**Carga y Exploración**

**Introducción**

Es muy común que los conjuntos de datos públicos se almacenen en archivos de texto.

Los archivos de texto se pueden leer en prácticamente cualquier computadora o sistema operativo, lo que hace que el formato sea casi universal.

También se pueden exportar e importar desde y hacia programas como Microsoft Excel, lo que proporciona una manera rápida y fácil de trabajar con datos de hojas de cálculo.

**Como se presentan un archivo de datos tabular**

Un archivo de datos tabular (como en "tabla") está estructurado en forma de matriz, de manera que cada línea de texto refleja una observación, y cada observación tiene el mismo número de características.

Los valores de las características en cada línea están separados por un símbolo predefinido, conocido como **delimitador**.

A menudo, la primera línea de un archivo de datos tabular enumera los nombres de las columnas de datos. Esto se llama una **línea de encabezado**.

**Nuestros Datos**

Comenzaremos nuevamente cargando los datos a memoria con lo que hemos venido trabajando de esta manera:

subject\_name **<-** c**(**"John Doe", "Jane Doe", "Steve Graves"**)**

temperature **<-** c**(**98.1, 98.6, 101.4**)**

flu\_status **<-** c**(FALSE**, **FALSE**, **TRUE)**

gender **<-** factor**(**c**(**"MALE", "FEMALE", "MALE"**))**

blood **<-** factor**(**c**(**"O", "AB", "A"**)**, levels **=** c**(**"A", "B", "AB", "O"**))**

symptoms **<-** factor**(**c**(**"SEVERE", "MILD", "MODERATE"**)**,levels **=** c**(**"MILD", "MODERATE", "SEVERE"**)**, ordered **=** **TRUE)**

pt\_data **<-** data.frame**(**subject\_name, temperature, flu\_status, gender, blood, symptoms, stringsAsFactors **=** **FALSE)**

**CSV**

Quizás el formato de archivo de texto tabular más común es el archivo CSV (valores separados por comas, en siglas en ingles), que como su nombre indica, utiliza la coma como delimitador.

Los archivos CSV se pueden importar y exportar desde muchas aplicaciones comunes.

Vamos a almacenar nuestros datos de conjuntos médicos que hemos ido construyendo en un CSV de esta manera:

write.csv**(**pt\_data,'pt\_data.csv', row.names **=** **TRUE)**

Aquí estamos usando la función base de R que es write.csv, que es una de las funciones que podemos usar para escribir y exportar un csv, con R podemos exportar hacia una cantidad muy grande de extensiones, ya sea para Excel, PDF, JSON, SAS, SPSS, XML y muchas otras.

Dentro de los argumentos de la función ingresamos el objeto que será el futuro CSV, en este caso si vemos el código anterior vamos a convertir un dataframe a un CSV.

El objeto se llama pt\_data.

Seguido le pondremos el nombre con la extensión del archivo (En esta área también se puede poner una ruta distinta a la que tenga por defecto nuestro libro de R).

El argumento row.names es para indicar si el CSV lo mandaremos con encabezado o no. (En este caso va con encabezado)

**Comentario Adicional**

Después de recopilar los datos y cargarlos en las estructuras de datos de R, el siguiente paso en el proceso de aprendizaje automático consiste en examinar los datos en detalle.

Es durante este paso que comenzaremos a explorar las características y los ejemplos de los datos, y se darán cuenta de las peculiaridades que hacen que sus datos sean únicos. **Cuanto mejor comprenda sus datos, mejor podrá relacionar un modelo de aprendizaje automático.**

**Exploración de Datos**

La mejor manera de aprender el proceso de exploración de datos es con ejemplos y práctica.

En esta sección, exploraremos el conjunto de datos usedcars.csv, que contiene datos reales sobre autos usados recientemente anunciados para la venta en un popular sitio web de los EE. UU.

El CSV se los enviaré por correo electrónico, o lo colgare en la nube para su descarga, lamentablemente no tenemos acceso a Google Drive, por lo que revisaremos las opciones en las presentaciones presenciales en donde puedan descargar los archivos que vamos a utilizar.

**Buena Practica**

Para ir ordenándonos, como buena práctica, les sugiero crear varios archivos o carpetas que contendrán los procesos de analítica de datos en el que nos encontremos.

Generalmente yo creo las siguientes carpetas:

* **Domado:** Aquí coloco las cargas iniciales de datos sin depurar, en está carpeta está alojado el código que generalmente extrae, transforma y carga datos de otras fuentes.
* **Análisis:** En esta carpeta tengo el código que utilizo para el análisis de datos, generalmente tiene modelos estadísticos, geometrías y mapas.
* **Datos:** Carpeta en la que almaceno las fuentes de información, ya sean primarias o depuradas, en último caso creo subcarpetas de datos, dependiendo del alcance del proyecto y las distintas formas de domado de los datos.
* **Modelos:** Aquí tengo tanto el código como los algoritmos entrenados, hay codificación de modelos de correlación y todo lo que se pueda usar en un proyecto de Machine y Deep Leaning.
* **App:** En esta carpeta está el código de las aplicaciones web que se desarrollen usando las librerías de Shiny Server, y las interacciones con el usuario.

No es completamente necesario que se llamen igual, pero si deben de comenzar a dividir los procesos de su trabajo de una forma ordenada y a la cual puedan acceder de una manera eficiente y sepan de que se trata.

**Cargando nuestro primer CSV**

Posterior al haber descargado nuestro archivo y guardarlo en donde nosotros creamos conveniente veremos que el conjunto de datos proporcionado esta almacenado en CSV.

Para poder extraerlo y cargarlo a nuestro equipo vamos a utilizar la función de R base (existen otras funciones de otras librerías que veremos más adelante), para cargar nuestro archivo.

Para ver los argumentos de la función podemos utilizar la ayuda, refrescando un poco, lo consultaremos de esta manera en la consola de nuestro RStudio:

**?**read.csv**()**

Con esta instrucción nos aparecerá el uso y los argumentos que lleva la función de read.csv de esta manera.

read.csv**(**file, header **=** **TRUE**, sep **=** ",", quote **=** "\"",

dec **=** ".", fill **=** **TRUE**, comment.char **=** "", ...**)**

Nos aparecen también las descripciones de lo que hace y significa cada argumento, no quiero entrar en estos detalles pues consumiría espacio innecesario y la idea primordial es que comiencen a usar su software y practiquen en él.

Vamos a importarlo de esta manera.

usedcars **<-** read.csv **(**"usedcars.csv", stringsAsFactors **=** **FALSE)**

En este caso vamos a asignarle al objeto “usedcars” las observaciones que tenga el archivo CSV, le indicamos la ubicación del archivo (En su caso puede ser “data/usedcars.csv”) y el argumento de stringsAsFactors = False con el fin de que no nos convierta todos los textos en datos categóricos.

Cargar nuestro primer set de datos, al principio puede ser retador pues se está aprendiendo a usar la herramienta, no se desespere y recuerde que estoy para apoyarlos, así que, si no lo logran hacer, siéntanse en plena confianza de pedirme apoyo cuando consideren necesario.

**Tomando un Rol**

El dataframe de vehículos usados está listo, ahora asumiremos el papel de un **científico de datos** que tiene la tarea de comprender los datos de automóviles usados.

Si bien la exploración de datos es un proceso fluido, los pasos se pueden imaginar como una especie de investigación apoyadas en **el método científico** en la que las preguntas sobre los datos son respondidas. Las preguntas exactas pueden variar según los proyectos, pero los tipos de preguntas son siempre similares.

Nuestros pasos básicos:

* Observación
* Investigación
* Hipótesis
* Experimentación
* Análisis
* Conclusiones

Nuestra tarea final es que debemos de poder adaptar los pasos básicos de esta investigación a cualquier conjunto de datos que desee, ya sea grande o pequeña.

**Explorando la estructura de datos**

Una de las primeras preguntas que debe hacerse en una investigación de un nuevo conjunto de datos, debe ser sobre cómo se organiza el mismo **(Observación)**.

Si tiene suerte, su fuente (Ingeniero de Datos), proporcionará un diccionario de datos, que es un documento que describe las características del conjunto de datos.

En nuestro caso, los datos de automóviles usados no vienen con esta documentación, por lo que necesitaremos crear una.

**Función STR()**

Esta función proporciona un método para mostrar la estructura de las estructuras de datos en R, como marcos de datos, vectores o listas.

Esta función se puede utilizar para crear el esquema básico de nuestro diccionario de datos.

Nuestro código entonces es el siguiente:

str**(**usedcars**)**

'data.frame'**:** 150 obs. of 6 variables**:**

**$** year **:** int 2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...

**$** model **:** chr "SEL" "SEL" "SEL" "SEL" ...

**$** price **:** int 21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...

**$** mileage **:** int 7413 10926 7351 11613 8367 25125 27393 21026 32655 36116 ...

**$** color **:** chr "Yellow" "Gray" "Silver" "Gray" ...

**$** transmission**:** chr "AUTO" "AUTO" "AUTO" "AUTO" ...

Esta simple función, nos permite observar una gran cantidad de información sobre esté conjunto de datos.

**Estructura de los Datos:**

Nos indica que estructura de datos en R estamos usando, en este caso ‘DataFrame’

**Observaciones**

La declaración 150 obs. nos informa que los datos incluyen 150 observaciones, que es solo otra forma de decir que el conjunto de datos contiene 150 registros o ejemplos.

El número de observaciones a menudo se abrevia como n.

Como sabemos que los datos describen autos usados, ahora podemos suponer que tenemos ejemplos de **n = 150.**

**Variables**

La declaración de **6 variables** se refiere a las seis características que se registraron en los datos.

Estas características se enumeran por nombre en líneas separadas. Mirando la línea para la característica llamada color, podemos notar algunos detalles adicionales:

**Variable $color**

chr

"Amarillo" "Gris" "Plata" "Gris" ...

Después del nombre de la variable, la etiqueta “**chr**” nos dice que la característica es el tipo de carácter.

En este conjunto de datos, tres de las variables son caracteres, mientras que tres se indican como “**int**”, lo que indica un tipo de entero.

Aunque el conjunto de datos usedcars incluye solo caracteres y variables enteras, también es probable que encuentre números o tipos numéricos mientras usa datos no integrales.

En este caso podríamos cambiar dos de las variables de chr a factores, ya que tienen secuencias categóricas como color y transmisión muy visibles.

**Variables o Predictores**

Dentro de este análisis exploratorio tenemos que aprender a ver los datos de un modo distinto, en este caso vamos a listar las variables o predictores.

He querido escribir desde capítulos anteriores esto, pero aún faltaba terminar las estructuras de datos en R, entendiendo ya lo de la estructura, podemos comenzar a clasificarlas de esta forma.

Aprender a clasificar bien nuestras variables de datos nos va a garantizar el éxito del entrenamiento de nuestro algoritmo en caso el proyecto sea de crear un modelo de inteligencia artificial.

**Variable o Predictor Categórico**

Las variables categóricas contienen un número finito de categorías o grupos distintos. Los datos categóricos pueden no tener un orden lógico.

Por ejemplo:

Los predictores categóricos incluyen sexo, tipo de material, método de pago, municipio, departamento, país.

Todas nuestras variables categóricas las clasificaremos en estructuras de factores, teniendo especial cuidado en los niveles y etiquetas que puedan tener.

**Variable o Predictor Discreto**

Las variables o predictores discretos son variables numéricas que tienen un número contable de valores entre dos valores cualesquiera.

Una variable discreta **siempre es numérica**.

Por ejemplo:

Los predictores discretos incluyen el número de quejas de los clientes o el número de fallas o defectos encontrados en el item de la queja.

**Variable o Predictor Continuo**

Los predictores o variables continuas son variables numéricas que tienen un número infinito de valores entre dos valores cualesquiera.

Una variable continua puede ser numérica o de fecha/hora.

Por ejemplo:

Fecha y hora que se recibe un pago o el tiempo de una carrera de una persona.

Un ejemplo práctico en el caso de como separar una variable predictora continua puede ser la siguiente:

Imagine que usted sale a correr todos los días y se toma el tiempo en horas, minutos y segundos.

Si usted hace eso todos los días, sus resultados siempre van a ser continuos es decir no serán los mismos ya que el día cambio y los resultados no dependen únicamente del tiempo, también dependen de por ejemplo si desayuno o no ese día, si se desvelo o no, si tiene o no una buena noticia, si tiene problemas familiares, personales o laborales antes, si esta corriendo en asfalto, tierra o arena, si llueve o hace sol, si hace frio o calor, si sus tenis son nuevos, usados o viejos, si sus calcetines le quedaron bien ese día o no.

Es decir, el predictor continuo depende de factores externos que en muchos casos pueden variar, por lo tanto, el resultado nunca será el mismo, en la clase práctica vamos a ahondar más en esto.