mayor profundidad en la Unidad correspondiente a Análisis Exploratorio de los datos.

Etapa de construcción y evaluación de los modelos:

En esta etapa se construyen los modelos, definiendo los parámetros con los que se entrenarán y luego poniendo en marcha el proceso de entrenamiento. Luego que el proceso concluye, se evalúan, si al compararlos con determinadas pautas no nos satisface, escogemos nuevos modelos y repetimos. O variamos los parámetros del modelo y tratamos de sintonizar nuestro modelo.

Esta evaluación de los modelos se hace por medio de métricas que de alguna manera están ya estandarizadas y nos permiten comparar modelos y decidir.

Cuando tenemos un modelo que creemos que se aproxima a lo que esperamos podemos pasar a la siguiente fase.

Etapa de despliegue:

Esta etapa depende de las herramientas que estemos utilizando para nuestro proceso de aprendizaje. El hecho es que, en esta etapa, el modelo que ya está entrenado se despliega a producción para ser usado por otras aplicaciones o por nuestros clientes.

El proceso de despliegue puede que sea diferente en cada organización. En una el personal de IT solo debe escribir un par de letras y pasarla al equipo de gobierno de datos para que se apruebe, mientras que en otras a lo mejor hay que escribir un informe y la decisión demore.

Ya en producción los modelos están sometidos a una evaluación constante que determina cuando hay que eliminarlo o reentrenarlo.

Conclusiones

Hasta aquí hemos hecho un recorrido por los conceptos que consideramos que

© Universidad Internacional de La Rioja (UNIR)

deben formar parte de una introducción al aprendizaje automático, sin embargo, hay conceptos que aunque deben tocarse desde un primer momento son más complejos y requeriría de mucho más espacio para su explicación. Me refiero a conceptos como el compromiso Bias/Varianza, Overfiting/underfiting, etc.

Ejercicios:

- El curso favorito de los alumnos de una escuela.
- 2. Cantidad de libros en un anaquel.
- 3. Diámetro de una esfera.
- 4. Cantidad de clientes atendidos en un restaurante en un día.
- 5. Lugar que ocupa un nadador en una competencia.
- 6. Volumen de agua dentro de una lavadora de 200 litros de capacidad máxima.
- 7. Longitud de 150 tornillos producidos en una fábrica.
- 8. Número de pétalos que tiene una flor.
- 9. Color de cabello de los niños que audicionan para una película de Netflix.
- 10. Tiempo requerido para responder las llamadas en un call center.
- 11. Candidato al cuál apoyan los votantes en las elecciones presidenciales de Perú.
- 12. Número de televisores en una casa.
- 13. Número de páginas de una serie de libros de estadística.
- 14. Tiempo de vuelo de los aviones que van de Lima a Santiago.
- 15. Marcas de autos que se venden en tu país.
- 16. Grado de satisfacción laboral en una empresa.

© Universidad Internacional de La Rioja (UNIR)

- 17. Número de presidentes que ha tenido tu país en los últimos 5 años.
- 18. Peso de una persona.
- 19. Resultado de tirar dos dados.
- 20. Se define una variable como la fracción de focos defectuosos en una inspección de 100 focos escogidos aleatoriamente en el almacén de una fábrica. ¿Qué tipo de variable es?

Respuestas:

- 1. Variable cualitativa nominal.
- 2. Variable cuantitativa discreta.
- 3. Variable cuantitativa continua.
- 4. Variable cuantitativa discreta.
- 5. Variable cualitativa ordinal.
- 6. Variable cuantitativa continua.
- 7. Variable cuantitativa continua.
- 8. Variable cuantitativa discreta.
- 9. Variable cualitativa nominal.
- 10. Variable cuantitativa continua.
- 11. Variable cualitativa nominal.
- 12. Variable cuantitativa discreta.
- 13. Variable cuantitativa discreta.

© Universidad Internacional de La Rioja (UNIR)

- 14. Variable cuantitativa continua.
- 15. Variable cualitativa nominal.
- 16. Variable cualitativa ordinal.
- 17. Variable cuantitativa discreta.
- 18. Variable cuantitativa continua.
- 19. Variable cuantitativa discreta.
- 20. Variable cuantitativa discreta.

© Universidad Internacional de La Rioja (UNIR)

Tema 2. Análisis exploratorio de datos y preprocesamiento

2.1. Introducción y objetivos

En esta unidad formativa se trata toda la etapa de preprocesamiento y exploración de los datos con los que serán entrenados los modelos de aprendizaje.

La unidad estará dividida en dos lecciones: en la primera se trata todo lo relacionado con la preparación, limpieza y reducción de los datos, mientras que en la segunda se trata el manejo de datos con clases desbalanceadas, la división del conjunto de datos en datos en conjunto de entrenamiento y prueba y por ultimo el análisis exploratorio de los datos.

La idea detrás de esta unidad formativa es que se adquieran las habilidades necesarias en el manejo de los conjuntos de datos con los que se entrenan los modelos de aprendizaje automático, la corrección a los problemas que se pueden presentar y las técnicas básicas para ello.

El objetivo de esta lección es mostrar el conjunto de técnicas que se engloban dentro de lo que se conoce como etapa de preprocesamiento de los datos y el análisis exploratorio de datos.

El preprocesamiento está formado por las siguientes actividades: Integración, limpieza, normalización y transformación e identificación de ruido más aquellas que pertenecen a lo que se conoce como reducción de datos, como la selección de rasgos, la selección de casos y la discretización.

No siempre es necesario aplicar todas las técnicas a los datos, de lo que se trata es de ir adaptando los datos a los requerimientos de los algoritmos y modelos que necesitamos para resolver el problema que tenemos a mano.

© Universidad Internacional de La Rioja (UNIR)

Tema 2. Análisis exploratorio de datos y preprocesamiento

- Conocer las etapas de preprocesamiento de los datos.
- Conocer y aplicar las técnicas de manejo de casos y rasgos duplicados en esta etapa.
- Conocer y aplicar las técnicas de limpieza de datos, como el manejo de datos faltantes y anómalos.
- Conocer y aplicar las técnicas de normalización de datos.
- Conocer y aplicar las técnicas de selección de rasgos y casos.
- Conocer y aplicar las técnicas de discretización.
- Conocer y aplicar técnicas de manejo de datos desbalanceados.
- Conocer y aplicar las técnicas para la división del conjunto de datos en conjunto de entrenamiento y conjunto de pruebas.
- Conocer y aplicar las técnicas de análisis exploratorio de datos.

© Universidad Internacional de La Rioja (UNIR)

Tema 2. Análisis exploratorio de datos y preprocesamiento

2.2. Análisis exploratorio de datos y preprocesamiento I

De la lección anterior sabemos que el proceso de aprendizaje automático está dividido en las siguientes etapas:

- Establecimiento del problema.
- Recolección y limpieza de los datos (preprocesamiento).
- Construcción y evaluación de los modelos.
- Despliegue.

La segunda etapa, la de recolección y limpieza de datos, es lo que conocemos como etapa de preprocesamiento. La función de esta etapa es la de preparar los datos recolectados para que puedan suministrarse a los algoritmos de aprendizaje.

¿Por qué hay que preprocesar los datos?, esta pregunta aflora de forma natural cuando comenzamos el estudio del aprendizaje automático y nuestro objetivo es responder esta pregunta y mostrar el conjunto de técnicas que se aplican para aliviar cada uno de los problemas que se presentan.

Durante la recolección de los datos es normal que estos, estén contenidos en fuentes distintas, por ejemplo, en libros de Excel, tablas de bases de datos, ficheros planos, etc. Sería necesario entonces integrar todos estos datos en un solo conjunto de datos, esto puede traer consigo duplicados o que los datos que estemos integrando le falten valores o que contengan ruido y valores extremos. Si usáramos estos datos en la etapa de construcción de los modelos, los algoritmos podrías arrojar errores y cuando no lo hagan, de ninguna forma los resultados pudieran ser confiables.

Por tanto, la importancia de esta etapa es vital para el buen desempeño de los

© Universidad Internacional de La Rioja (UNIR)

modelos. Si los datos no son correctamente preparados, no se pueden asegurar buenos resultados en las prestaciones de los modelos.

Hay autores que sostienen que a esta etapa se le dedica más del 60% del tiempo de desarrollo de un modelo de aprendizaje automático.

Es importante entender desde el principio que esta etapa no es una etapa con reglas de actuación rígidas, es una etapa donde los problemas que tiene el conjunto de datos se van descubriendo poco a poco y se van solucionando de forma conjunta con el Análisis exploratorio de datos.

Por eso hemos combinamos en una sola lección lo correspondiente a estos dos procesos.

La etapa de procesamiento comprende dos conjuntos de técnicas dirigidas a objetivos diferentes, la primera, conocida como preparación de los datos y la segunda llamada como reducción de datos.

Entonces, la etapa de preprocesamiento de datos está formada por las siguientes tareas:

- Preparación de los datos.
- Reducción de datos.
- División del conjunto de datos. (Entrenamiento, validación y prueba)

La tarea de preparación de los datos, a su vez, está compuesta por las siguientes técnicas:

- Integración.
- Manejo de datos faltantes. (Missing Values)
- Detección de Datos Anómalos. (Outliers)

© Universidad Internacional de La Rioja (UNIR)

- Normalización

Por otra parte, la tarea de reducción de datos, está compuesta de las siguientes técnicas:

- Selección de rasgos (Variables).
- Selección de casos o instancias.
- Discretización.

En esta lección, por tanto, comenzaremos por el estudio de las técnicas de preparación de datos, luego las de reducción y por último, todo lo concerniente al análisis exploratorio.

¿Qué tipos de datos esperan cada uno de los métodos de aprendizaje automático? La respuesta a esta pregunta es importante por la simple razón que la mayor parte de esta etapa la dedicamos a adaptar los datos que recolectamos a los métodos o algoritmos que vamos a usar para resolver la tarea que nos planteamos.

Para tratar de responder esa pregunta hagamos un breve recorrido por los métodos de aprendizaje y su clasificación y luego veamos, aunque de forma muy somera, algunas de las exigencias que le impone cada algoritmo de aprendizaje a los datos.

Los algoritmos de aprendizaje automático pueden clasificarse en dos grandes grupos si nos basamos en su función:

- Predictivos. (aprendizaje supervisado)
- Descriptivos. (no supervisado)

A su vez, los predictivos se pueden dividir en:

- Estadísticos.

© Universidad Internacional de La Rioja (UNIR)

- Simbólicos.

Dentro de los métodos Predictivos y estadísticos tenemos métodos tales como:

- Regresión.
- Redes Neuronales.
- Aprendizaje Bayesiano.
- Aprendizaje basado en instancias. (K-NN)
- SVM. (Máquinas con vectores de apoyo)

Por otra parte, dentro de los métodos Predictivos y Simbólicos, están:

- Aprendizaje de Reglas.
- Árboles de decisión.

Métodos de regresión

Estos métodos son usados en tareas de estimación. Los métodos más conocidos son la Regresión Lineal, la Cuadrática y la logística. Aunque tienen muchos primos cercanos como Lasso y Mínimos cuadrados parciales (PLS), entre otros.

Este tipo de método requiere de atributos numéricos, es decir con rasgos o variables cuantitativas. No toleran muy bien los datos faltantes en los rasgos y son muy sensibles a datos anómalos. (Outliers)

Este tipo de método usa todos los rasgos presentes en el conjunto de datos sin hacer distinción si son útiles o no o si están correlacionados.

Redes neuronales

Al igual que los modelos de regresión, necesitan rasgos numéricos y no toleran

© Universidad Internacional de La Rioja (UNIR)

valores faltantes (Missing Values). Sin embargo, si son correctamente configuradas son relativamente tolerantes a datos anómalos (outliers) y ruido.

Aprendizaje bayesiano

Este tipo de método usa la Teoría de las Probabilidades como marco para la toma de decisiones racionales bajo incertidumbre. Se basa en el teorema de Bayes. El método más aplicado es el conocido como Naives Bayes. La definición inicial de este algoritmo solo trabaja con rasgos categóricos, debido a que el cálculo de la probabilidad solo puede hacerse sobre dominios discretos.

Este tipo de algoritmo es muy sensible a valores faltantes (Missing Values).

Debido a que asumen que los rasgos o atributos son independientes entre si, son muy sensibles a la redundancia tanto en los rasgos como en los casos, así como a los datos anómalos (outliers) o ruidosos.

Aprendizaje basado en instancias

Dentro de los métodos que se pueden incluir en esta clase el más conocido es K-NN. La diferencia entre estos métodos es la medida de distancia o similaridad que usan.

Uno de los principales problemas que sufre este tipo de método es la cantidad de espacio de almacenamiento que usa y una baja tolerancia al ruido. Esto hace que sea necesario tratar los datos por medio de procedimientos de reducción.

Máquinas con vectores de apoyo (Suppor Vector Machines)

En la literatura en español puede que la traducción sea Máquinas con vectores de soporte, sin embargo, para nosotros estos términos (Apoyo y soporte) son sinónimos y solo hace referencia a que usa algunos de los casos del conjunto de datos como apoyo o soporte en la construcción del hiperplano separante entre las clases. No importa que ahora no entiendan completamente esta idea, pero si debe quedar claro el hecho del uso de estas diferencias en la traducción del SVM se refieren a la misma

© Universidad Internacional de La Rioja (UNIR)

cosa.

© Universidad Internacional de La Rioja (UNIR)

Requiere datos con rasgos numéricos y sin valores faltantes y comúnmente son robustos antes datos anómalos y ruidos.

Aprendizaje Basado en Reglas

Los tipos de algoritmos que se clasifican bajo este nombre, por la naturaleza de su funcionamiento, normalmente necesitan rasgos o atributos nominales o discretizados. Los valores faltantes o anómalos, así como ruidosos pueden impactar de forma negativa la calidad del modelo final.

Ejemplo de algoritmos que se incluyen en esta categoría están: AQ, CN2, RIPPER, PART y FURIA.

Arboles de decisión

Estos algoritmos trabajan intentando separar los datos usando una de las variables independiente, dentro de subgrupos homogéneos. Estos están emparentados con los métodos de aprendizaje basado en reglas y sufren de las mismas dificultades que ellos. Usan rasgos nominales o discretizados. Los valores faltantes o anómalos así como ruidosos pueden impactar de forma negativa la calidad del modelo final.

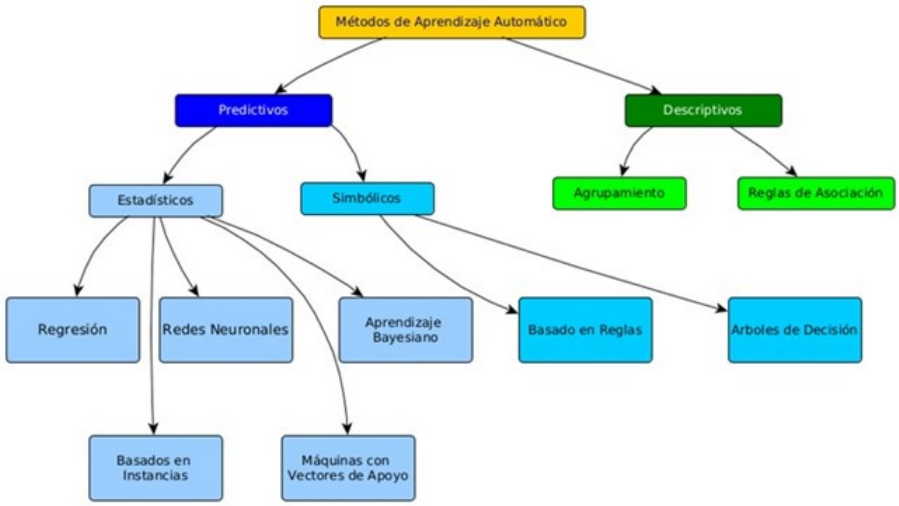

Figura 1. Clasificación de los métodos de aprendizaje automático.

© Universidad Internacional de La Rioja (UNIR)

Los métodos correspondientes a la clasificación Descriptiva, no los veremos en detalles, pues están de manera intima asociados a los predictivos en lo que respecta al uso de los datos.

Preparación de los datos

En esta sección analizaremos las técnicas usadas en la preparación de los datos antes de poder entrenar los modelos de Aprendizaje. Como ya mencionamos al principio de esta lección, la recolección de los datos es una etapa susceptible a errores que si no se tratan adecuadamente pueden poner en duda los resultados de los modelos de aprendizaje.

En la recolección de datos pueden darse problemas como la duplicación de casos y de rasgos. Pueden faltar valores a los rasgos de algunos de los casos o pueden ser anómalos, es decir que estén alejados de forma considerable de los valores considerados normales. Como los datos, puede que se integren desde medios diferentes, estos, pueden estar en unidades de medida diferente o los rasgos, aunque se refieran a los mismo, se llamen diferente. Etc.

Es en esta etapa donde se comienzan a lidiar con este tipo de problemas y las técnicas para su solución las iremos describiendo de manera más o menos en detalles.

Integración de datos

En esta etapa se trata de integrar en un solo conjunto, los datos que pueden provenir de diferentes fuentes. Estas fuentes pueden ser variadas, tales como tablas en bases de datos relacionales, documentos almacenados como libros de Excel, ficheros planos, etc.

Uno de los problemas que se da en este proceso de integración es la duplicación de casos o tuplas. (Una tupla es una fila en una tabla de una BD relacional) Por otro lado puede que también, por algún motivo, tengamos rasgos o atributos redundantes.

© Universidad Internacional de La Rioja (UNIR)

Estos problemas, pues, harán más demorado el proceso de Aprendizaje en el mejor de los casos y como norma evitará resultados confiables.

Técnicas para la detección de rasgos duplicados o redundantes.

El problema de la redundancia de los atributos puede hacer que el tamaño del conjunto de datos crezca, provocando que el tiempo de entrenamiento de un modelo también crezca y puede conducir a un sobre ajuste. (En la parte correspondiente a la división de un dataset antes de usarlo en el entrenamiento veremos que es el sobre ajuste).

Se dice que un rasgo o atributo es redundante cuando su información está contenida de alguna forma o puede ser derivada de otro atributo. La detección de este tipo de redundancia puede encontrarse por medio de análisis de correlación.

Aquí, lo más importante es entender que si dos rasgos están altamente correlacionados puede, de manera segura, eliminarse uno de los dos.

Hay varias maneras de acometer el análisis de correlación y en la práctica se lleva a cabo de formas diferentes también. Sin embargo, las técnicas básicas son las siguientes:

- Cuando los atributos o rasgos son nominales y los valores finitos se usa x2(Llamada Ji Cuadrado).
- Cuando los atributos son numéricos se usa los coeficientes de correlación y la covarianza.

No nos vamos a detener en el análisis de las formulaciones matemáticas de cada una de estas pruebas, pues existe muchísima documentación al respecto y librerías como pandas en python o R pueden ejecutarlas. Aquí nos centraremos en cómo aplicarlas y la interpretación de los resultados.

© Universidad Internacional de La Rioja (UNIR)

Veamos cómo se aplica la prueba x2:

Supongamos que tenemos un conjunto de datos sobre el que estamos trabajando y recién hemos integrado. Bueno, necesitamos comprobar si hay atributos redundantes. Además, sabemos que los atributos son nominales, tales como, género por ejemplo, que contiene valores del tipo masculino/femenino.

Tomemos dos rasgos y apliquemos la prueba ji cuadrado:

Como esta es una prueba de hipótesis, entonces la llamada hipótesis nula, es que las dos variables son independientes. Y podemos aceptarla o rechazarla en función del valor del valor p arrojado. (p-value)

Si p>alfa: Los rasgos son independientes. No están relacionados. Acepta la hipótesis nula.

Si p<= alfa : Los rasgos están correlacionados. Se rechaza la hipótesis nula.

A alfa se le conoce como nivel de significación estadística y un valor aceptado de forma general es de 0,05 o 5%.

Cuando los rasgos son numéricos, entonces usamos los coeficientes de correlación tales como los de Pearson o Spearman.

En el caso de usar el coeficiente de correlación de Pearson para determinar si dos rasgos son redundantes, pues tenemos que buscar en los valores que arroja este al usarlos entre dos rasgos. Aquí tampoco nos detendremos en la formulación matemática de este coeficiente y si en la interpretación de los resultados.

Si aplicar el coeficiente a un par de rasgos o atributos con el fin de determinar si son redundantes o no, la interpretación de los resultados sería:

Si r=1: Los dos rasgos están perfectamente correlacionados. Es decir, son redundantes. Esto implica que uno de los dos puede eliminarse.

© Universidad Internacional de La Rioja (UNIR)

Si 0 < r > 1: Existe una correlación positiva. Aquí puede tomarse como altamente correlacionados los mayores que 0,85 o 0,90. Es un criterio que depende de cada caso.

Si -1< r > 0: Entonces existe una correlación negativa.

Si r=-1: Existe una correlación negativa perfecta.

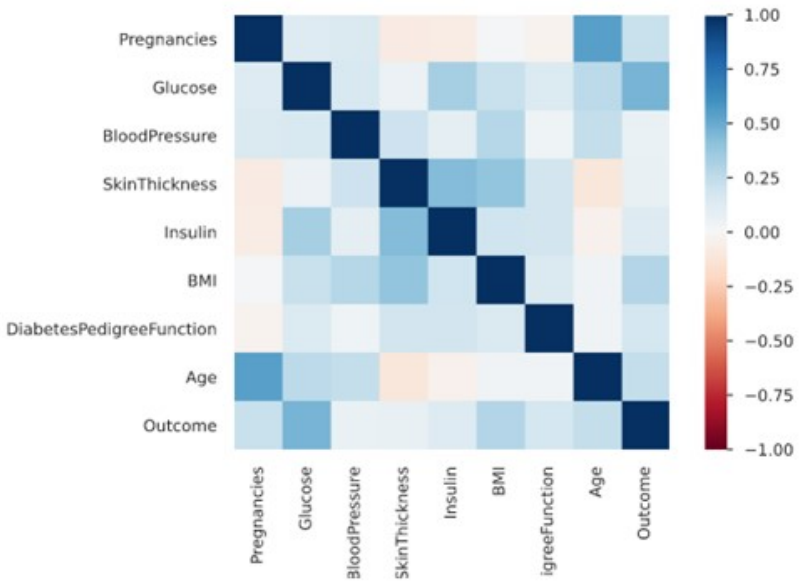

Figura 2. Ejemplo de una matriz de correlación usando coeficiente de correlación de Pearson.

En la parte correspondiente al análisis exploratorio de datos de esta lección daremos muchos más detalles de estos coeficientes.

Técnicas para la detección de casos duplicados o redundantes

Tener tuplas duplicadas o casos duplicados puede ser problemático, no solo por el hecho de desperdiciar espacio y tiempo en el entrenamiento de los modelo, sino porque puede ser una fuente importante de inconsistencia. Además es un problema formidable y difícil de atacar por la complejidad y sutileza de las formas en que puede

© Universidad Internacional de La Rioja (UNIR)

presentarse.

Estos problemas pueden presentarse, por ejemplo, a lo hora de integrar los datos, podrían darse el caso que con identificadores diferentes represente tuplas iguales. Este tipo de problema puede pasar inadvertido.

Otro de los problemas puede ser la aparición de tuplas o casos que aunque sean iguales los atributos estén, por ejemplo, representado en una con el Sistema Internacional de unidades y en la otra con Sistema métrico decimal.

Las técnicas para la detección de casos duplicados no están tan formalizadas como otras técnicas y los enfoques más usados son los siguientes:

- Enfoque probabilista: en este enfoque se formula el problema de detección de duplicados como uno de inferencia bayesiana.
- Enfoque aprendizaje supervisado: en este enfoque se han usado algoritmos tales como CART y SVM.
- Enfoque basado en distancia: en este enfoque se usan métricas de distancia o similaridad para la detección de duplicados.

La mayoría de las veces se usan técnicas a la medida cuando de filas duplicadas se trata. El proceso de integración puede ejecutarse de muchas maneras, pueden usarse herramientas propias de las bases de datos, herramientas de integración como el Servicio de Integración de SQL Server o Pentaho; pero también puede hacerse desde Python o R.

En cualquiera de los casos es muy conveniente que tenga un especial cuidado con el tema de los duplicados.

© Universidad Internacional de La Rioja (UNIR)

Figura 3. Integración de datos, según la ubicación.

En la Figura 3, puede verse que, durante la etapa de integración, los datos pueden estar físicamente en cualquier lugar. Pueden estar almacenados en Datalakes en la nube, en servidores de terceros a los cuales accedemos por medio de API REST u otros métodos, pueden estar en los Datawarehouses de las empresas o en redes locales y hasta en lo que se conoce como Silos de Información. Es decir, en los ordenadores de analistas en formas de ficheros, aunque esto es cada vez menos frecuente.

Los datos que necesitamos para el entrenamiento de nuestro modelo, pueden estar disponible, también, en diferentes tecnologías de almacenamiento, pueden estar en Bases de Datos relacionales como SQL Server, en BD de MySQL o en servidores Oracle. Pueden estar en Servidores noSQL como MongoDB u otro. Puede también, que estén en Bases de Datos de Grafos como Neo4J. Ver la Figura 4.

© Universidad Internacional de La Rioja (UNIR)

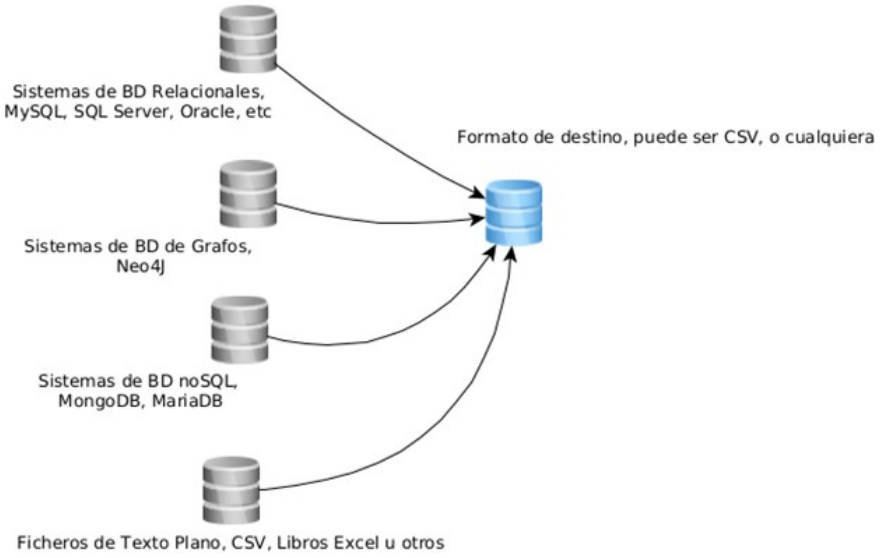

Figura 4. Integración de Datos desde diferentes tecnologías de almacenamiento.

El destino de la integración puede ser una tabla en una base de datos SQL o por lo común un fichero CSV, o de valores separados por coma. Sin embargo, luego de la integración no ha terminado todo, pasamos a una siguiente etapa, la de limpieza de los datos.

Limpieza de datos (Data Cleaning)

Después que se han integrado los datos en un dataset, no quiere decir que los datos estén correctos. Los datos pueden estar representados de manera no estándar, faltar datos o tener ruidos.

Todos estos problemas en los datos impactan de manera negativa en los resultados de los modelos de aprendizaje y por tanto hay que manejarlos de manera adecuada.

© Universidad Internacional de La Rioja (UNIR)

Manejo de valores faltantes (Missing Values)

Uno de los problemas que impacta de manera directa la calidad de los modelos es el llamado “Valores Faltantes” o Missing Values y puede darse de muchas formas. Para ejemplificar una de tantas formas en que puede mostrarse, supongamos que tenemos un dataset donde una de las variables es la tensión sanguínea, como en el caso del dataset Pima-indians-diabetes-database. En este dataset hay un rasgo llamado BloodPressure que tiene 35 ceros de los 768 casos con que cuenta. Está claro, de manera intuitiva, que la presión sanguínea no puede ser cero y por tanto se hace evidente que hay un problema a la hora de capturar los datos. Ahora bien, ¿cómo sustituimos estos ceros? ¿Qué valores les asignamos?

Para resolver el problema de los Missing Values o MV, se han desarrollado varias técnicas, llamadas técnicas de imputación.

La solución más rápida sería la de eliminar aquellos casos que contienen atributos con MV. Sin embargo, corremos el riesgo de provocar sesgo en los modelos al perder información valiosa contenida en estos casos. Esto es aún peor si el conjunto de datos tiene relativamente pocos casos.

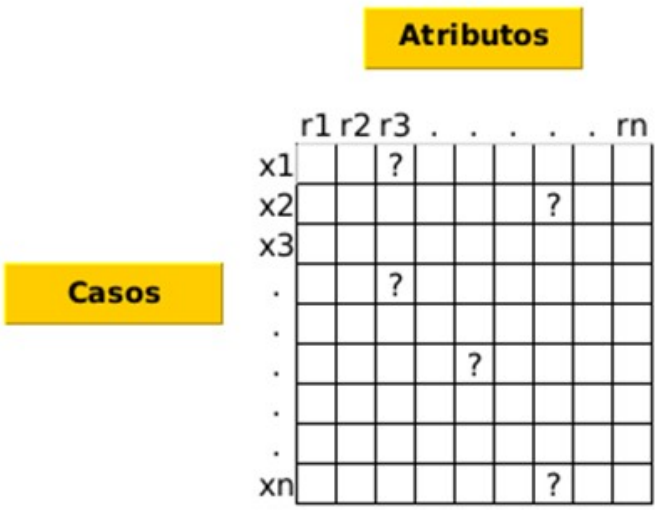

Figura 5. Representación general de un dataset con MV.

© Universidad Internacional de La Rioja (UNIR)

Dentro de las técnicas más sencillas para el manejo de MV, están las de sustituir sus valores con otros del mismo rasgo. A continuación, las mencionaremos.

Sustitución por valores del mismo rasgo:

- Sustituir MV, con la media o la mediana del rasgo si es cuantitativo.
- Sustituir MV, con la moda del rasgo si es categórico.

Si el conjunto tiene varias clases, los MV puede sustituirse con la media o la moda de la clase.

Otros métodos más elaborados de imputación de MV son los siguientes:

- Métodos basados en Máxima Verosimilitud. (Maximum Likelihood Imputation Methods).
- Métodos basados en Maximización de la Esperanza. (Expectation-Maximization).
- Imputación Múltiple.
- Métodos de imputación basados en Análisis de componentes principales Bayesiano. (Bayesian Principal Component Analysis (BPCA)).
- Imputación con KNN. (K vecinos más cercanos (KNNI)).
- Imputación con KNN Ponderado. (Weighted K-NN (WKNNI)).
- Imputación por medio de clustering usando k-mean. (K-means Clustering Imputation (KMI)).
- Imputación usando Fuzzy K-mean. (Imputation with Fuzzy K-means Clustering (FKMI)).
- Imputación usando SVM. (Support Vector Machines Imputation (SVMI)).

© Universidad Internacional de La Rioja (UNIR)

- Imputación usando Descomposición de Valores Singulares. (Singular Value Decomposition Imputation (SVDI)).
- Imputación por medio de Mínimos Cuadrados. (Local Least Squares Imputation (LLSI)).

Aunque, como puede verse, de la lista anterior hay varios métodos de imputación, sin embargo, ninguno es universal o mejor que otro.

Desde una perspectiva práctica podemos explorar los métodos de imputación disponibles en el módulo scikit-learn de python.

En python los valores faltantes o MV suelen representarse por medio de NaN, que significa “No es un Número”.

Este módulo contiene un conjunto de clases que cubren dos grupos de métodos de imputación de MV. Un grupo llamado métodos univariados y el otros, métodos multivariados, además de varios métodos basados aprendizaje automático.

Los métodos univariados son aquellos que usan el valor del rasgo con MV para sustituir los valores faltantes. En contraparte, los multivariados usan el dataset completo.

Estos métodos están disponibles en sklearn.impute y los métodos univariados se pueden usar por medio de la clase SimpleImputer, mientras que los multivariados, por ejemplo, podrían usar IterativeImputer.

El listado que mostramos a continuación es un ejemplo de uso de un método invariado.

© Universidad Internacional de La Rioja (UNIR)

```
In [1]: import numpy as np
        from sklearn.impute import SimpleImputer
        #*************************************************************
        imp = SimpleImputer(missing_values=np.nan, strategy='mean')
        imp.fit([[1, 2], [np.nan, 3], [7, 6]])
        X = [[np.nan, 2], [6, np.nan], [7, 6]]
        print(imp.transform(X))

        [[4.         2.        ]
         [6.         3.66666667]
         [7.         6.        ]]
```

Figura 6. Método invariado. Fuente: elaboración propia.

Otra de las formas en que se puede acometer esta sustitución de MV, es usando la librería Pandas.

La función fillna(), ofrece varios métodos para hacerlo. Si se le pasa como argumento un cero, por ejemplo, este sustituirá todos los nan con 0. Si se le pasa como argumento method=”bfill” sustituye el nan con el valor del mismo atributo del próximo caso. Si se le pasa el argumento method=”ffill” entonces lo hará en el valor del caso anterior. Ver Figura 7.

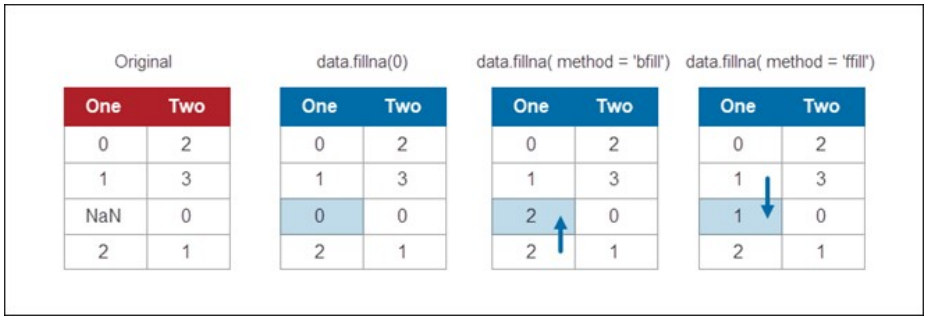

Figura 7. Funcionamiento del método fillna() de Pandas para el tratamiento de MV.

Una vez que tengamos resuelto el problema de los MV, es necesario chequear otras posibles anomalías en nuestro Dataset.

© Universidad Internacional de La Rioja (UNIR)

Manejo de datos anómalos (Outliers)

Al igual que el problema de los valores faltantes o Missing Values, el problema de datos anómalos, extremos o outliers es muy frecuente. Se les llama outliers a aquellas observaciones que se desvían en alguna dirección respecto al comportamiento general del resto de los valores.

Este tipo de valores atípicos pueden tener un efecto adverso en algunos de los métodos de aprendizaje y por tanto es importante tratarlos de manera adecuada.

Una de las definiciones más usadas en la literatura es la dada por Hawkins(1980):

“Un outlier es una observación que se desvía tanto del resto de las observaciones que hace sospechar que fue generada por otro mecanismo”

La intuición detrás de esto es que todos los datos son generados por determinado mecanismo, por ejemplo, algún proceso estadístico. Los datos considerados normales tienen propiedades comunes y si aparece alguno que se desvía demasiado de estas, puede hacer que se sospeche que su mecanismo de generación es otro.

Para entender esto de los mecanismos de generación pongamos como ejemplo el dado por Barnett.

En este ejemplo, el hijo del señor Mr. Hadlum nació 349 días después de que él salir del servicio militar, si como promedio el periodo de gestación humano es de 280 días (40 semanas), entonces este periodo de 349 días puede considerarse estadísticamente un outlier.

Los outliers pueden catalogarse en univariados o multivariados en dependencia de si tomamos en cuenta solo los valores de un rasgo o variable o si se tienen en cuenta todos los rasgos de una instancia.

En el caso de la detección de outliers univariados, la técnica más sencilla es la de

© Universidad Internacional de La Rioja (UNIR)

tratar como anómalos todos aquellos valores que están por encima o por debajo de 1,5 veces el rango intercuartílico.

El rango intercuartílico es la diferencia entre el tercer y primer cuartil.

$$IQR = Q3 - Q1$$

Por tanto, podemos tratar como anómalos leves a aquellos valores que estén por debajo de 1,5 veces el rango intercuartílico o por encima de ellos.

$$x < Q1 - 1{,}5 * IQR obien x > Q3 + 1{,}5 * IQR$$

También se le llama valores anómalos severos a aquellos que están hasta 3 veces por debajo o por encima de IQR.

La manera más sencilla de aplicar esta técnica es usando los llamados gráficos de caja y bigote. (Boxplot) Estos gráficos muestran de manera sencilla los valores anómalos, ubicándolos a la derecha o encima del fin del bigote o por debajo o a la izquierda del inicio de él.

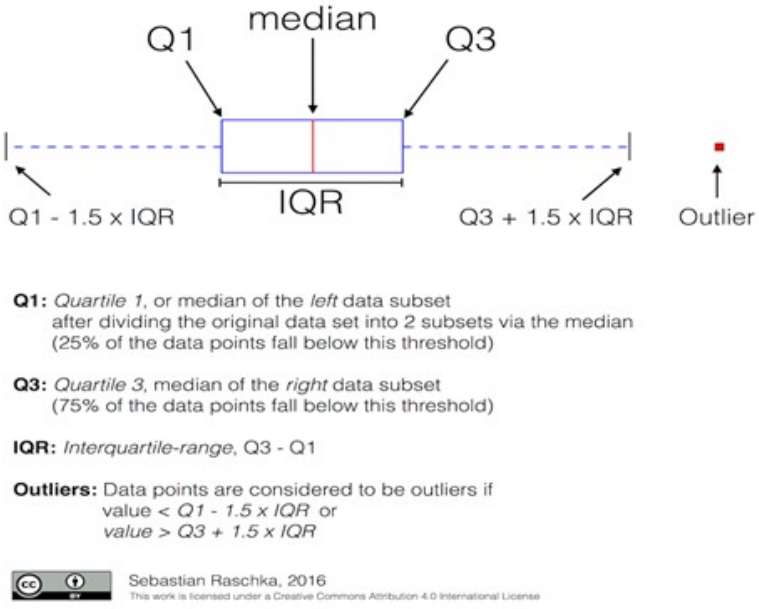

Figura 8. Gráfico de caja y bigote. (Boxplot).

© Universidad Internacional de La Rioja (UNIR)

Una vez que detectamos los outliers, estamos en condiciones de determinar cómo debemos tratarlos. La vía más fácil y la del primer impulso sería eliminarlos. Si contamos con muchos datos y los anómalos o extremos son pocos a lo mejor sería buena idea hacerlo; pero sería prudente preguntarse si estos valores se deben a errores en el proceso de integración o en la recolección de los datos o son una muestra de valores genuinos. En cualquiera de las situaciones es necesario preguntarse el origen y tomar la vía de tratamiento en función de eso.

Lo cierto es que, cuando tratamos con valores extremos univariados, podemos aplicar algunas variantes, como sustituirlos por la media de los valores considerados no extremos o con la mediana del conjunto. Otras serían, sustituirlos por los valores que lindan con los bordes de los bigotes en el gráfico de caja y bigote. En cualquiera de los casos el método de tratamiento de estos valores depende del problema que tengamos a mano.

Es importante no perder de vista que las técnicas de detección de datos extremos o anómalos pueden usarse en dos contextos diferentes, uno en el proceso de limpieza de datos y el otro en la detección de eventos anómalos. Y lo llamamos así para diferenciarlos de alguna manera. En el primero, el objetivo es evitar que estos datos influyan de manera negativa en el resultado del entrenamiento de los métodos de aprendizaje automático. Mientras en el segundo, su objetivo es detectar eventos que se desvían de lo normal, como, por ejemplo, fraudes en tarjetas de crédito.

Para la detección de outliers multivariados se han desarrollado varios métodos; pero de manera general todos caen en alguna de las siguientes categorías:

- Métodos basados en pruebas estadísticas.
- Métodos basados en profundidad.
- Métodos basados en desviación.

© Universidad Internacional de La Rioja (UNIR)

- Métodos basados en distancia.
- Métodos basados en densidad.
- Métodos de alta dimensionalidad.

No describiremos ninguno de estos métodos aquí; pero es importante que se tengan en cuenta y se conozcan. Basta luego, si necesitamos usarlas, con buscar en la documentación de las herramientas si están implementadas o buscar documentación adicional para hacerlo. La librería PyOD implementa más de 30 algoritmos de detección.

Veamos un ejemplo en python de cómo se usa pyod.

```
import pandas as pd
from pyod.models.abod import ABOD
from pyod.utils.data import evaluate_print
from pyod.utils.example import visualize
from sklearn.model_selection import train_test_split
from pyod.utils.data import generate_data
#*************************************
contamination = 0.1 # percentage of outliers
n_train = 500       # number of training points
n_test = 500        # number of testing points
n_features = 2      # number of features

X_train, X_test, y_train, y_test = generate_data(n_train=n_train, n_test=n_test, n_features= n_features,
                                                 contamination=contamination,behaviour="new")
#*************************************
OD_Name = "ABOD"
od=ABOD(method="fast")
od.fit(X_train)
#*************************************
y_train_pred = od.labels_  # binary labels (0: inliers, 1: outliers)
y_train_scores = od.decision_scores_  # raw outlier scores
y_test_pred = od.predict(X_test)  # outlier labels (0 or 1)
y_test_scores = od.decision_function(X_test)  # outlier scores
visualize(OD_Name, X_train, y_train, X_test, y_test, y_train_pred,y_test_pred, show_figure=True, save_figure=False)
```

Figura 9. Código Python para un detector de Outlier usando ABOD. (Angle Based Outlier Detection).

Fuente: elaboración propia.

© Universidad Internacional de La Rioja (UNIR)

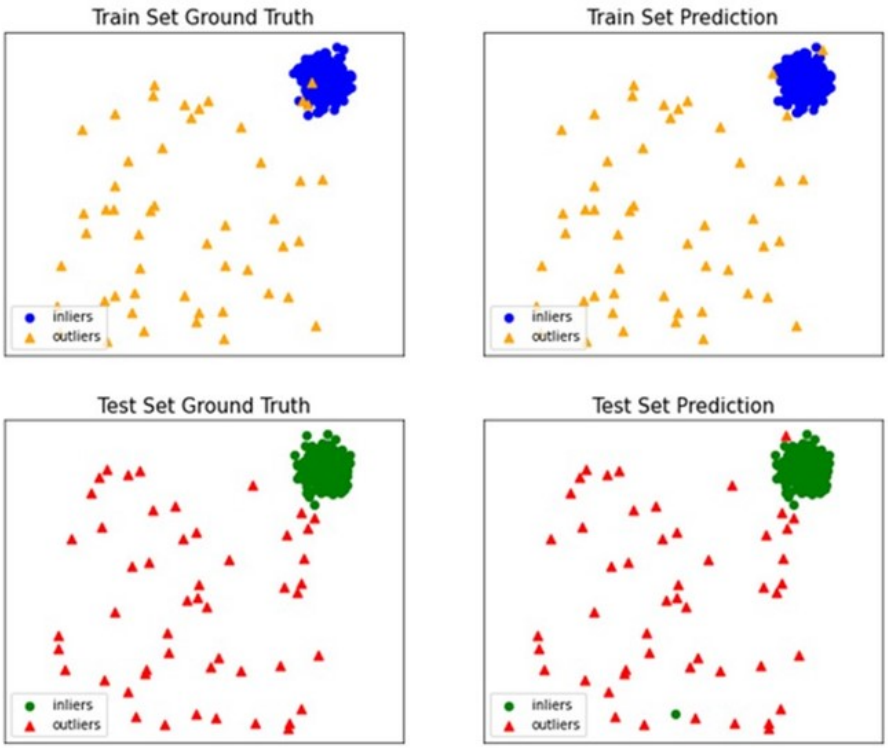

Figura 10. Resultados de la corrida del detector de outliers.

Normalización

Hay varios algoritmos de aprendizaje que son especialmente susceptibles a las escalas de los valores de los rasgos, por ejemplo, las redes neuronales y los algoritmos de regresión o los métodos basados en medidas de distancia. La normalización, representa todos los atributos de un dataset en una escala común, evitando que determinados valores influyan más que otros en los resultados de los modelos.

El objetivo de la normalización es cambiar los valores de los atributos numéricos en el dataset a una escala común, sin distorsionar las diferencias en los rangos de valores. Para el aprendizaje automático, todos los conjuntos de datos no requieren normalización.

© Universidad Internacional de La Rioja (UNIR)

Existen varias formas de normalizar, listemos dos de ellas:

- Normalización min-max.
- Normalización z-Score.

Normalización min-max:

El objetivo de la normalización min-max es escalar todos los valores numéricos x de un rasgo o atributo dado a un rango especificado denotado por [new-minA, new-maxA].

Esto se hace aplicando la siguiente expresión a x para obtener x[1]:

$$x' = (x - minx) / (xmax - xmin)$$

Donde minA es el valor mínimo del atributo A, maxA es el valor máximo del atributo A.

En la literatura "normalización" se refiere a un tipo particular de normalización min-max en la cual el intervalo final es [0,1], aunque también es típico usar un intervalo de [-1, 1].

Normalización z-Score:

Otro de los métodos de normalización usados comúnmente en el z-score. Este puede obtenerse aplicando a cada valor del atributo la siguiente expresión:

$$x' = (x - {}^{-}X) / \sigma X$$

donde $\overline{X}$ es la media del atributo y σX es la desviación estándar.

Una vez más, con el fin de ver en la práctica como se usan estas técnicas apelamos al módulo scikit-learn de Python.

Este contiene una clase llamada MinMaxScaler que permite normalizar por el método

© Universidad Internacional de La Rioja (UNIR)