datos[c(974:980),],n.trees=100))


datos3=datos[,c(2:5,7)]
datos3_norm=as.data.frame(lapply(datos3,FUN=normalize))

# Se eliminan las columnas categóricas
datos3_norm=datos3_norm[-c(3,11,24)]
new_nrm=datos3_norm[c(974:980), ]
modeloresult_new=compute(modelorna,new_nrm[-1])
predic_new=modeloresult_new$net.result
ActualCantidad_new=unnormalize(predic_new,str_min,str_max)
RNA=round(ActualCantidad_new,0)
Dia=c("Jueves","Viernes","Sábado","Domingo","Lunes","Martes","Miércoles")
datospred1=data.frame(Dia,Poisson_I,Poisson,BN,
                      ST_mes,ST,
                      GB,RF,RNA)

 resul=kable(datospred1, row.names = FALSE, caption=titulo)
  return(resul)

}
```

```
newpred(datos,datospred1,"Cantidad predicha según el precio medio")
newpred(datos,datospred2,"Cantidad predicha según el precio máximo")
newpred(datos,datospred3,"Cantidad predicha según el precio mínimo")
newpred(datos,datospred4,"Cantidad predicha según la mediana
        de los precios")
```

Bibliografía

Alcaide, M. (2015). *Modelo de Regresión Binomial Negativa.* Trabajo fin de grado, Universidad de Sevilla.

Ataz, J. (2006). *Guía casi completa de BIBTEX.*
URL https://ctan.math.illinois.edu/info/spanish/guia-bibtex/guia-bibtex.pdf

Atoche, P. (2017). *Modelos de regresión con datos de conteo: aplicación a competiciones deportivas.* Trabajo fin de grado, Universidad de Sevilla.

Camps, G. (2003). *Predicción de series temporales.* Curso de Doctorado, Universidad de Valencia.

Carrascal, L. M. (2015). Teoría y praxis de modelos generalizados: infiriendo patrones con el paquete estadístico r.
URL https://digital.csic.es/bitstream/10261/128391/1/curso2015_an%c3%a1lisis%20con%20una%20variable%20respuesta.pdf

Céspedes, A. I. (2017). *Construcción de modelo de FORECAST para estimación de demanda en una empresa multinacional de retail.* Universidad Técnica Federico Santa María.
URL https://repositorio.usm.cl/handle/11673/41250

Chanialidis, C., & Dean, N. (2018). *Data Mining and Machine Learning I: Artificial Neural Networks (Introduction to Neural Networks and Regression).* University of Glasgow.

Crawley, M. J. (2012). *The R book*, chap. 13, Generalized Linear Models. John Wiley & Sons.

Cryer, J. D., & Chan, K.-S. (2008). *Time series analysis: with applications in R.* Springer Science & Business Media.

González, M. P. (2009). *Técnicas de predicción económica.* Economía Aplicada III/Ekonomia Aplikatua III, UPV/EHU.

Hardin, J., & Hilbe, J. (2012). *Generalized Linear Models and Extensions.* Texas: A State Press Publication StataCorp LP, College Station, third ed.

Instituto Nacional de Estadística (2012). Análisis espectral y ajuste estacional.
URL https://www.ine.es/clasifi/analisisyajuste.pdf

Jiménez, M. D. (2019). Apuntes asignaturas modelos lineales y series temporales.

Khakpour, A. (2020). *Data Science for Decision Support: Using Machine Learning and Big data in Sales Forecasting for Production and Retail*. Master's thesis.

López, E., & Ruiz, M. (2011). Análisis de datos con el modelo lineal generalizado. una aplicación con r. *Revista española de pedagogía*, (pp. 59–80).

Luque, P. L. (2017). *Escribir un Trabajo Fin de Estudios con R Markdown*.

Luque, P. L. (2019). *Cómo crear Tablas de información en R Markdown*.

McCullagh, P., & Nelder, J. A. (1989). *Generalized Linear Models*. London:Chapman and Hall.

McCullagh, P., & Nelder, J. A. (2019). *Generalized linear models*. Routledge.

Pichardo, J. M. (2020). Apuntes asignatura estadística computacional II.

Python and R tutorials (2019). *neuralnet: Train and Test Neural Networks Using R*. datascienceplus.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. Vienna, Austria.
URL https://www.R-project.org/

Román, V. (2019). *Introducción al Machine Learning: Una Guía Desde Cero*. Obtenido de ciencia & datos: https://medium. com/datos-y-ciencia ....

Saavedra, Y. A. (2019). *Creación de una red neuronal con R*.
URL https://yuasaavedraco.github.io/Docs/Redes_Neuronales_con_R.html

San-José, M. L. (2017). *Métodos estadísticos para la predicción del desvío en la demanda de energía eléctrica aplicado al mercado ibérico*. Trabajo fin de grado, Universidad Politécnica de Madrid.

Valencia, L. (2020). Apuntes asignatura inteligencia artificial.

Wikipedia (2021a). *Aprendizaje automático*.
URL https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico

Wikipedia (2021b). *Artificial neural network*.
URL https://en.wikipedia.org/wiki/Artificial_neural_network

Wikipedia (2021c). *Machine learning*.
URL https://en.wikipedia.org/wiki/Machine_learning