USER MANUAL

AlchemyML **Product**

AlchemyML

Alchemy Machine Learning, S.L.
alchemyml.com // Twitter: @AlchemyML
Email: hello@alchemyml.com

Contenido

Tabla de figuras

1. Características de AlchemyML

¿Qué tipo de ficheros acepta AlchemyML?

AlchemyML es capaz de leer datos de diferentes tipos de fuentes de información como excel, csv, json, xml, sql, arff y logs.

- **Excel: .XLSX, .XLS**. AlchemyML diferencia si sus ficheros Excel se componen de una o más hojas de cálculo.

  Por el momento, es capaz de gestionar conjuntos de datos que contengan la información en una única hoja Excel.

  Es decir, si el conjunto de datos históricos que desea subir se encuentra distribuido en varias hojas de cálculo de un mismo documento, AlchemyML le notificará tanto vía WEB como vía API de que esta carga no está permitida.

- **Archivos separados por comas (y no solo por comas): .CSV.**
- **Información transmitida en formato .JSON.** Los ficheros de tipo JSON son un tipo de archivo escritos en un formato de texto sencillo para el intercambio de datos.

  Tienen la particularidad de poder anidar objetos dentro de otros objetos y de poder estructurarlos libremente. Véase: https://cutt.ly/ytH09TL

  Debido a esta característica, es necesario *parsear* el fichero e identificar el lugar donde se encuentra un conjunto de datos dentro del árbol, pero además también es necesario conocer su estructura.

  Actualmente, AlchemyML realiza un trabajo interesante para encontrar un conjunto de datos estructurado dentro de un fichero como lo es el archivo en formato JSON.

Pero no nos engañemos. AlchemyML no es infalible y seguramente a usted le gustaría que le recomendásemos algún tipo de estructura que pueda seguir si quiere subir sus archivos en formato JSON.

Está bien… Si AlchemyML falla puede subir sus datos siguiendo alguno de los siguientes esquemas (y también puede escribirnos e informarnos sobre la incorrecta lectura del fichero en cuestión para que podamos estudiar su estructura y seguir mejorando):

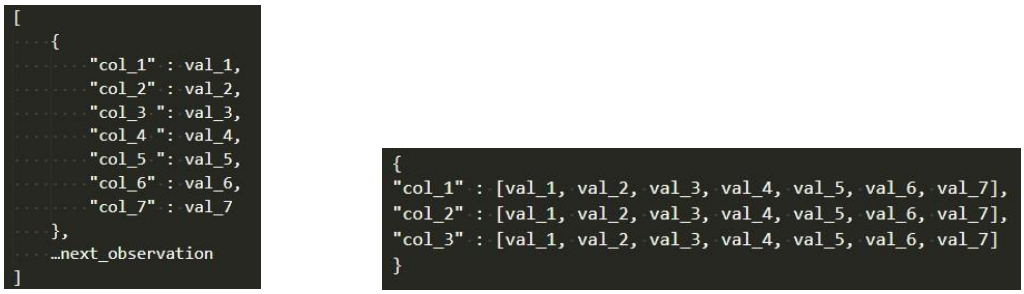


Figura 1 Estructuras recomendadas para la subida de ficheros JSON.

- **Ficheros .XML.** De forma similar a los archivos de tipo JSON un fichero .xml también ha de ser *parseado* para encontrar el contenido que se encuentra anidado a lo largo del árbol del fichero.

  En el caso de que el parseo de AlchemyML no funcione, se le recomienda seguir la siguiente estructura si quiere subir su archivo como un fichero .xml.

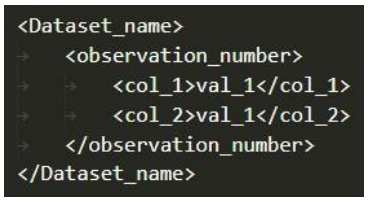


Figura 2 Estructura de ficheros recomendada para la subida de ficheros XML

- **Ficheros de bases de datos relacionales: .SQL.** Usted puede subir la tabla de su base de datos relacional para ser trabajada como un *dataset*. Para utilizar los conectores con bases de datos relacionales póngase en contacto con nosotros en hello@alchemyml.com o admin@alchemyml.com

- **En desarrollo: Ficheros .LOG.** En estos momentos estamos buscando la manera de automatizar la lectura para las estructuras más típicas en las que suelen escribirse los logs.

- **Ficheros .ARFF.** Los archivos ARFF tienen dos secciones distintas.

  La primera sección es la Información de ***Header***, seguida de la Información de datos en la sección ***Data***.

  La cabecera del fichero ARFF contiene el nombre de la relación, una lista de los atributos (las columnas de los datos) y sus tipos.

¿Qué condiciones debe cumplir un fichero para poder subirlo?

Le recomendamos que siga las siguientes indicaciones si desea que la subida de su fichero resulte exitosa:

- Su fichero debe estar entre una de las extensiones reconocidas por AlchemyML: excel, csv, json, xml, sql, arff y logs. Le informamos de que los formatos con mejor respuesta de subida (y por tanto, los más trabajados por el equipo de AlchemyML) son: Excel, csv, json y sql.

- Evite que sus ficheros se guarden con doble extensión. Por ejemplo *./Iris.xslx.csv*.

- En AlchemyML requerimos que los ficheros que suba como mínimo contengan 50 observaciones.

- Tenga en cuenta que para la versión WEB de AlchemyML, el tamaño de fichero máximo admisible está limitado a 20MB. En el caso de que usted y/o su equipo formen parte del modelo de alianzas o *partnership* este límite será consensuado entre las partes.

- Finalmente, asegúrese de que sus datos están considerablemente completos.

Primer preproceso

Cuando AlchemyML comienza un nuevo experimento, realiza una primera adaptación de los datos para poder continuar trabajando con unos datos que se ajusten a un formato adecuado. A continuación, le indicamos cuáles son las tareas que se llevan a cabo durante este proceso:

- Se quitan las **comillas.**

    - Por ejemplo, si el elemento de una celda lleva por valor “3.2”, se remueven las comillas para conseguir un string que sea 3.2. En las operaciones siguientes se seguirá tratando con este tipo de dato.

- Se eliminan los **porcentajes** y se da el formato correspondiente.

    - Imagine que se ha registrado el porcentaje de humedad relativa de un ambiente y el dato viene dado como “%72.4”. El objetivo es convertir este dato que viene como una cadena de caracteres a un valor de coma flotante 0.724.

- Transformación de dígitos (+/-) a verdadero formato **numérico.**

    - En ocasiones un valor de -0.72, por ejemplo, es leído como string cuando en realidad debería ser tratado como un número de coma flotante.

- Transformación de los números que vienen como **"strings"** en formato **numérico**.

    - Esta transformación es la operación realizada tras haber retirado las comillas de un elemento, tal y como le mencionábamos en el primer punto.

- Conversión de columnas candidatas a tipo "Fecha" en **DateTime**.

    - Esta operación busca posibles columnas de tipo Fecha y las convierte a ese mismo tipo.

Para muestra un botón:

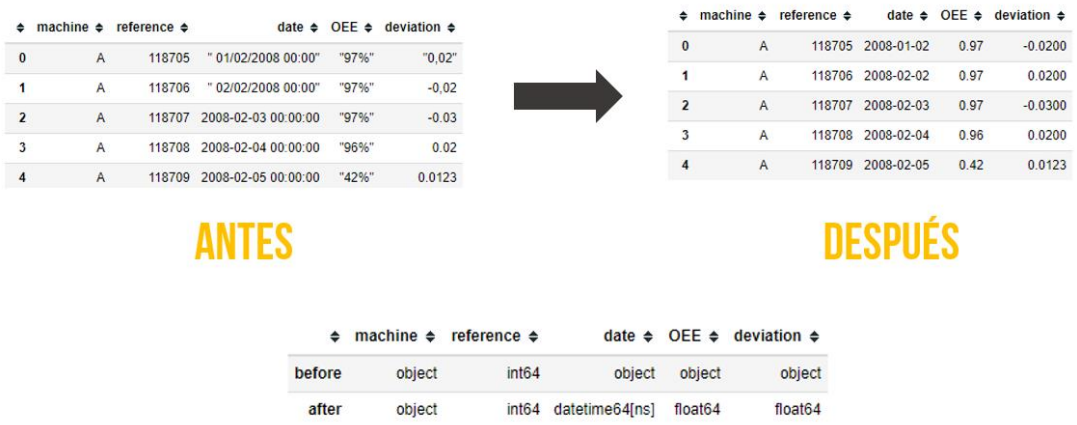


Figura 3 Ejemplo de visualización del primer preproceso llevado a cabo por AlchemyML.

Comprobación de la validez de la columna de interés

Una vez adecuado el formato de los datos, AlchemyML determinará si la columna de salida elegida como variable a predecir, clasificar, prever, cumple con las condiciones para poder continuar con el experimento.

En ese sentido, AlchemyML comprobará que su variable de salida no sea **constante**, de formato **Fecha** o una columna **demasiado vacía**, casos en los que resultaría improbable poder llevar un experimento de Machine Learning adelante.

Análisis de balanceo en la columna Target y Balanceo

Si la columna de salida ha pasado el filtro anterior, entonces, se identificará su tipo. En el caso de que la columna sea de tipo categórica o booleana, se analizará si su *dataset* está balanceado. En la siguiente figura puede observar dos casos diferentes de un *dataset* desbalanceado, donde existe una gran desproporción entre la clase mayoritaria y las clases minoritarias.

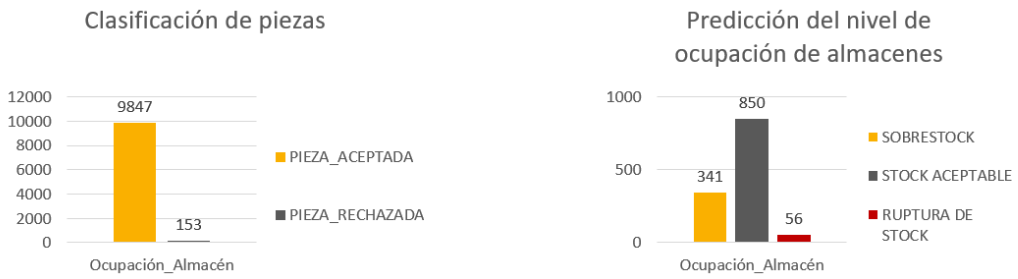


Figura 4 Ejemplos de datasets desabalanceados.

Continuando con la explicación, si resulta que su *dataset* está desbalanceado, AlchemyML estudiará si es prudente balancearlo. En caso negativo, se detendrá el experimento. Sin embargo, en caso afirmativo, AlchemyML determinará cuál es la mejor estrategia para balancear sus datos automáticamente.

AlchemyML utiliza dos tipos de tácticas para balancear datos:

- Manipulación de los algoritmos

    - Consiste en la asignación de pesos a los algoritmos para dar la misma importancia a la hora de clasificar a todas las clases, evitando el sesgo que tienen los algoritmos a la hora de favorecer la clasificación hacia la clase mayoritaria.

      Por favor, no entre en pánico. Usted no deberá preocuparse por este tipo de tareas. Recuerde que AlchemyML nació para facilitarle la vida y no para complicársela.

- Manipulación de los datos

    - Se basa en la creación artificiales y/o supresión de muestras reales parar conseguir un conjunto de datos más proporcionado.

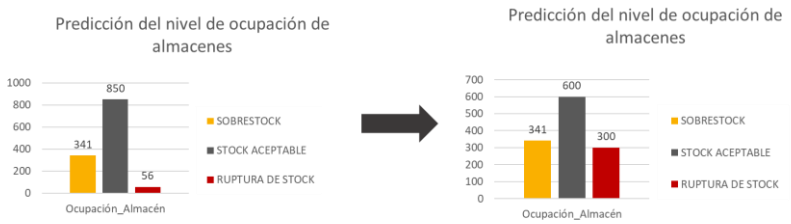


Figura 5 Ejemplo de balanceo de un dataset a través de la manipulación de los datos.

Manejo de inconsistencias

Otra de las operaciones que AlchemyML realiza por usted es manejar las inconsistencias encontradas en los datos. Aunque el término *inconsistencia* puede resultar un poco difuso, en AlchemyML llamamos inconsistencias a aquellos valores que han sido introducidos erróneamente.

Por ejemplo:

- Un operario que haya introducido erróneamente ACCCEPTED en lugar de ACCEPTED.

- Si se encuentra alguna ocurrencia puntual como casos en los que un valor que debería ser un 3, por ejemplo, aparece una vez escrito como 333333333333.

Eliminación de columnas consideradas como no válidas

A la hora de crear un experimento, también es importante tener en cuenta que solamente hay que contar con los datos imprescindibles, aquellas variables que aportan valor.

AlchemyML eliminará de su conjunto de datos columnas con valor **constante**, columnas con **URLS**, **emails** y **textos largos** y **columnas extremadamente vacías**.

Eliminación de columnas altamente correlacionadas

Además de los anteriores tipos de columnas, también se eliminarán columnas duplicadas y columnas altamente correlacionadas entre ellas y también con la variable de salida.

Una columna duplicada, en fin, todos y todas sabemos lo que es una columna duplicada. Sin embargo, qué significa que una variable esté altamente correlacionada con otra.

Pueden darse dos tipos de correlaciones lineales:

- Directamente:

número de unidades de X	precio €
1	3,5
2	8
3	10,5
4	13
5	17,5
6	21
7	24,5
8	28
9	31,5
10	35
11	37
12	43
13	45,5
14	49
15	52,5
16	56
17	58,5
18	63

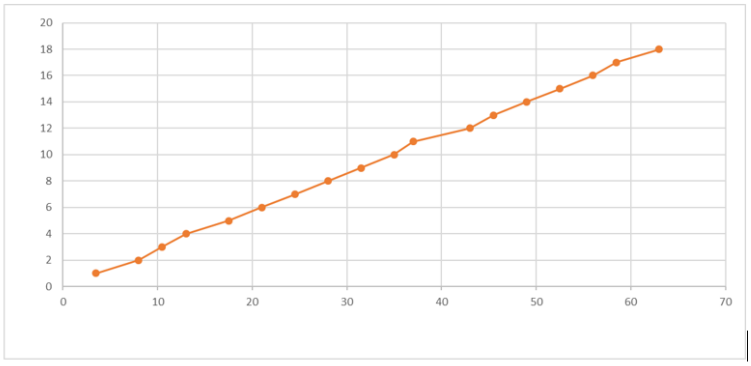


Figura 6 Ejemplo de dos columnas alta y directamente relacionadas.

A medida que aumenta el número de unidades de un producto X, aumenta prácticamente en la misma medida el precio a pagar por *n* unidades de ese producto. Cuando el grado de correlación entre las variables es tan alto, las dos están aportando prácticamente la misma información al conjunto de datos. En ese caso se eliminaría una de ellas.

- Inversamente

Número de errores en el código	Número de errores corregidos
84	1
82	2
80	3
78	4
76	5
74	6
72	7
70	8
68	9
66	10
64	11
62	12
60	13
58	14
56	15
54	16
52	17
50	18

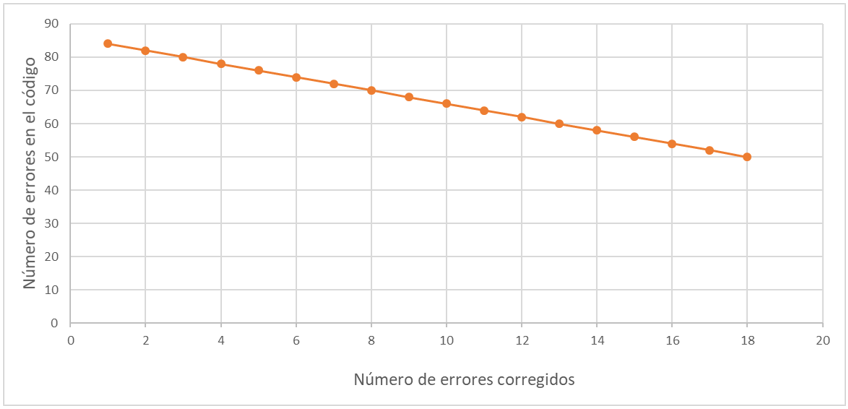


Figura 7 Ejemplo de dos columnas alta e inversamente relacionadas.

Eliminación de filas duplicadas

Al igual que sucedía con las columnas duplicadas, AlchemyML eliminará las filas duplicadas por usted.

Area(m^2)	Número_Hab	Número_Baños	Número_Balcones	Orientación	Jardín	Area_jardín(m^2)	Núcleo_Urbano	REFORMADA	FIANZA(MESES)	EFICIENCIA_ENERG	NÚMERO_PISO	DUPLEX	ASCENSOR	CONSERJE
30	1	1	0	Norte	NO	0	SÍ	SI	2	A	2	NO	SI	NO
40	1	1	0	Norte	NO	0	SÍ	NO	2	A	2	NO	SI	NO
50	1	1	1	Norte	NO	0	SÍ	SI	1	B	1	NO	SI	NO
60	2	1	1	Norte	NO	0	SÍ	NO	2	B	3	NO	SI	NO
60	2	1	1	Norte	NO	0	SÍ	NO	2	B	3	NO	SI	NO
65	1	1	1	Norte	NO	0	SÍ	SI	2	B	2	NO	SI	NO
62	2	1	1	Norte	NO	0	SÍ	NO	1	B	2	NO	SI	NO
67	1	1	1	Norte	NO	0	SÍ	SI	2	B	9	NO	SI	NO
72	2	1	1	Norte	NO	0	SÍ	NO	2	B	2	NO	SI	NO
80	1	1	1	Norte	NO	0	SÍ	SI	2	A	2	NO	SI	NO

Figura 8 Ejemplo de dataset con filas duplicadas.

Manejo de columnas vacías

Uno de los problemas más comunes a los que se enfrenta un/una analista o científico/a de datos durante el *Data Cleaning*, es el manejo de los valores faltantes.

Cuando no se almacena ningún valor de datos para la variable en una observación, se dice que nos encontramos ante un valor faltante o *missing value*.

Este valor faltante puede ser provocado (por ejemplo, una persona encuestada que se niega a responder una de las preguntas del cuestionario) o involuntario (un sensor que se deteriora, un sistema informático cuya red cae, entre otros).

Cuando existen *missing values* en su *dataset*, AlchemyML intentará tratarlos si la cantidad de valores faltantes y su patrón de ocurrencia son razonables.

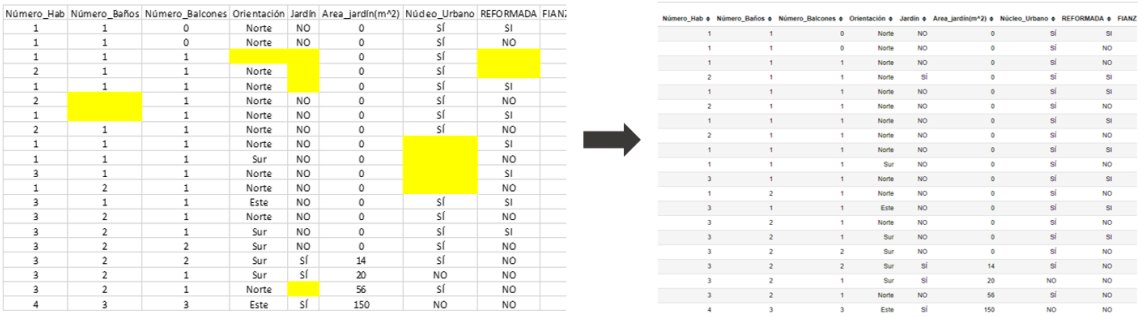
No es correcto rellenar una columna en la que el 98% de los valores son valores faltantes pues la máquina estaría aprendiendo de un conjunto de datos en el que una parte de la información ha sido rellenada sin conocer la realidad.

AlchemyML cuenta con diferentes técnicas para llevar a cabo dicha imputación de valores faltantes de manera automática. A continuación, le mencionamos cuáles son dichas estrategias:

- Rellenar fechas faltantes si se encuentra un patrón evidente en la columna fecha de los datos.
- Rellenar con cero
- Rellenar los valores aleatoriamente
- Rellenar teniendo en cuenta el porcentaje de aparición de los datos que sí existen dentro de una columna
- Rellenar con el valor menos frecuente: *antimoda*.
- Eliminación de filas
- Relleno de missing values empleando la información temporal, si la hubiera.

Para cada columna de su dataset AlchemyML determinará qué estrategia es la más adecuada.

La finalidad es conseguir unos datos completos, si bien estos pueden no ajustarse perfectamente a lo que hubiera sucedido en la realidad.



Número_Hab	Número_Baños	Número_Balcones	Orientación	Jardín	Area_jardín(m^2)	Núcleo_Urbano	REFORMADA	FIANZ
1	1	0	Norte	NO	0	SÍ	SI	
1	1	0	Norte	NO	0	SÍ	NO	
1	1	1			0	SÍ		
2	1	1	Norte		0	SÍ		
1	1	1	Norte		0	SÍ	SI	
2		1	Norte	NO	0	SÍ	NO	
1		1	Norte	NO	0	SÍ	SI	
2	1	1	Norte	NO	0	SÍ	NO	
1	1	1	Norte	NO	0		SI	
1	1	1	Sur	NO	0		NO	
3	1	1	Norte	NO	0		SI	
1	2	1	Norte	NO	0		NO	
3	1	1	Este	NO	0	SÍ	SI	
3	2	1	Norte	NO	0	SÍ	NO	
3	2	1	Sur	NO	0	SÍ	SI	
3	2	2	Sur	NO	0	SÍ	NO	
3	2	2	Sur	SÍ	14	SÍ	NO	
3	2	1	Sur	SÍ	20	NO	NO	
3	2	1	Norte		56	SÍ	NO	
4	3	3	Este	SÍ	150	NO	NO	

Número_Hab	Número_Baños	Número_Balcones	Orientación	Jardín	Area_jardín(m^2)	Núcleo_Urbano	REFORMADA	FIANZ
1	1	0	Norte	NO	0	SÍ	SI	
1	1	0	Norte	NO	0	SÍ	NO	
1	1	1	Norte	NO	0	SÍ	NO	
2	1	1	Norte	SÍ	0	SÍ	SI	
1	1	1	Norte	NO	0	SÍ	SI	
2	1	1	Norte	NO	0	SÍ	NO	
1	1	1	Norte	NO	0	SÍ	SI	
2	1	1	Norte	NO	0	SÍ	NO	
1	1	1	Norte	NO	0	SÍ	SI	
1	1	1	Sur	NO	0	SÍ	NO	
3	1	1	Norte	NO	0	SÍ	SI	
1	2	1	Norte	NO	0	SÍ	NO	
3	1	1	Este	NO	0	SÍ	SI	
3	2	1	Norte	NO	0	SÍ	NO	
3	2	1	Sur	NO	0	SÍ	SI	
3	2	2	Sur	NO	0	SÍ	NO	
3	2	2	Sur	SÍ	14	SÍ	NO	
3	2	1	Sur	SÍ	20	NO	NO	
3	2	1	Norte	NO	56	SÍ	NO	
4	3	3	Este	SÍ	150	NO	NO	

Figura 9 Antes y después de un dataset tras pasar por la operación de imputar valores faltantes.

Codificación de columnas categóricas

Otra de las tareas que hay que realizar durante el preprocesamiento o preparación de los datos es manejar las columnas categóricas/boleanas y codificarlas. Codificar, en este caso, significa convertir todo valor que representa una categoría (Verdadro/Falso o Mucho/Poco/Nada, etc) a un código numérico (Verdadero =1, Falso = 0). El valor del número per se 1no es relevante.

Esto es así porque muchos algoritmos de Machine Learning necesitan aprender de un conjunto de datos en el que todos los valores son numéricos. A continuación, se muestra en la figura un antes y después de la operación de codificación de variables categóricas:

[1] La locución latina ***per se*** significa 'por sí' o 'por sí mismo'.

Area(m^2)	Número_Hab	Número_Baños	Número_Balcones	Orientación	Jardín	Area_jardín(m^2)	Núcleo_Urbano	REFORMADA	FIANZ
30	1	1	0	Norte	NO	0	SÍ	SI	
40	1	1	0	Norte	NO	0	SÍ	NO	
50	1	1	1	Norte	NO	0	SÍ	SI	
60	2	1	1	Norte	NO	0	SÍ	NO	
65	1	1	1	Norte	NO	0	SÍ	SI	

Area(m^2)	Número_Hab	Número_Baños	Número_Balcones	Orientación	Jardín	Area_jardín(m^2)	Núcleo_Urbano	REFORMADA	FIANZ
30	1	1	0	1	0	0	1	1	
40	1	1	0	1	0	0	1	0	
50	1	1	1	1	0	0	1	1	
60	2	1	1	1	0	0	1	0	
65	1	1	1	1	0	0	1	1	

Figura 10 Ejemplo de codificación de columnas categóricas en numéricas.

Unificación de columnas candidatas a tipo "Fecha"

El objetivo de este tipo de operación es detectar si a través de las diferentes columnas de un *dataset* la variable tiempo está distribuida en distintas columnas, siendo cada columna alguna de las diferentes unidades temporales posibles para una fecha (Año | Mes |Día | Hora | Minutos| Segundos|).

Una vez detectadas, las columnas se unificarán en una sola columna ('Fecha') que se colocará como índice del *dataset*.

Esta operación solamente se lleva a cabo automáticamente si en el paso siguiente se determina que el tipo de experimento a llevar a cabo es un experimento de series temporales.

Detección del tipo de experimento

Cuando usted se encuentre utilizando AlchemyML, ya sea dentro de la plataforma en el área de clientes o realizando peticiones a través de nuestra API, en el momento de creación de un experimento podrá especificar qué tipo de experimento quiere llevar a cabo.

Dicho esto, usted podrá elegir entre las siguientes opciones:

- Clasificación
- Regresión
- Time Series
- Auto Detect

Si usted tiene conocimientos sobre Machine Learning y conoce cuál es el tipo de reto al que se enfrenta puede indicárselo a AlchemyML a través de cualquiera de las tres primeras opciones.

Si, por el contrario, usted no tiene los conocimientos necesarios, le invitamos a que primero lea la explicación para realizar su selección si se encuentra trabajando en la plataforma de AlchemyML.

Si aún no consigue esclarecer sus dudas o simplemente quiere poner a prueba las capacidades de AlchemyML, seleccione la opción Auto Detect.

Sea cual sea su selección, AlchemyML contrastará la información aportada con la extraída a partir de las propiedades de los datos.

Si su opción seleccionada no es descabellada y puede llevarse a cabo el experimento, AlchemyML le dejará continuar a sabiendas de que, si los resultados son malos, este hecho pueda deberse a un planteamiento incorrecto sobre el tipo de experimento a realizar.

Selección de variables

El tamaño de un conjunto de datos (número de filas y columnas) es vital para acelerar el rendimiento de los algoritmos de Machine Learning. No todas las columnas de un determinado conjunto de datos tienen la misma importancia, y tal vez algunas de ellas son prescindibles debido a la falta de información significativa. Si pudiéramos borrar estas columnas prescindibles, podríamos acelerar el proceso y reducir los tiempos de entrenamiento de los algoritmos.

Además, en algunos casos, una columna que contenga datos engañosos puede perjudicar la calidad del modelo ajustado, por lo que no sólo es deseable, sino también necesario suprimir las columnas no pertinentes.

Por ello, AlchemyML está dotado de un módulo de selección de variables relevantes cuya finalidad es buscar aquellas variables que aportan el mayor valor a su conjunto de datos. Todo ello se realiza antes de construir el modelo predictivo.

Detección de Outliers

Un valor atípico es un valor de una observación que parece estar muy lejos y difiere de un patrón general en una muestra de datos.

Tomemos un ejemplo, hacemos un estudio de perfiles de clientes y averiguamos que el ingreso anual promedio de los clientes es de 29.235€. Pero, hay dos clientes que tienen ingresos anuales de 150.000€ y 2.000€. Estos dos clientes tienen unos ingresos anuales muy diferentes a los del resto de la población. Estas dos observaciones se considerarán valores atípicos que, pueden deberse a un error o no.

Errores de entrada de datos: Los errores humanos, tales como los errores causados durante la recolección de datos, el registro o la entrada de datos, pueden causar valores

atípicos en los datos.

Por ejemplo: El ingreso anual de un cliente es de $100,000. Accidentalmente, el operador de entrada de datos pone un cero adicional en la figura. Ahora los ingresos se convierten en 1.000.000 de dólares.

Un valor atípico puede deberse a:

- **Errores de entrada de datos**: Los errores humanos, tales como los errores causados durante la recolección de datos, el registro o la entrada de datos, pueden causar valores atípicos en los datos.

  Por ejemplo: El ingreso anual de un cliente es de $100,000. Accidentalmente, el operador de entrada de datos pone un cero adicional en la figura. Ahora los ingresos se convierten en 1.000.000 de dólares.

- **Error de medición**: Es una de las fuentes más comunes de valores atípicos. Se produce cuando el instrumento de medición utilizado resulta defectuoso. Por ejemplo: un sensor de temperatura que, debido a una sobreintensidad que circulaba por el circuito que lo alimentaba, ha resultado dañado y debido a las condiciones ambientales, ahora cada vez que se calienta da mediciones fuera de rango.

- **Intencionalmente atípico**: Esto se encuentra comúnmente en las medidas autodeclaradas que involucran datos sensibles.

Por ejemplo, los adolescentes típicamente no reportan la cantidad de alcohol que consumen. Sólo una fracción de ellos reportaría el valor real.

En este momento el equipo de AlchemyML está trabajando para identificar los motivos de un valor atípico en sus datos y así poder darle un tratamiento mucho más específico e individualizado. Sin embargo, AlchemyML es capaz de detectar estos valores atípicos y tratarlos a través de diferentes metodologías:

- Eliminación de observaciones
- Imputación de la media
- Imputación de la mediana
- Interpolaciones
- Imputación tomando en cuenta la información temporal, si existiera.

Construcción del modelo predictivo

Por si no fuera suficiente con las etapas de limpieza y preprocesamiento de los datos, AlchemyML también cubre la etapa de creación, entrenamiento y ajuste del modelo predictivo.

Para ello, AlchemyML cuenta con un conjunto de algoritmos de Machine Learning preparados para resolver experimentos de clasificación, regresión y series temporales.

Esta etapa, si bien no representa en porcentaje el mayor número de horas para una persona dedicada a la ciencia de datos (habitualmente el preproceso, limpieza y enriquecimiento de los datos supone el 80% del tiempo mientras al modelado le corresponde el 20% de las horas totales), sí resulta de vital importancia para la entrega de resultados fiables.

Si usted trabaja dentro de este sector lo que vamos a explicar a continuación le resultará familiar.

Cuando una persona especialista en la construcción de modelos de Machine Learning se enfrenta a esta fase de modelado tiene que cubrir varios aspectos:

- **Elegir el algoritmo adecuado para cada experimento**. Si usted tuviera que colocar un cuadro en la pared de su casa, ¿Utilizaría un martillo, una zapatilla o un ladrillo para clavarlo? Las tres opciones pueden terminar dando el mismo resultado, eso es cierto. Sin embargo, lo más sensato sería utilizar el…martillo, sí. (Por un momento dudó sobre nuestra merecida existencia en la tierra).

  A pesar de que existen pautas y recomendaciones a la hora de elegir el algoritmo apropiado, este es un trabajo de ensayo y error y requiere mucha experiencia.

- **Ajustar los parámetros del algoritmo escogido**. Este es otro de los momentos más delicados durante la construcción del modelo. Cada algoritmo de Machine Learning tiene una serie de parámetros que hay que configurar. Para que se haga una idea, sería como tener delante de usted el panel de mandos que controla una radio emisora mundial. Dependiendo de qué emisora quisiera escuchar, sería necesario realizar un tipo de ajuste u otro.

  Dentro de esos ajustes, existen unos que son más fáciles de resolver (como lo sería seleccionar el continente, el país o la región de la emisora donde se retransmite el programa de radio que desea escuchar.

  Sin embargo, ajustar la frecuencia es algo más complicado ya que hay que tener en cuenta que existen frecuencias que se solapan unas a otras. Por último, quedaría afinar la frecuencia para escuchar con total claridad el programa y ajustar el volumen.

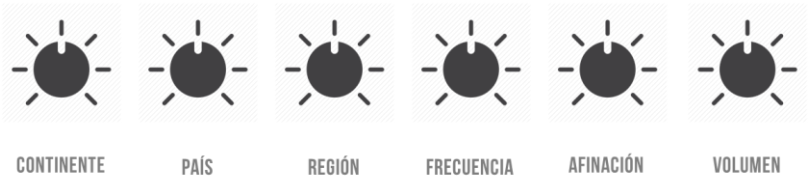


Figura 11 Analogía del ajuste de parámetros con la sintonización de una emisora de radio.

A nivel matemático, estos parámetros son parámetros de regulación, de semillas, selección del número de estimadores, tipo de estrategia a utilizar durante el aprendizaje del algoritmo, etc.

En fin, un *rollo macabeo*[2] del que estaría muy bien desentenderse y que alguna herramienta lo hiciera de manera automática. ¿Adivina cuál?

- **Entrenarlo y evaluar su desempeño**. Digamos que el entrenamiento es algo automático y que es cuestión de tiempo. Se trata de ejecutar la instrucción que hace que el algoritmo aprenda de su histórico de datos.

  No obstante, si es importante evaluar el desempeño del algoritmo y hacerlo bien ya que existen diversas técnicas.

  Asegurar que no se está cometiendo un sobreajuste de los datos u *overfitting*. Esto sería equivalente a estudiarse de memoria los ejercicios realizados durante las clases para un examen y esperar que el profesor no cambie los ejercicios.

  El verdadero objetivo sería aprender a adquirir el conocimiento necesario para poder extrapolar lo aprendido a cualquier ejercicio que pudiera poner el profesor.

[2] La expresión **rollo macabeo** se utiliza como sinónimo para expresar algo largo y pesado.

> De forma análoga, el algoritmo de aprendizaje utilizado debe alcanzar un estado en el que sea capaz de predecir el resultado en otros casos a partir de lo aprendido con los datos de entrenamiento, generalizando para poder resolver situaciones distintas a las acaecidas durante el entrenamiento.
>
> Cuando un sistema se entrena demasiado (se sobreentrena) o se entrena con datos extraños, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo. A este hecho se le llama sobreajuste u ***overfitting.***

Si todavía no siente agobio alguno y ha conseguido terminar la lectura siendo consciente del trabajo que toma modelar experimentos de Machine Learning, nos alegra comunicarle que AlchemyML es capaz de encontrar el algoritmo más adecuado para su conjunto de datos, ajustar los parámetros del algoritmo escogido y asegurar que no hay un sobreajuste cuando se le entreguen los resultados.

Pero aún nos queda un último detalle. En el caso de que su experimento sea un experimento de series temporales, AlchemyML realizará las siguientes operaciones también en esta fase:

- **Limpieza de los albores de los tiempos de la serie**: En ocasiones cuando se registra un suceso que depende de su evolución en el tiempo sucede que al principio los datos recogidos no son fiables. Esto puede tener diferentes explicaciones:
  - El comienzo de un nuevo proyecto
  - La calibración inicial necesaria del instrumento de medición
  - Etc.

  El objetivo sería obtener una serie limpia como se muestra en la siguiente figura:

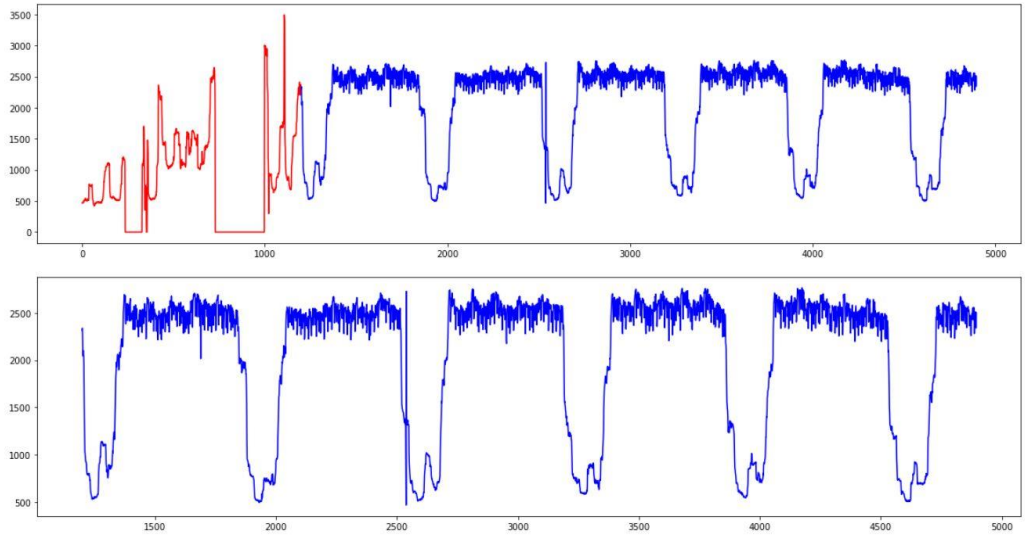


Figura 12 Limpieza de los primeros valores de una serie temporal.

- **Manejo de outliers:** Aunque en fases anteriores le hablábamos sobre la detección de valores atípicos, en el caso de las series temporales importa conocer la evolución del valor de la variable de interés.

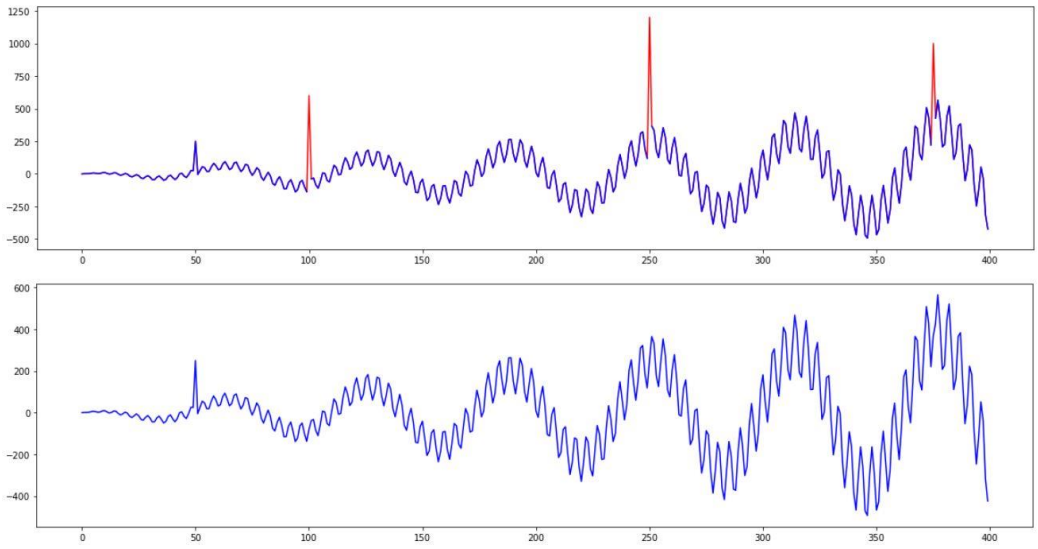


Figura 13 Detección y tratamiento de outliers en una serie temporal.

- **Manejo de valores faltantes:** Este fenómeno suele darse cuando un sensor deja de funcionar y deja de enviar datos, o cuando debido al error humano, un empleado olvida registrar un dato. AlchemyML rellena esos valores faltantes, si se identifican unos patrones de comportamiento evidentes y la cantidad de

valores faltantes en la serie temporal es manejable. A continuación, puede ver dos ejemplos reales en los que se comparan las series con valores *missings* con las series después de haber sido tratadas por AlchemyML:

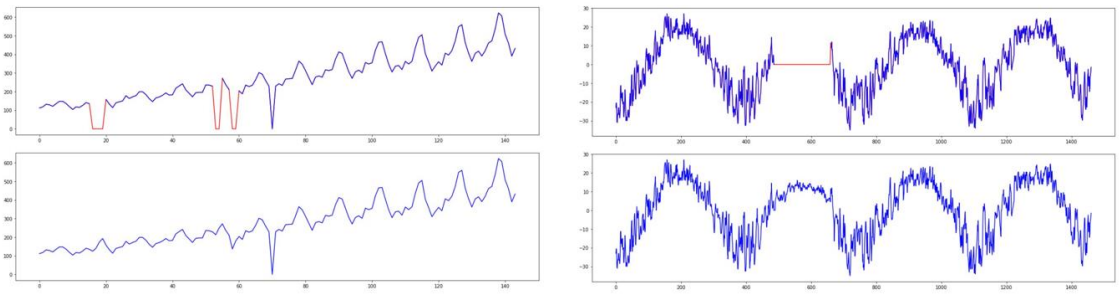


Figura 14 Dos ejemplos de relleno de valores missings en una serie temporal.

2. Predicciones

Si desea saber cómo puede poner sus modelos en producción o descargarse los modelos ya entrenados para utilizarlos a su antojo, contáctenos a través de hello@alchemyml.com o admin@alchemyml.com

3. Contacto

Esperamos que esta documentación haya sido útil para usted. Para cualquier duda o sugerencia puede escribirnos a hello@alchemyml.com o en admin@alchemyml.com.