OBS Business School | THREEPOINTS

De:
Planeta Formación y Universidades

Alfonso Ibáñez Martín

Modelos de Machine Learning: Optimización y Aplicaciones

Contenido Teórico

Índice

TEMA: Fortalezas y Debilidades de los algoritmos

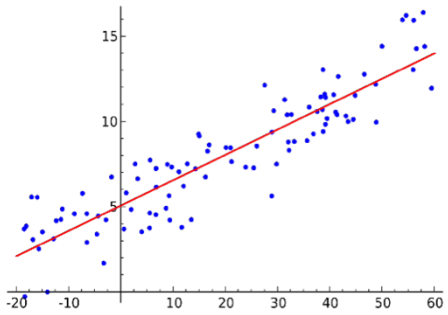
Durante este tema se analizan las ventajas e inconvenientes de los principales algoritmos, tanto de aprendizaje supervisado como de aprendizaje no supervisado. Además, se detallan los algoritmos referidos a la selección de variables y a la extracción de variables.

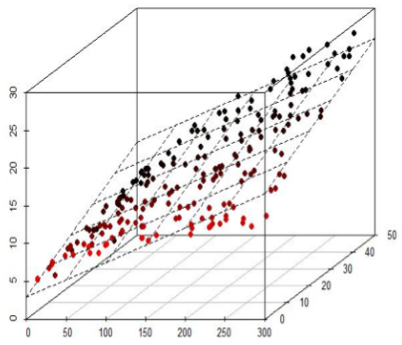
En los problemas de aprendizaje supervisado se genera un modelo a partir de un conjunto de datos que ya está etiquetado con la respuesta correcta. A partir de la experiencia adquirida en la fase de entrenamiento, el modelo es capaz de realizar una predicción sobre un nuevo conjunto de datos. En cambio, los modelos de aprendizaje no supervisado se generan usando un conjunto de datos que no tiene ninguna etiqueta, es decir, nunca se le dice al algoritmo lo que representan los datos. La idea es que el algoritmo pueda encontrar, por si solo, patrones que ayuden a entender el conjunto de datos. Finalmente, la aplicación de técnicas de reducción de la dimensionalidad tiene como objetivo sustituir el conjunto de variables explicativas por otro de menor dimensión, que incluya las variables más relevantes del problema.

1.1. Algoritmos de aprendizaje supervisado

Las principales tareas que resuelven las técnicas de aprendizaje supervisado son los problemas de clasificación y de regresión. A partir de un conjunto de variables predictoras, un modelo de clasificación trata de descubrir los patrones existentes entre las variables y así, predecir la variable categórica objetivo. En cambio, un modelo de regresión trata de identificar las relaciones existentes en los datos y realizar predicciones sobre la variable continua, objetivo del problema. A continuación, se comentan las ventajas e inconvenientes de los algoritmos más utilizados para resolver dichos problemas.

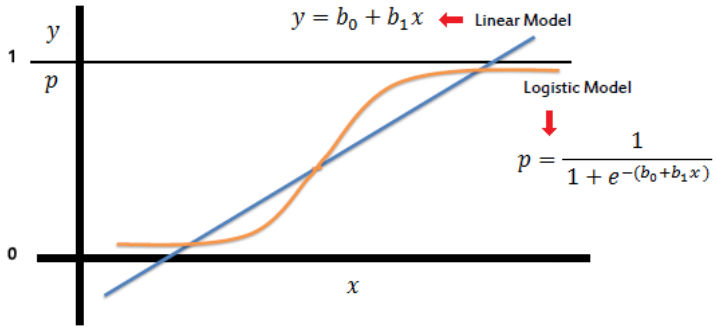
- **Linear Regression:** Esta técnica genera un modelo de regresión que permite predecir el valor de una variable continua a través de una función lineal de las variables predictoras. La regresión lineal genera buenos resultados cuando existen relaciones lineales entre las variables independientes y la variable dependiente del conjunto de datos analizado. Los siguientes gráficos muestran el concepto de la regresión lineal cuando solo existe una variable predictora (izquierda) y cuando existen varias variables predictoras (derecha). En el primer caso, el modelo está formado por una recta, mientras que, en el segundo caso, el modelo está formado por un hiperplano.





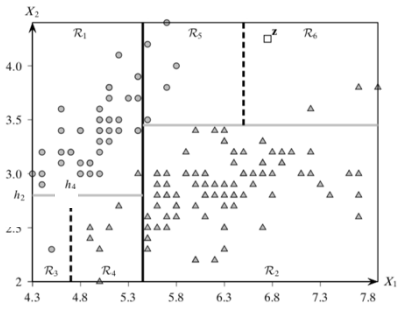
En ambos casos la principal ventaja que tienen estos algoritmos es que son muy interpretables, haciendo fácil la explicación del modelo y de los resultados a perfiles no técnicos. Además, el entrenamiento de este modelo es computacionalmente muy eficiente y la realización de las predicciones son de baja latencia. Esto conlleva a que sea una de las técnicas más utilizadas para tomar decisiones de negocio. Sin embargo, estas técnicas son muy rígidas y no son capaces de capturar patrones complejos en los datos, produciendo malos resultados cuando existen relaciones no lineales entre las variables analizadas. De manera adicional, la presencia de valores atípicos en los datos, la ausencia de distribuciones normales y la colinealidad entre las variables analizadas afectan negativamente a los resultados del modelo.

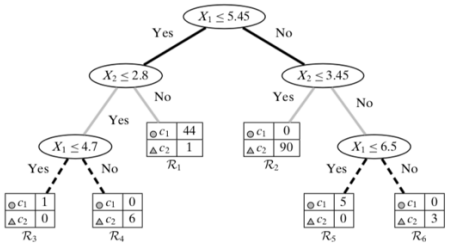
- **Logistic Regression:** Este algoritmo se basa en aplicar una función logística a una combinación lineal de variables independientes para generar un modelo de clasificación. Una de sus principales características es que permite interpretar los resultados como la probabilidad de que un evento ocurra. Además, permite conocer qué factores son los que más influyen en el evento analizado, no asume una relación lineal entre la variable dependiente y las variables independientes, y tiene la flexibilidad de que las variables predictoras pueden tener distintas tipologías de distribuciones. Las siguientes imágenes muestran visualmente las principales diferencias entre la regresión lineal y la regresión logística.



En cambio, estas técnicas necesitan muchos más datos que otras técnicas de regresión para conseguir unos resultados más estables y significativos. Además, a pesar de ser un modelo no lineal, hay situaciones en las que la regresión logística no consigue buenos resultados, ya que no es flexible para capturar relaciones muy complejas entre los datos. Finalmente, estas técnicas también se ven afectadas ante la presencial de valores outliers y missing.

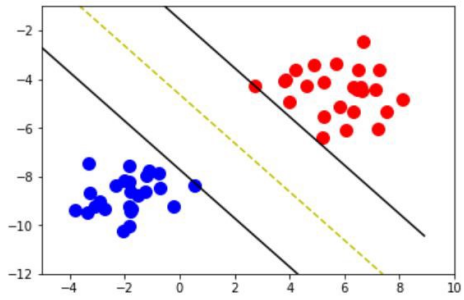
- **Decision Tree:** Los árboles de clasificación (cuando la variable objetivo es categórica) y los árboles de regresión (cuando la variable objetivo es continua) son técnicas que permiten predecir la asignación de muestras a grupos predefinidos en función de un conjunto de reglas. La idea subyacente de estos algoritmos consiste en segmentar el espacio de variables predictoras en varias regiones, intentando lograr que se maximice la ganancia de información sobre la variable dependiente. Las siguientes imágenes muestran como se generan las distintas regiones de una variable categórica en base al conjunto de variables predictoras.





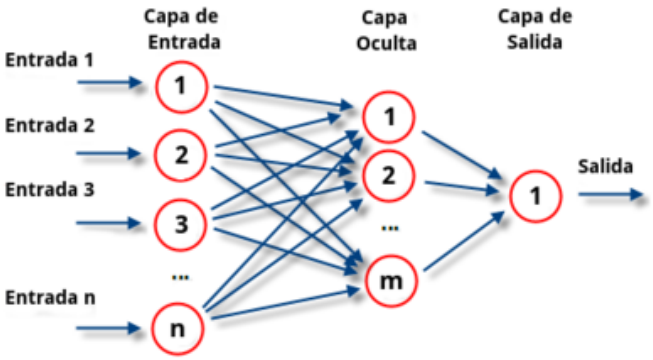
Una de las principales ventajas de estos algoritmos es que son sencillos, fácilmente interpretables y capturan mejor las relaciones no lineales debido a su estructura jerárquica basada en reglas. Otra ventaja que tienen estos modelos es que son robustos a valores vacíos, valores extremos, permiten la identificación de variables relevantes y las tareas de normalización no son necesarias sobre las variables. Sin embargo, tareas como la poda son requeridas, ya que estos algoritmos son propensos al sobreajuste de los datos debido a un elevado número de decisiones. Otro inconveniente asociado a los árboles de regresión es que la variable respuesta puede llegar a tener, como máximo, el mismo número de valores distintos como nodos terminales tenga el modelo.

- **Support Vector Machine:** Este algoritmo de aprendizaje es utilizado para resolver, principalmente, problemas de clasificación. El objetivo de este algoritmo es encontrar, a través de una función kernel, un hiperplano que sea capaz de separar el conjunto de datos en diferentes clases. Debido a la existencia de muchos hiperplanos candidatos que satisfagan la condición anterior, este algoritmo selecciona el hiperplano que maximiza la distancia entre las distintas clases. El siguiente gráfico representa la idea subyacente de dicho modelo.



En los casos en los que estos algoritmos utilizan una función kernel lineal, el modelo obtenido es muy similar al concepto de la regresión logística, por lo tanto, la gran ventaja de estas técnicas es utilizar una función kernel no lineal que sea capaz de modelar límites de decisión complejos. En este contexto, estos modelos son capaces de solventar cualquier problema con una función kernel apropiada. Además, los resultados de estos modelos son robustos al sobreajuste de los datos, a los óptimos locales y a la aparición de valores extremos, entre otros. Finalmente, estas técnicas también pueden ser utilizadas para abordar problemas de regresión. Por otro lado, estos modelos son computacionalmente muy intensos, ya que necesita mucho tiempo para el entrenamiento de grandes volúmenes de datos. Además, la elección de un buen kernel es muy complicado y necesita de muchas iteraciones. Estos modelos tienen también la limitación de que son difíciles de interpretar por lo que difícilmente se puede incorporar a la lógica del negocio.

- **Neural Network:** Las redes neuronales artificiales son métodos utilizados para resolver tanto problemas de clasificación como de regresión. Estas técnicas modelan la relación existente entre un conjunto de variables input y la variable output inspirándose en el funcionamiento de las redes neuronales biológicas. Las estructuras de las redes neuronales artificiales pueden llegar a ser muy complejas dependiendo del número de capas de neuronas ocultas involucradas en el modelo. En estas capas ocultas se puede modelar relaciones muy complejas que otros algoritmos no pueden detectar fácilmente.



Los modelos de redes neuronales con múltiples capas son técnicas muy novedosas que obtienen buenos resultados cuando se utilizan para clasificar audios, textos e imágenes. Estas técnicas son muy efectivas para modelar relaciones no lineales, y, además, mejoran su performance inicial a medida que son entrenados con un volumen mayor de información. El principal inconveniente de estos modelos es que no pueden ser interpretados con facilidad debido a su alta complejidad interna. Además, son técnicas muy costosas computacionalmente y requieren la optimización de los parámetros del algoritmo. En general, estas técnicas no pueden considerarse como un algoritmo de propósito general ya que tienen algunas restricciones importantes, y, a menudo, su performance es inferior al de otras técnicas cuando no se dispone de una elevada cantidad de datos.

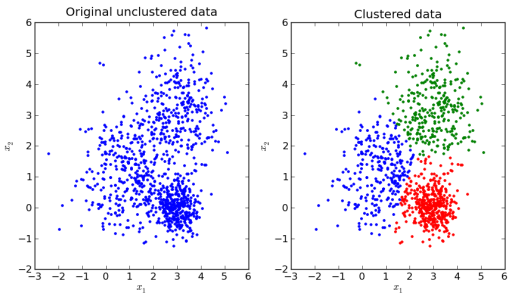
1.2. Aprendizaje no supervisado

El principal problema que resuelven las técnicas de aprendizaje no supervisado es el de encontrar la estructura intrínseca existente dentro de una colección de datos sin etiquetar y caracterizados por un conjunto de variables. Este problema se denomina clustering y tiene como objetivo agrupar los elementos de una colección, de tal forma que, los elementos que pertenecen a un mismo grupo sean muy similares entre sí, y los grupos resultantes muy heterogéneos entre sí.

Aunque existen muchas tipologías de algoritmos para solucionar el problema del clustering, los enfoques más comunes son: partitional clustering, hierarchical clustering y density-based clustering. Los algoritmos particionales tienen como objetivo dividir el conjunto de datos en varios grupos, tratando de optimizar una función de distancia. En cambio, los algoritmos jerárquicos realizan una descomposición de los datos a distintos niveles jerárquicos en base a la métrica objetivo. Finalmente, los algoritmos basados en densidad no tienen en cuenta la proximidad entre los datos, y agrupan los datos según un criterio de densidad. Otros enfoques para abordar los problemas de clustering son: algoritmos borrosos, algoritmos basado en modelos, algoritmos de clustering espacial, etc.

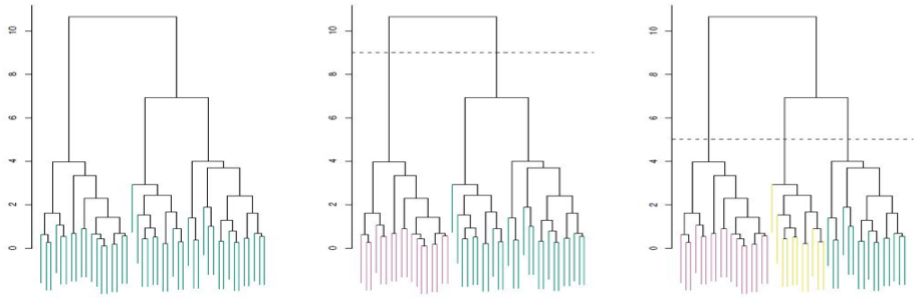
En esta sección se presentan varios de los algoritmos más utilizados para resolver los enfoques anteriormente mencionados. Además de una pequeña descripción de cada uno de ellos, se comentan las principales ventajas e inconvenientes de los mismos.

- **Partitional clustering:** El objetivo de esta familia de algoritmos es identificar, como clusters, áreas muy pobladas de datos dentro del espacio de búsqueda. Una de las técnicas más utilizadas de esta familia es el algoritmo k-means, que consiste en definir un punto central de referencia de cada cluster (denominado centroide) y asignar a cada individuo al cluster del centroide más próximo en función de las distancias existentes entre los atributos de entrada. El algoritmo parte de la elección de k centroides aleatoriamente y, mediante un proceso iterativo, se asigna cada punto al cluster con el centroide más próximo, procediendo a actualizar el valor de los centroides. Este proceso termina cuando se alcanza un criterio de convergencia. Otros algoritmos particionales son PAM y CLARA.



Este es el método de clustering más popular porque es simple, fácil de implementar y rápido respecto otros métodos, incluso con un elevado número de variables. Además, este algoritmo genera muy buenos resultados cuando el número de clústers es elevado o cuando los datos tienen una estructura esférica o elíptica. A pesar de las ventajas mencionadas anteriormente, este algoritmo tiene algunos inconvenientes. En este contexto, los valores de los parámetros (número de clusters y centroides iniciales) afectan en gran medida al resultado final del modelo. Además, este algoritmo no es capaz de identificar elementos de ruido y outliers en el conjunto de datos. Finalmente, es muy sensible a la escala de valores, por lo que una normalización o estandarización de los valores en las variables originales modifica sustancialmente los resultados del modelo.

- **Hierarchical clustering:** Este paradigma es una alternativa al clásico enfoque particional al que pertenece el algoritmo k-means. El objetivo de los algoritmos jerárquicos es dividir el conjunto de datos por niveles y, en cada nivel, unir o dividir dos grupos del nivel anterior, según si es un algoritmo aglomerativo o divisivo. Dicho paradigma genera una representación (dendograma) de la agrupación final de los datos en la que se observan los distintos clusters generados en función del número de cortes seleccionado. Entre los algoritmos jerárquicos más representativos están BIRCH, CURE y CHAMELEON.

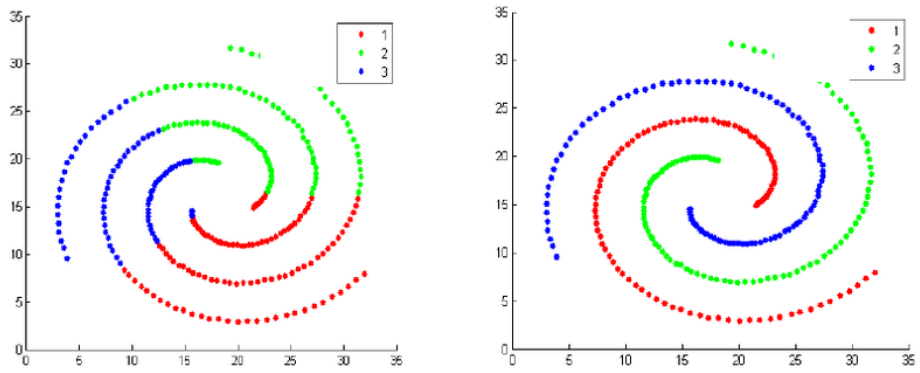


Una de las principales ventajas del clustering jerárquico, a diferencia del k-means, es que genera buenos resultados cuando los datos no siguen una estructura globular. Además, los algoritmos utilizados son menos sensibles al ruido de los datos y el dendograma generado permite un análisis a distinta granularidad y facilita mucho la decisión sobre el número de clústers final.

Sin embargo, aunque el número de clústers no sea un requisito de entrada al algoritmo, el usuario debe seleccionar dicho número tras la finalización del proceso para conseguir un resultado final. Otra limitación del algoritmo es que no hay posibilidad de deshacer las iteraciones iniciales o previas, es decir, no se puede reasignar un determinado elemento que ya haya sido asignado a un cluster previamente. Además, al igual que otros algoritmos basados en distancias, los resultados son sensibles a la métrica utilizada, no escala con facilidad para grandes volúmenes de datos y requiere de mucho tiempo de computación.

- **Density-based clustering:** Esta tipología de algoritmos tiene como objetivo identificar regiones de alta densidad que estén rodeadas de áreas poco densas. En este contexto, cada una de las regiones densas identificadas se corresponde con un cluster. Algunos ejemplos de algoritmos que se basan en este enfoque son: DBSCAN, OPTICS y DENCLUE, entre otros.

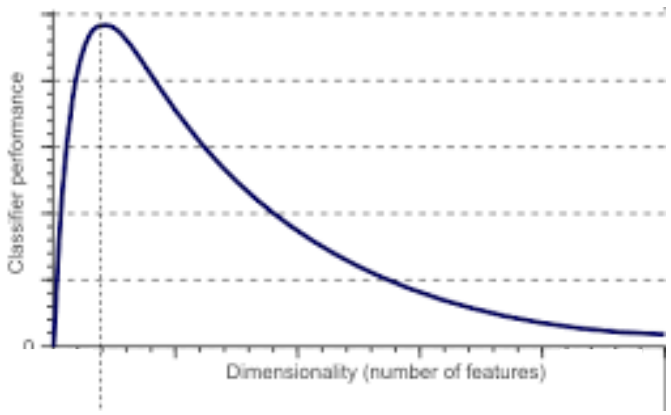
  Uno de los principales representantes de este paradigma es el algoritmo DBSCAN. Este método es apropiado para detectar clusters que contienen ruido, valores extremos o una forma geométrica indefinida. A diferencia del algoritmo k-means, este método no requiere de la especificación del número final de clusters, ni asume distribuciones globulares. La siguiente imagen muestra las diferencias entre los clústers generados por el algoritmo k-means (izquierda) y el algoritmo DBSCAN (derecha).



A pesar de sus bondades, el algoritmo DBSCAN tiene dos parámetros que se deben fijar y optimizar. Los valores óptimos de dichos parámetros no son fáciles de encontrar y cualquier variación sobre ellos influye en gran medida en el resultado final del clustering. Además, este algoritmo no obtiene buenos resultados cuando el conjunto de datos es de muy alta dimensionalidad, ya que los datos están muy dispersos y no se generan regiones de alta densidad. Finalmente, este algoritmo es no determinista por lo que los resultados pueden variar en varias ejecuciones del modelo.

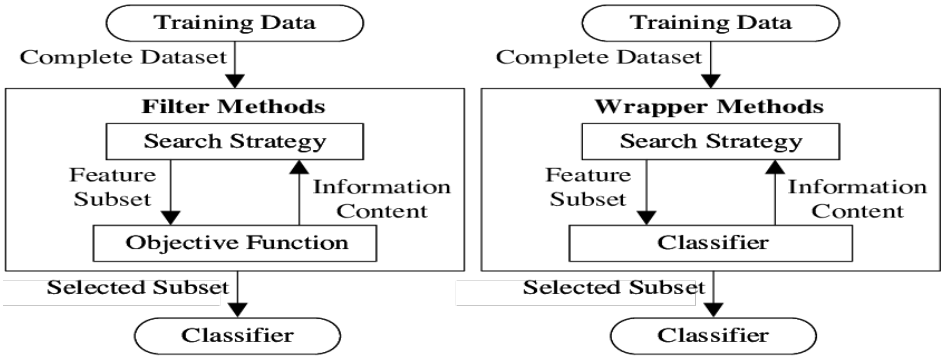
1.3. Reducción de la dimensionalidad

En ocasiones un alto número de variables en problemas de aprendizaje automático puede resultar negativo a la hora de modelar la solución. Algunos de los problemas más comunes que pueden ocurrir son: el sobreajuste de los modelos, la dificultad en la interpretación de los resultados, la existencia de colinealidad entre variables y la curse of dimensionality que refleja el problema de la disminución del performace del modelo a medida que aumenta el número de variables una vez alcanzado el número óptimo



Las técnicas de reducción de la dimensionalidad tienen como objetivo disminuir el número de dimensiones del conjunto de variables en las situaciones anteriores (sobreajuste, interpretación, colinealidad, etc.). Como resultado de aplicar estas técnicas el conjunto de variables explicativas es sustituido por otro conjunto, de menor dimensión, con las variables más relevantes del problema analizado. Estas técnicas suelen dividirse en dos enfoques (feature selection y feature extraction) según las características de cada uno de ellos. La principal diferencia entre estos enfoques es que las técnicas de feature selection elige un subconjunto entre las variables originales del problema, mientras que las técnicas de feature extraction crean nuevas variables basadas en las originales.

- **Feature selection:** El objetivo de estas técnicas es seleccionar un subconjunto de variables originales que no contenga redundancia y sea relevante para el problema en cuestión. Con el uso de estas técnicas mejoran los resultados de los modelos, disminuye el tiempo de entrenamiento de los modelos, reduce el problema del sobreajuste y facilita la interpretación, entre otros. Estas técnicas se pueden agrupar en dos tipos de estrategias: filter y wrapper.



Las técnicas basadas en una estrategia filter evalúan la bondad de las variables por medio a una métrica de relevancia que analiza, de forma intrínseca, su relación con la variable de estudio. Algunas de las métricas más comunes son: la ganancia de información, la información mutua, la correlación de Pearson, análisis de la varianza, entre otras muchas. A partir de la métrica seleccionada, las variables quedan ordenadas, seleccionándose las k primeras para inducir con ellas el modelo deseado y suprimiendo las variables menos interesantes

Estos métodos son particularmente efectivos en el tiempo de cómputo, pueden escalar fácilmente a conjuntos de datos de dimensiones muy elevadas y son independientes de la técnica de modelado. Como resultado, la selección del subconjunto de variables se realiza una sola vez y luego se pueden evaluar diferentes paradigmas de modelado.

Una desventaja de estas técnicas es que cada variable se considera por separado, ignorando así las dependencias con otras variables, lo que puede llevar a un peor rendimiento en el modelo. Además, en este tipo de técnicas se debe fijar el punto de corte en el ranking de variables. El valor de este umbral afecta directamente al conjunto de variables seleccionados como relevantes y, consecuentemente, a los resultados del modelo.

Las técnicas basadas en una estrategia wrapper evalúan la bondad de los subconjuntos de variables por medio de un algoritmo de aprendizaje. Para cada subconjunto de variables se entrena un modelo y se obtiene un score del performance de dicho subconjunto. Analizar exhaustivamente cada subconjunto de variables es computacionalmente muy intenso por lo que, normalmente, se utilizan métodos de búsqueda para guiar el proceso de exploración sobre el subconjunto óptimo de variables. Algunos de los métodos más empleados son: forward selection, backward selection y genetic algorithms, entre otros.

Las ventajas de este tipo de estrategia es que, a diferencia de la estrategia anterior, tiene en cuenta las dependencias entre las variables del subconjunto seleccionado y la relación de éstas con el algoritmo empleado. Como resultado, estas técnicas normalmente proporcionan el mejor subconjunto de variables para el algoritmo seleccionado. Sin embargo, los principales inconvenientes de estos métodos son: el riesgo a producir sobreajuste en el modelo cuando el número de observaciones es insuficiente y el aumento en el coste computacional cuando el número de variables es elevado, especialmente cuando el algoritmo seleccionado es muy complejo.

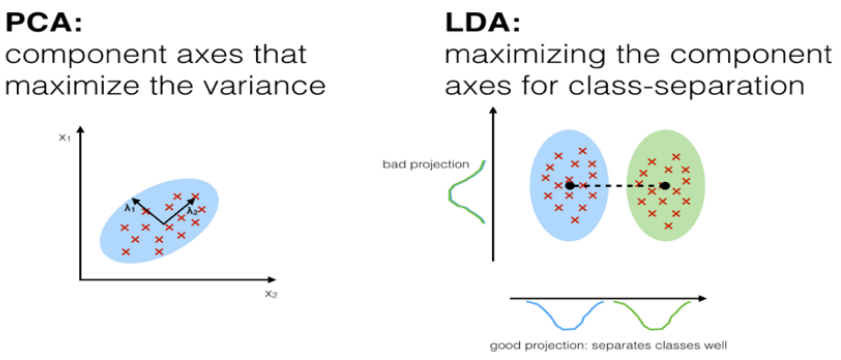
- **Feature extraction:** Las técnicas pertenecientes a este enfoque transforman el conjunto de variables originales en otro subconjunto de menor dimensión con el objetivo de que, en este nuevo espacio de variables, el problema tenga menor complejidad en el análisis. La aplicación de estos métodos permite crear nuevas variables con mucho poder predictivo y reduce el problema del sobreajuste. Sin embargo, uno de los principales inconvenientes es la pérdida de interpretabilidad tras la generación de las nuevas variables. Además, este proceso puede ser costoso computacionalmente.

  Los algoritmos basados en este paradigma se agrupan en dos grandes categorías: algoritmos lineales y algoritmos no lineales. Los algoritmos lineales son más robustos, más interpretables y requieren de menos tiempo de computación que los algoritmos no lineales. En cambio, los métodos no lineales pueden descubrir complejas estructuras en los datos que los métodos lineales no consiguen identificar.

  Principal Component Analysis (PCA) es el principal algoritmo lineal para abordar la reducción de la dimensionalidad debido a su versatilidad y sencillez. La idea subyacente de esta técnica es que el conjunto de datos inicial con alta dimensionalidad es linealmente separable en un espacio de menor dimensión. Para ello, se crean un conjunto de nuevas variables, denominadas componentes principales, mediante la combinación lineal de las variables originales. Las componentes principales generadas definen el nuevo espacio de dimensión reducida y maximizan la varianza del conjunto de datos inicial. Además, estas componentes quedan ordenadas en función de la cantidad de varianza explicada en los datos, por lo que se puede establecer un compromiso entre el número de dimensionales del conjunto de variables final y la varianza acumulada del conjunto de datos inicial.

Algunas de las ventajas que proporciona este algoritmo es la efectividad en la identificación de nuevas variables relevantes para el problema. Estas variables se corresponden con las componentes principales situadas en las primeras posiciones del ranking. Además, las nuevas componentes no están correlacionadas entre sí, lo que permite obtener mejor performance en los modelos. Sin embargo, el principal inconveniente de estas técnicas es la interpretabilidad de sus componentes. Otros inconvenientes de estas técnicas son: la imposibilidad de capturar relaciones no lineales, la necesidad de eliminar los outliers y normalizar los datos antes de su ejecución. En este contexto, algoritmos como Robust PCA, Kernel PCA y Nonlinear PCA, entre otros, han sido propuestos para dar solución a algunos inconvenientes.

Linear Discriminant Analysis (LDA) es otro de los algoritmos más comunes para reducción de la dimensionalidad. Este método genera nuevas variables mediante combinaciones lineales de las variables originales, pero a diferencia del PCA, se focaliza en optimizar la separación entre las clases del problema. Existen otras alternativas como quadratic LDA que es capaz de capturar relaciones no lineales. Además, al igual que PCA, estos algoritmo son sensible al número de componentes y a la escala de los valores.



1.4. Ejemplos de planteamientos analíticos

En esta esta sección se muestran varios ejemplos de planteamientos analíticos para resolver casos de uso. Concretamente, se analizan tres problemáticas de negocio:

- **Detectar los clientes más propensos a realizar upselling**

  El objetivo de este caso de uso es identificar a un subconjunto, de los clientes ya existentes, que tenga más probabilidad de aceptar una campaña de upselling. Concretamente, se trata de analizar la posibilidad de que los clientes de una compañía promocionen de la "tarjeta plata" a la "tarjeta oro". Para ello, disponemos de un conjunto de variables asociadas a cada cliente y el histórico de los clientes que han promocionado a la tarjeta oro.

  Este problema se afronta con un modelo supervisado ya que para algunos registros del histórico disponemos de la información que queremos predecir (promoción a la tarjeta oro). Dado este problema, lo que queremos predecir es si un cliente "promociona" o "no promociona" a la tarjeta oro, por lo que es necesario la utilización de los algoritmos de clasificación. Dependiendo del algoritmo utilizado, los modelos podrán ser más o menos precisos en los resultados, explicativos para entender las predicciones realizadas o flexibles para identificar relaciones entre las variables.

- **Estimar las ventas anuales de un nuevo comercio**

  El objetivo de este caso de uso es conocer el valor total de las ventas anuales de un determinado comercio en base a información relacionada con comercios ya existentes. Para ello, disponemos de un conjunto de variables asociadas a cada comercio y el histórico de las ventas anuales, en euros, registradas por cada comercio.

El problema de la estimación de ventas se resuelve a través de modelos supervisados, ya que disponemos de las ventas de otros comercios para guiar el aprendizaje del modelo. El número exacto de euros de la facturación anual es la predicción realizada por lo que, al estimar una variable continua, es necesario el uso de algoritmos de regresión. Si las variables predictoras tienen una relación lineal con las ventas anuales, entonces se puede utilizar el algoritmo linear regression, si no es el caso, se pueden utilizar algoritmos de regresión basados en árboles de decisión o redes neuronales, entre otros. En el caso de que la predicción realizada fuese un conjunto de valores categóricos (alta, media, baja) sobre las ventas anuales, entonces se pueden utilizar los algoritmos de clasificación.

- **Identificar anomalías producidas en transacciones bancarias**

  El objetivo de este caso de uso es detectar aquellas transacciones que tienen un comportamiento anómalo distinto a las demás. Para ello, se dispone de muchas características sobre las transacciones.

  Dado que no tenemos ninguna información que guíe el proceso de aprendizaje, se utiliza un enfoque no supervisado para la resolución del problema. Según la salida esperada, se puede utilizar un algoritmo particional, jerárquico o probabilístico, entre otros. La gran mayoría de estos algoritmos de clustering no son apropiados cuando existen un gran número de variables predictoras, por lo que hay que emplear técnicas de reducción de la dimensionalidad. Dependiendo del requisito de interpretabilidad de las variables, se selecciona un método de feature selection o de feature extraction. Finalmente, con el conjunto de variables resultantes, se procede a la construcción del modelo de clustering. Los clústers formados por un conjunto pequeños de registros tienden a ser los más anómalos.

1.5. Comparativa de algoritmos

En esta sección se comparan los principales algoritmos supervisados y no supervisados descritos anteriormente mediante unas tablas resumen sobre las ventajas e inconvenientes de cada uno de ellos.

La primera tabla muestra la comparativa entre los algoritmos supervisados. Dicha tabla incorpora información relacionada con el problema que resuelven, la tipología de relaciones que detectan, la complejidad de los patrones detectados, la interpretabilidad de las predicciones realizadas, su capacidad al sobreajuste y la complejidad computaciones, entre otras métricas

Supervised Learning	Linear Regression	Logistic Regression	Decision Tree	Support Vector	Neural Network
Problema	Regresión	Clasificación	Clasificación Regresión	Clasificación Regresión	Clasificación Regresión
Relaciones	Lineales	No lineales	No lineales	No lineales	No lineales
Patrones	Sencillos	Sencillos	Sencillos	Complejos	Muy complejos
Explicables	Si	Si	Si	No	No
Overfitting	No	No	Si	Si	Si
Sesgo	Alto	Alto	Bajo	Bajo	Bajo
Varianza	Baja	Baja	Alta	Alta	Alta
Cómputo	Eficiente	Eficiente	Eficiente	Costoso	Muy costoso

La segunda tabla muestra la comparativa entre los algoritmos no supervisados. Dicha tabla muestra el enfoque utilizado por cada algoritmo, su adecuación antes situaciones con muchos registros, muchas variables, valores atípicos y su complejidad computacional.

Unsupervised Learning	Tipología algoritmo	Muchos registros	Muchas variables	Valores outliers	Complejidad computacional
Kmeans	Particional	Adecuado	No	Sensible	Baja
PAM	Particional	No	No	Robusto	Alta
CLARA	Particional	Adecuado	No	Robusto	Media
BIRCH	Jerárquico	Adecuado	No	Robusto	Baja
CURE	Jerárquico	Adecuado	Adecuado	Robusto	Baja
CHAMELEON	Jerárquico	No	No	Robusto	Alta
DBSCAN	Densidad	Adecuado	No	Robusto	Media

TEMA: Riesgos en los modelos generados

Durante el entrenamiento de los modelos de aprendizaje supervisado pueden surgir varios problemas como el sesgo y la varianza del algoritmo, el sobreajuste del modelo a los datos de entrenamiento, el desbalanceo en la distribución de las categorías del problema y los errores cometidos en la selección de procesos y métricas de evaluación, entre otros.

El efecto del sobreajuste o overfitting ocurre cuando el modelo generado es capaz de captura el ruido de los datos y los patrones subyacentes en los mismos. Estos modelos tienen bajo sesgo y alta varianza. En cambio, el efecto contrario, underfitting, se produce cuando el modelo no puede capturar los patrones subyacentes en los datos. A diferencia de los anteriores, estos modelos suelen tener alto sesgo y baja varianza. Esto puede ocurrir cuando tenemos poca información, tanto a nivel registros como variables, para construir un modelo preciso o cuando se intenta construir un modelo lineal con datos no lineales.

El desbalanceo de clases en un problema de clasificación supervisada es otro de los riesgos más importantes, ya que la mayoría de los algoritmos existentes tienden a ignorar la clase minoritaria y se focalizan en clasificar correctamente las clases mayoritarias. Por lo tanto, se optimiza la precisión general del modelo sin considerar una distribución relativa de cada clase.

El último de los riesgos analizados es la elección de métricas de evaluación para identificar, correctamente, como de bueno es el modelo construido. Estas métricas deben cumplir con los requisitos técnicos y de negocio. Además de seleccionar las métricas, para cada problema es necesario establecer un conjunto de métodos honestos que estimen, de formar correcta, las métricas propuestas.

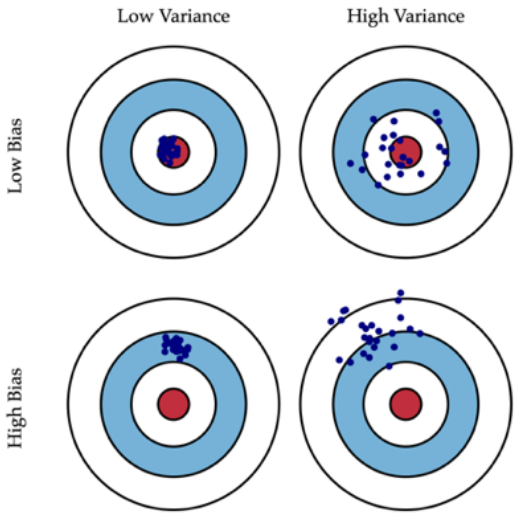
2.1. Selección del algoritmo

El error de predicción para cualquier algoritmo supervisado se puede dividir en el error de sesgo (bias), el error de varianza (variance) y el error irreducible. El error irreducible o ruido no se puede eliminar. Independientemente del algoritmo utilizado, los datos siempre tendrán cierta cantidad de ruido. Sin embargo, los otros dos tipos de errores se pueden disminuir porque se derivan de la elección del algoritmo.

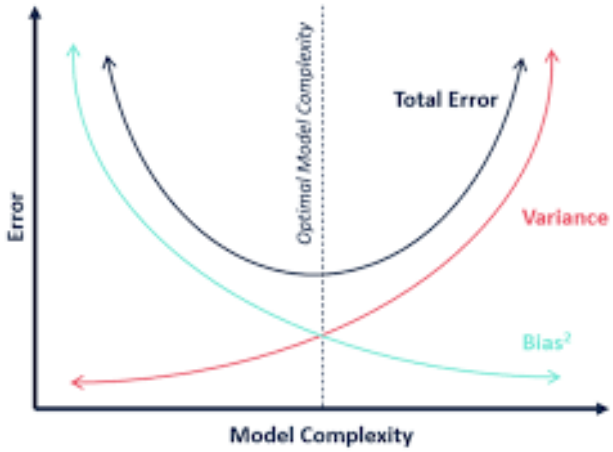
- **Bias:** Es una medida que refleja la rigidez del modelo y, para ello, estima la diferencia entre los valores predicho y reales. El objetivo de cualquier modelo es predecir datos muy próximos a los valores reales, pero esto no es siempre posible por la rigidez de algunos algoritmos. En general, algoritmos como la regresión lineal o la regresión logística tienen un alto sesgo que los hace rápidos de aprender y más fácil de entender. Estos algoritmos son menos flexibles para capturar relaciones complejas en los datos y, por lo tanto, tienen peor rendimiento predictivo, tanto en la fase de entrenamiento como en la fase de validación. Por el contrario, los algoritmos que tienen bajo sesgo son los árboles de decisión, los k-vecinos más cercanos y las máquinas de vectores de soporte.

- **Variance:** Es una medida que refleja la inconsistencia del modelo y, para ello, calcula la variabilidad en las predicciones del modelo cuando se modifican los datos de entrenamiento. Los algoritmos que tienen una gran varianza están influenciados por los detalles de los datos de entrenamiento, incapacitando al modelo a generalizar los patrones aprendidos. Algoritmos como los árboles de decisión, los k-vecinos más cercanos y las máquinas de vectores de soporte tienen una alta varianza. Estos métodos logran muy poco error sobre el conjunto de entrenamiento, pero obtienen mucho error sobre los datos de validación. Por el contrario, los algoritmos que tienen baja varianza son la regresión lineal, el análisis discriminante lineal y la regresión logística, entre otros.

Aunque el objetivo de cualquier algoritmo supervisado es lograr un error de sesgo bajo y un error de varianza bajo, existe un trade-off entre ambas métricas. En este contexto, cuando aumenta el sesgo de un modelo, disminuye la varianza de este, y viceversa.



Analizar los errores de sesgo y varianza es fundamental para entender el comportamiento de los modelos de predicción, pero en general lo que realmente importa es el error general, no la descomposición específica. El punto óptimo para cualquier modelo es el nivel de complejidad en el que el aumento en el sesgo es equivalente a la reducción en la varianza.

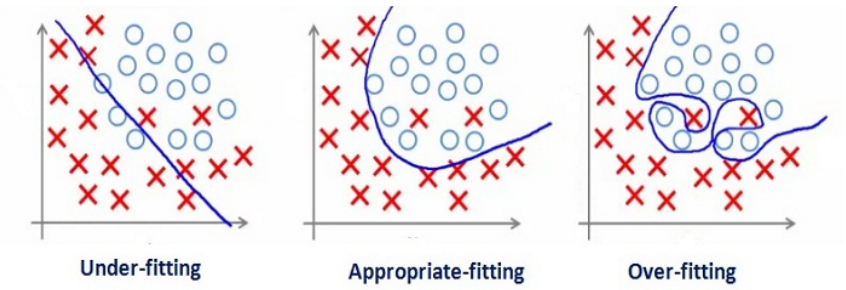


2.2. Sobreajuste de los modelos

El objetivo principal de cualquier algoritmo de aprendizaje automático es descubrir patrones en el conjunto de datos de entrenamiento y, posteriormente, aplicar dichas generalizaciones en la predicción de nuevos datos. En este contexto, puede ocurrir que durante el entrenamiento del modelo sólo se descubran casualidades en los datos que insinúen patrones interesantes pero que no generalicen bien. Esto es lo que se conoce con el nombre de overfitting o sobreajuste.

Un modelo está sobreajustado cuando las métricas de evaluación son buenas para el conjunto de entrenamiento y notablemente peores para el conjunto de validación. Esto se debe a que el modelo ha aprendido el detalle, incluyendo el ruido y fluctuaciones de los datos observados y no ha sido capaz de generalizar unos patrones para los datos no observados. Este hecho es más probable cuando se utilizan algoritmos no lineales, ya que tienen más libertad para construir el modelo según el detalle de los datos de entrenamiento.

El problema del sobreajuste se produce debido al exceso de complejidad en el diseño de la frontera entre las clases analizadas. En cambio, cuando existe un exceso de generalización, el modelo no es capaz de diferenciar las muestras del conjunto de entrenamiento, produciéndose así, el efecto llamado underfitting o subajuste. La siguiente imagen muestra los dos conceptos descritos.



Todos los algoritmos de aprendizaje automático tienen tendencia al sobreajuste y, por este motivo, se han de realizar ciertas acciones preventivas para que no se produzca o reducirlo. Las principales estrategias son las siguientes:

- **Número de registros:** Se debe establecer una cantidad mínima de muestras tanto para entrenar el modelo como para validarlo. Generalmente, el uso de grandes conjuntos de datos ayuda a los algoritmos a identificar patrones en los datos y, así, detectar la frontera de decisión apropiada, evitando el sobreajuste.
- **Distribución de las clases:** En caso de la clasificación supervisada es importante que los datos de entrenamiento estén balanceados, ya que, los algoritmos tienen menos predisposición al sobreajuste de los datos cuando las clases equilibradas en cantidad.
- **Número de variables:** Una cantidad excesiva de dimensiones con muchas variantes y sin suficientes muestras también puede generar sobreajuste. A veces conviene reducir la dimensionalidad de las variables que son usadas para el entrenamiento del modelo.
- **Selección del algoritmo:** Para evitar el sobreajuste en los datos se aconseja iniciar el aprendizaje con un modelo simple e ir probando progresivamente con modelos más complejos hasta encontrar el que mejor se ajuste a las necesidades. En esa búsqueda del modelo óptimo se recomienda utilizar métodos que eviten el sobreajuste de los datos como la poda para los árboles de decisión, las penalizaciones de los coeficientes en las regresiones y la reducción de las capas ocultas en las redes neuronales.
- **Ajuste de parámetros:** Al entrenar iterativamente un modelo se detecta que, hasta cierto número de iteraciones, el rendimiento del modelo mejora. Después de cierto punto, si aumenta el número de iteraciones, el modelo tiene un mejor rendimiento en el conjunto de entrenamiento y peor en el conjunto de validación. Por lo tanto, se deben detener las optimizaciones en los ajustes de los parámetros antes de que exista un ajuste excesivo en el modelo.

2.3. Desbalanceo de clases

El desarrollo de modelos analíticos a partir de conjuntos de datos no balanceados es uno de los desafíos que actualmente está enfrentando el aprendizaje automático, ya que muchos de los algoritmos de clasificación están diseñados para optimizar la precisión general del modelo, sin considerar una distribución relativa de cada clase.

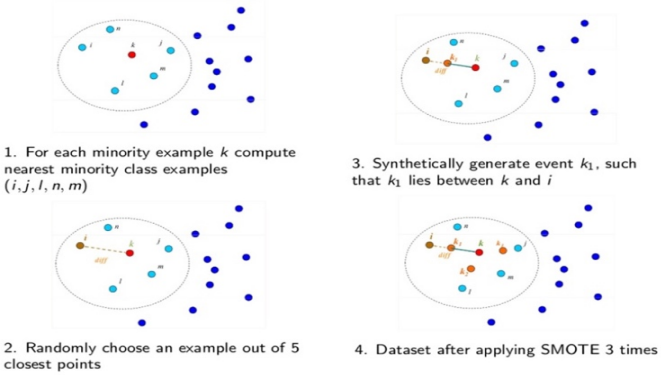
En los problemas con datos desbalanceados, el conocimiento más novedoso suele residir en los datos menos representados, sin embargo, muchos clasificadores, como los árboles de decisión, los perceptrones multicapas y los clasificadores bayesianos, pueden considerarlos como ruido, ignorando las clases minoritarias y focalizándose en clasificar correctamente las clases mayoritarias.

Numerosas técnicas han sido propuestas con el objetivo de solucionar dicho problema. Estas técnicas se agrupan en dos enfoques dependiendo de si modifican o no la distribución de los datos del conjunto inicial.

Las técnicas basadas en cost-sensitive learning no modifican la distribución de los datos y se focalizan en modificar el algoritmo de aprendizaje. Dichos métodos utilizan los costes asociados a la clasificación errónea de las muestras durante el entrenamiento del modelo. Sin embargo, en estos métodos, el coste del error debe ser conocido de antemano y, en un problema de clasificación real, dicho coste es a menudo desconocido. Por otro lado, las técnicas de sampling tratan de balancear la distribución original de los datos por medio de la generación de nuevas instancias de la clase minoritaria (over-sampling), de la eliminación de instancias de la clase mayoritaria (under-sampling), de la hibridación de ambas (hybrid-sampling). A su vez, estas técnicas de sampling pueden categorizarse en tres grupos en función del tipo de método de muestreo: aleatorio, duplicados o proximidad de sus vecinos. A continuación, veremos algunas de las técnicas más importantes para el muestreo de la información.

- **Over-Sampling:** Una manera rápida de generar nuevas instancias de la clase minoritaria es mediante la utilización de la técnica Random OverSampling (ROS). Esta técnica consiste en agregar un subconjunto aleatorio de registros de la clase minoritaria dentro del conjunto de datos original. Otra alternativa de esta técnica es la modificación, de manera aleatoria, de una o más variables del registro, pero manteniendo el mismo valor de la clase inicial. OverSampling by Duplication (OBD) es una técnica muy similar a la anterior y consiste en realizar copias de todos los patrones de la clase minoritaria hasta obtener un grado de balance deseado. Una desventaja de las técnicas es que los registros duplicados generan información redundante y produce sobreajuste en el modelo.

  Una de las estrategias más conocidas para la generación de nuevas instancias de la clase minoritaria es el algoritmo Synthetic Minority OverSampling Technique (SMOTE). Este método busca los vecinos más cercanos para cada elemento de la clase minoritaria y, a través de la extrapolación de uno de sus vecinos, genera un nuevo registro entre el registro inicial y su vecino seleccionado. Sin embargo, esta técnica presenta el inconveniente de que puede introducir registros de la clase minoritaria en el área de la clase mayoritaria, creando ejemplos malos que posteriormente pueden confundir a los clasificadores. En este contexto, algoritmos como SMOTE Editing Nearest Neighbor y Bordeline SMOTE, entre otros, han sido propuestos para corregir algunos de los inconvenientes del método original.



Otros enfoques que persiguen el oversampling de la clase minoritaria son: Agglomerative Hierarchical Clustering (AHC) y Clustered Based Oversampling (CBOS), entre otros.

- **Under-Sampling:** Una manera sencilla de abordar este problema es mediante la técnica Random UnderSampling (RUS). Esta técnica consiste en eliminar datos, de forma aleatoria, asociados a la clase mayoritaria, consiguiendo un conjunto de datos más balanceado. El principal inconveniente de esta técnica es que la eliminación de los registros aleatorios puede implicar la eliminación de patrones relevantes para el modelo. Con el objetivo de solucionar el problema anterior, el algoritmo One-Sided Selection (OSS) no selecciona de forma aleatoria los registros a eliminar, sino que se seleccionan aquellos elementos que son redundantes o ruidosos.

  El algoritmo Neighborhood Cleaning Rule (NCR) selecciona, para cada registro del conjunto de datos, sus tres vecinos más próximos. Una vez calculados, si el registro seleccionado es de la clase mayoritaria y los tres vecinos son de la minoritaria, entonces se elimina el registro seleccionado. En cambio, si el registro pertenece a la clase minoritaria entonces se eliminan los vecinos que sean de la mayoritaria. Otros algoritmos que también utilizan el enfoque de los vecinos más próximos son Wilson's Editing (WE) y Nearest Synthetic Undersamplig (NSU), entre otros.

- **Hybrid-Sampling:** A pesar de que las técnicas de oversampling y undersampling logran buenos resultados por separado, varias técnicas híbridas han sido propuestas para obtener las ventajas de los dos enfoques. Uno de los algoritmos más comunes es llamado SMOTE-Bootstrap Hybrid que genera nuevas instancias de la clase minoritaria haciendo uso de SMOTE y luego reduce la clase mayoritaria a través de Bootstrap hasta lograr que las clases queden con similar número de instancias. Otros algoritmos basados en el enfoque híbrido son SMOTE-Tomek Hybrid y AHC-KM Hybrid.

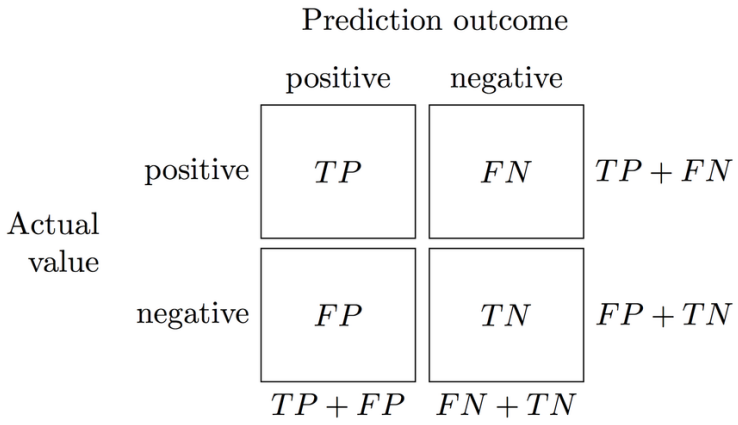
2.4. Evaluación de los modelos

El proceso de evaluación es la fase más importante dentro de cualquier problema de aprendizaje supervisado, ya que su objetivo es conocer cómo de buenos son los valores predichos para los nuevos registros. La evaluación de los resultados es un procedimiento relativamente simple debido a la presencia de los valores reales. En este contexto, un algoritmo de clasificación clasifica correctamente un nuevo registro si la clase predicha es la misma que la clase real, mientras que un algoritmo de regresión realiza buenas predicciones si la diferencia entre el valor predicho y valor real es mínima. En caso contrario, se considera que el algoritmo ha cometido un error en la predicción.

Para evaluar un modelo de clasificación se puede utilizar la proporción entre las predicciones correctas que ha hecho el modelo y el total de predicciones. Esta es una métrica muy sencilla denominada accuracy. Sin embargo, aunque en ocasiones resulta práctica por su facilidad de cálculo, otras veces es necesario profundizar un poco más y tener en cuenta los tipos de predicciones correctas e incorrectas que realiza el clasificador. Es aquí donde entra en juego la confusion matrix.

La matriz de confusión es una herramienta importante para validar los resultados de los algoritmos de clasificación. Dado un problema de clasificación binaria, cada columna de la matriz de confusión representa las instancias que han sido predichas como positivas o negativas, mientras que en las filas se representan el valor real.

Los valores de la diagonal principal de la matriz corresponden a las instancias correctamente predichas, que son el número de verdaderos positivos (TP) y el número de verdaderos negativos (TN). Los otros valores corresponden a errores en la predicción y se dividen en falsos negativos (FN) y falsos positivos (FP).



La matriz de confusión juega un papel fundamental en la evaluación de conjuntos de datos desbalanceados. Un algoritmo de clasificación que siempre predice la clase mayoritaria tiene un accuracy cercano al 100% pero, obviamente, no es un clasificador útil. En este contexto, la métrica accuracy conduce a conclusiones erróneas debido a que las clases minoritarias solo tienen un efecto pequeño en esta métrica.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

En situaciones como la anterior, es recomendable analizar y calcular otras métricas más adecuadas para evaluar la exactitud del modelo. Algunos de las métricas más comunes son: precision, recall y F-score.

Las métricas precision y recall reflejan el rendimiento del modelo cuando realiza sus respectivas predicciones. En este contexto, un algoritmo muy exacto (valor alto de precision) identifica muchos más registros relevantes que irrelevantes, mientras que un algoritmo muy específico, (valor alto de recall) detecta la mayoría de los registros relevantes para el problema.

- **Precision:** Es la métrica de evaluación que permite conocer qué proporción de los predichos como positivos lo son realmente. El valor de la métrica oscila entre cero y uno. El modelo que obtenga un valor de precision igual a uno significa que el número de verdaderos positivos (TP) es igual al número total de valores positivos predichos (TP + FP).

$$\text{precision} = \frac{TP}{TP + FP}$$

- **Recall:** Permite conocer qué proporción de todos los casos positivos se clasifican como tal. El valor de la métrica oscila entre cero y uno. El modelo que obtenga un valor de recall igual a uno significa que el número de verdaderos positivos (TP) es igual al número total de valores positivos reales (TP + FN). Esta métrica también se denomina True Positive Rate.

$$\text{recall} = \frac{TP}{TP + FN}$$

- **F1-score:** Es una medida que combina los valores de las métricas precision y recall. Esta nueva métrica, que oscila entre cero y uno, alcanza su máxima puntuación cuando los valores de precision y recall son uno, es decir, cuando modelo realiza predicciones perfectas. El F1-score es necesario cuando se busca un balance entre precision and recall.

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Para evaluar un modelo de regresión basta con calcular la diferencia entre la predicción realizada y el valor real. A medida que esta diferencia es mayor, mayor es el error cometido por el modelo. A continuación, se muestran las principales métricas para regresión.

- **Mean Squared Error:** Es la métrica de evaluación más utilizada en los problemas de regresión. El cálculo del error cuadrático medio se obtiene al promediar la diferencia al cuadrado entre los valores reales y predichos para todos los puntos de datos. Cuanto menor es el valor de esta métrica, mejores son las predicciones del modelo.

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2, \text{where } e_t = original_t - predict_t$$

- **R-squared:** Es otra de las métricas más importantes para determinar la precisión de un modelo de regresión. En esta ocasión, cuanto mayor es el valor de la métrica, mejores son las predicciones del modelo. Un valor igual a 1 indica que el modelo predice de manera óptima. El inconveniente que tiene la métrica es que tiende a sobreestimar el ajuste de la regresión, ya que a medida que se agregan predictores al modelo, el valor de la métrica aumenta. Este aumento es artificial cuando los predictores no mejoran el ajuste del modelo. Para solucionarlo, la métrica Adjusted R-squared permite disminuir el valor calculado, a medida que se agregan variables predictoras poco relevantes para el modelo.

Otras métricas asociadas a la evaluación de problemas de regresión son: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Root Mean Absolute Error (RMAE), Mean Square Percentage Error (MSPE), Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE) y Root Mean Squared Logarithmic Error (RMSLE).

Además de seleccionar las métricas de evaluación oportunas para cada problema es necesario establecer un conjunto de métodos honestos que estimen, de formar correcta, las métricas propuestas. Algunos de los métodos más comunes son:

- **Resubstitution:** Es un método de estimación muy simple que consiste en entrenar un modelo utilizando el conjunto de datos completo y evaluar su rendimiento con los mismos datos. Este no es un buen método, ya que entrenar y validar con los mismos datos produce resultados muy optimistas. A pesar de ello, este método puede ser útil para establecer el límite superior en los resultados.
- **Hold-out:** Soluciona el problema de los resultados optimistas al dividir el conjunto de datos en dos subconjuntos separados, uno para inducir el modelo y el otro para su evaluación. La desventaja de este método es que el subconjunto de registros asociados a la evaluación no se utiliza para construir el modelo, y esto puede ser un problema si el tamaño del conjunto de datos no es muy elevado.
- **K-fold cross validation:** Es el método más utilizado para la evaluación de modelos. Esta técnica soluciona el problema anterior al dividir el conjunto de datos en k subconjunto aleatorios de igual tamaño aproximadamente. Cada subconjunto es utilizado para evaluar un modelo que ha sido aprendido con el resto de los subconjuntos generados. Los resultados de las distintas evaluaciones son promediados para conseguir el resultado final para el modelo.
- **Stratified k-fold cross validation:** Es una mejora del método anterior ya que intenta preservar la proporción de las clases en cada uno de los subconjuntos generados. Este método obtiene métricas más realistas especialmente cuando el conjunto de datos está desbalanceado.

2.5. Ejemplos de riesgos detectados

En esta esta sección se muestran varios ejemplos de los riesgos detectados durante la construcción de los modelos analíticos. Concretamente, se analiza el problema del overfitting y elección de la métrica de evaluación.

- **Identificación de perros a través de imágenes**

  Cuando se genera un modelo analítico a través de un conjunto de entrenamiento lo que se pretende es que **el algoritmo sea capaz de generalizar un concepto,** para que, al consultarle por un nuevo conjunto de datos **desconocido,** *éste sea capaz de comprenderlo y devolver un resultado fiable* dada su capacidad de generalización.

  Los conjuntos de entrenamiento pueden tener registros anómalos o muestras que pueden no ser del todo representativas. En el caso de uso de la identificación de perros, si el conjunto de imágenes de entrenamiento está formado por razas de perros solo de color marrón, entonces el modelo está ante un problema de overfitting. Esto se produce porque al mostrar una foto de un perro blanco, el modelo no es capaz de reconocerlo como perro, ya que esa foto no cumple con el patrón aprendido (el color forzosamente debe ser marrón). En este contexto, el modelo está considerando como perros válidos sólo aquellos idénticos a los del conjunto de entrenamiento, incluidos sus defectos, y no es capaz de distinguir nuevas imágenes de perros si se salen de los rangos establecidos.

- **Desarrollo de pruebas clínicas para diagnosticar enfermedades**

  Supongamos que dos laboratorios han desarrollado un conjunto de pruebas clínicas para diagnosticar una enfermedad presente en la población. Los resultados de las pruebas clínicas de cada laboratorio son las siguientes:

Laboratorio A		Valor real	
		Sano	Enfermo
Valor predicho	Sano	977	1
	Enfermo	13	9

Accuracy = 98,6% = (977+9) / (977+9+13+1)
Precision = 40,9% = (9) / (13+9)
Recall = 90% = (9) / (9+1)

Laboratorio B		Valor real	
		Sano	Enfermo
Valor predicho	Sano	1965	5
	Enfermo	7	23

Accuracy = 99,4% = (1965+23) / (1965+23+7+5)
Precision = 76,6% = (23) / (23+7)
Recall = 82,1% = (23) / (23+5)

A priori se puede pensar que el laboratorio B presenta una prueba de diagnosis más efectiva que la del laboratorio A, ya que tiene un mayor valor de la métrica accuracy. Además, al valorar las otras métricas se observa que el laboratorio A presenta una prueba con un 40,9% de precisión, esto es, de cada 10 veces que la prueba concluye enfermo, sólo 4 son realmente enfermedad, mientras que el laboratorio B obtiene un 76,6% en esta métrica. Sin embargo, al valorar una prueba de diagnosis lo primero que hay que que tener en cuenta es cuántos pacientes enfermos no son diagnosticados, y esto es lo que mide la métrica recall. Según la prueba del laboratorio A: 9 de cada 10 pacientes con la enfermedad son diagnosticados, mientras que para el laboratorio B ese porcentaje disminuye a 8 de cada 10. Por tanto, debemos concluir que la prueba A es mejor que la prueba B.

En conclusión, se ha observado que la métrica accuracy no es una buena métrica para valorar datos desbalanceados y que las métricas precisión y recall ofrecen una mejor visión de cuán bueno es un clasificador ya que se focaliza sólo en la parte positiva de la muestra. La decisión de escoger una métrica u otra depende del caso de uso analizado en cada momento y de cuanto queremos tener en cuenta los falsos positivos y falsos negativos

TEMA: Optimización de los modelos desarrollados

Como se ha descrito en el tema anterior, bias y variance son los dos componentes relacionados con la imprecisión en los modelos predictivos. Si el modelo obtiene buenos resultados en el conjunto de entrenamiento y malos en el conjunto de test, entonces el modelo tiene alta varianza (variance). Por otro lado, si el modelo obtiene malos resultados, tanto en el conjunto de entrenamiento como en el conjunto de test, entonces el modelo tiene alto sesgo (bias). Dependiendo del tipo de imprecisiones cometidas en los modelos, varias técnicas pueden ayudar a conseguir mejores resultados.

La primera técnica para mejorar los resultados de un modelo consiste en la incorporación de nuevos registros y variables al conjunto de entrenamiento. La inclusión de nuevos registros, normalmente, reduce la varianza del modelo (sin empeorar el sesgo), mientras que la incorporación de nuevas variables permite reducir el sesgo del modelo, a expensas de la varianza. Ambas modalidades permiten la utilización de algoritmos más flexibles, permitiendo así la posible detección de relaciones complejas. La única situación en la que no es recomendable incrementar el número de variables es cuando el conjunto de entrenamiento tiene muy pocos registros.

La transformación del conjunto de datos existente es otra de las opciones para mejorar los resultados del modelo analítico. Esta transformación consiste en generar nuevas variables predictoras a través de varios métodos como las funciones logarítmicas, funciones exponenciales, la generación de variables dummy, la discretización de variables y otras muchas técnicas. Estas nuevas variables deben tener gran capacidad para explicar la variabilidad sobre los datos y, así, proporcionar mejores resultados en el modelo. Además, la optimización de los procesos relacionados con el tratamiento de valores missing, la identificación de valores outliers y la elección de variables relevantes, también ayuda a mejorar los resultados.

Otra opción para mejorar los resultados de los modelos es cambiar la tipología del algoritmo seleccionado para el desarrollo del modelo. El aprendizaje ensamblado permite combinar los resultados de modelos simples, y genera un nuevo modelo más robusto y con mejores resultados. Bagging y Boosting son los algoritmos más empleados de este enfoque. Bagging reduce la varianza del modelo sin ningún efecto notable en el sesgo, mientras que Boosting disminuye el sesgo y apenas afecta la varianza. Por otro lado, la optimización de los parámetros de estos algoritmos es fundamental, ya que el valor de estos influye directamente en el proceso de aprendizaje y en el resultado final del modelo.

Por último, se debe tener en cuenta que la precisión del modelo no es el único objetivo. A pesar de que algunos modelos sean muy precisos, pueden ser muy difíciles de implementar en entornos de producción y cajas negras difíciles de interpretar o depurar. Por ello, muchos sistemas de producción optan por un modelo más simple, menos preciso y más fácil de implementar.

3.1. Mejoras sobre los datos

La aplicación de técnicas que aporten valor a la información existente puede ser crucial para obtener mejores resultados en los modelos analíticos. Aunque existen tratamientos sobre los datos que, tras su aplicación, incrementan el rendimiento del modelo, es el proceso de feature engineering el más relevante para conseguir tal fin.

Este proceso es considerado como el arte de extraer información relevante sobre los datos ya existentes. Feature engineering pretende transformar los datos actuales en información aún más útil de cara al proceso de modelización. Varios métodos como feature transformation y feature creation son empleados para lograr mejores resultados en los distintos modelos analíticos.

- **Feature transformation:** Esta tarea consiste en la transformación de las variables del conjunto de datos para poder ajustarse mejor a las asunciones y requisitos del algoritmo seleccionado para la construcción del modelo. A continuación, se mencionan algunos escenarios en los que estos métodos son aplicables: realizar un cambio de escala para tener los valores normalizados entre cero y uno, eliminar el sesgo de una variable para obtener normalidad en los datos, o discretizar una variable numérica en intervalos, y así, aprovechar el valor de los datos atípicos.

- **Feature creation:** Esta tarea se basa en la creación de nuevas variables derivadas de las ya existentes, que ayuden a detectar la relación oculta entre los datos del conjunto de entrenamiento. Uno de los enfoques trata de aislar la información relevante para el problema en variables que representen umbrales, o combinaciones de variables. Otros enfoques tratan de destacar las relaciones entre variables, generando así, nuevas variables que son las sumas, restas, multiplicaciones o divisiones de variables ya existentes. Por otro lado, la agregación de valores sparse, junto con la generación de variables dummy, son otros métodos para representar la misma información, ya existente, de una manera muy diferente.

La presencia no deseada de valores missing y outliers en el conjunto de entrenamiento a menudo reduce la precisión del modelo, generando modelos con mucho sesgo. En este contexto, también es importante tratar dichos valores antes de la construcción del modelo.

Existen varios métodos para el tratamiento de los valores misssing como, por ejemplo, la eliminación de los registros que contengan dichos valores, la sustitución de dichos valores por la media o la mediana en variables numéricas y la moda en variables categóricas, la predicción de esos valores por medio de algoritmos de aprendizaje automáticos, y la imputación de valores por medio del algoritmo nearest neighbours. Dependiendo del método utilizado, los resultados de los modelos pueden variar.

Los valores outliers pueden cambiar drásticamente los resultados de un modelo, ya que, si no son tratados antes de su construcción, pueden generar un gran incremento en la varianza del error. La mayoría de los métodos utilizados para lidiar con los valores atípicos son similares a los usados para los valores missing. La eliminación de observaciones, la transformación de los datos, la agrupación de estos y la imputación de valores son algunos de los métodos más comunes.

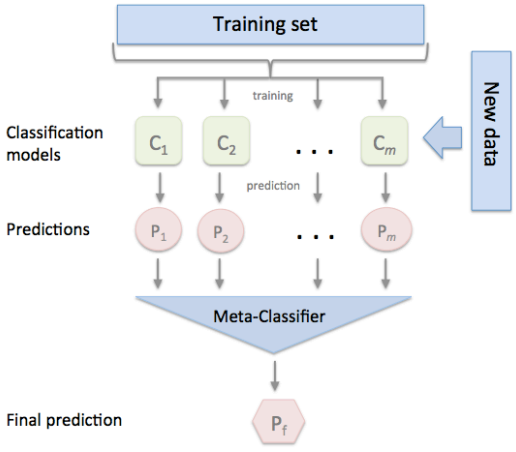
Finalmente, el uso de técnicas de selección de variables y regularización también pueden afectar a la mejora en los resultados de los modelos. Estas técnicas son recomendables cuando existen muchas variables y pocos registros en el conjunto de entrenamiento. Su aplicación permite disminuir la varianza, a expensas del sesgo. Si el conjunto de entrenamiento dispone de mucha información, los algoritmos actuales pueden detectar el ruido en los datos y variables poco relevantes, sin tener la necesidad de aplicar dichas técnicas.

3.2. Mejoras sobre los algoritmos

La aproximación básica para crear modelos supervisados es inducirlos a partir de un conjunto de datos que contengan tanto las variables predictoras como la variable respuesta. Sin embargo, las técnicas conocidas como ensemble pueden construir modelos de mejor capacidad predictiva utilizando como input las predicciones realizadas por varios modelos previos. Estas técnicas consisten en la agregación de modelos individuales para generar un modelo global más predictivo y estable que aproveche el conocimiento del colectivo. En cambio, la combinación de distintos algoritmos en un solo modelo hace muy compleja su interpretación y la elección de los parámetros que mejor se ajustan a los datos.

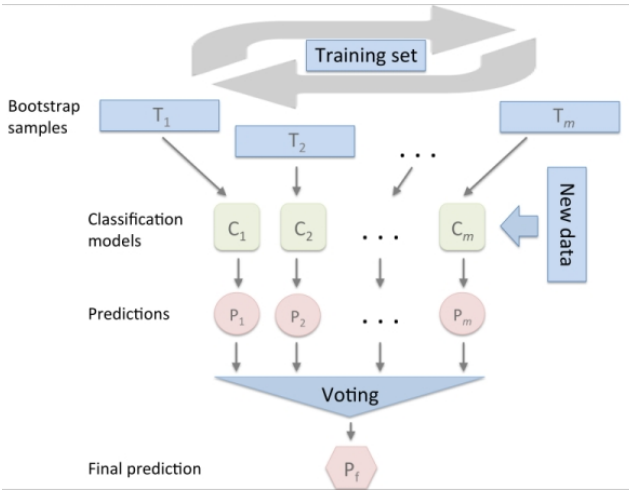
Existen dos enfoques principales para agregar el conjunto de modelos individuales. La primera aproximación consiste en combinar modelos heterogéneos a través de la técnica stacking, mientras que la segunda aproximación fusiona modelos homogéneos usando técnicas como bagging y boosting, entre otras.

- **Stacking:** Esta técnica permite entrenar un conjunto de modelos individuales generados por distintos algoritmos de aprendizaje. A partir de las predicciones de dichos modelos, esta técnica genera un segundo modelo de aprendizaje automático que combina la información heterogénea de los modelos previos.



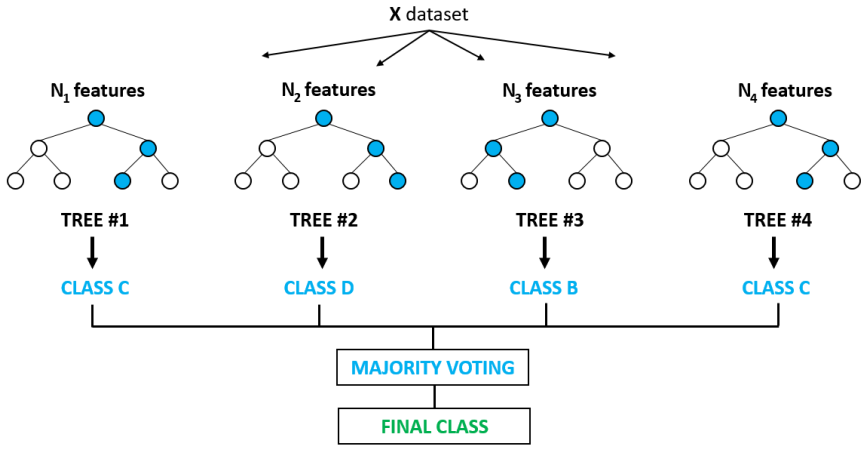
Cuando el modelo de segundo nivel es una regresión lineal, la predicción final es una combinación lineal de las estimaciones realizadas por los modelos del primer nivel. Sin embargo, el modelo del segundo nivel no esta restringido a un modelo lineal, ya que la relación entre los predictores puede ser de mucha complejidad.

- **Bagging:** La técnica Booststrap aggregating es un método de agregación de modelos homogéneos que combina las predicciones realizadas por los distintos clasificadores. Aunque lo más común es aplicar árboles de decisión en dicho enfoque, también se puede utilizar otras familias de algoritmos supervisados. Algoritmos como random forest y extremely randomized trees han reutilizado dicho enfoque para generar sus propias versiones del algoritmo.



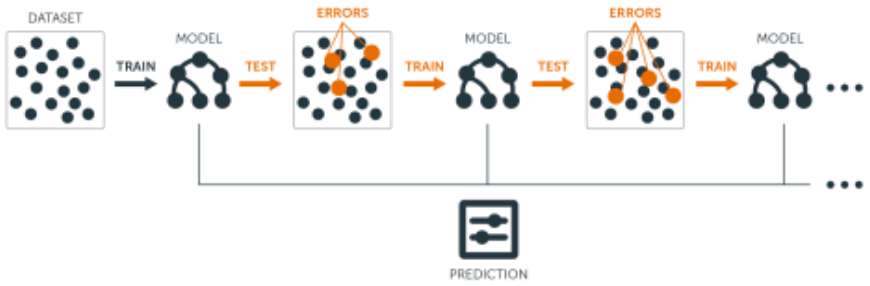
Gracias al enfoque utilizado, el sesgo de un determinado algoritmo no influye, de manera drástica, en la predicción final del modelo, ya que éste tiene en cuenta el resto de las predicciones parciales. Esto permite una mayor robustez ante el problema de la varianza y del sobreajuste de los datos. En cambio, la combinación de algoritmos en un solo modelo hace muy compleja su interpretación y conlleva mucho tiempo de computo la elección de los parámetros del modelo que mejor se ajustan a los datos.

- **Random Forest:** Este algoritmo utiliza una modificación del enfoque bagging para agregar las predicciones de distintos árboles de decisión. Cada uno de los árboles se entrena con una muestra aleatoria del conjunto de datos y la salida del modelo final se construye combinando los resultados parciales mediante métodos de votación o valores promedios. Este algoritmo es muy similar a la técnica bagging cuando el algoritmo base es un árbol de decisión. La única diferencia entre ellos es que, el algoritmo random forest selecciona un subconjunto de variables para la identificar la mejor división en cada uno de los árboles del modelo, mientras que la técnica bagging tiene en cuenta todas las variables del conjunto de datos para analizar dicha situación.



Este algoritmo puede ser considerado como de propósito general, ya que obtiene buenos resultados en la gran mayoría de los problemas. Su parametrización es sencilla y computacionalmente es muy efectivo debido a la capacidad de ejecución en paralelo. Al igual que los árboles de decisión, esta técnica captura relaciones no lineales entre las variables, admite distintos tipos de variables predictoras, valores vacíos, valores extremos y consigue una identificación de características relevantes del problema. Además, obtiene resultados satisfactorios en conjuntos de datos con alta dimensionalidad, mejorando las métricas de performance de los árboles más simples.

- **Boosting:** Este nuevo enfoque consiste en generar una secuencia de modelos que corrijan, de manera iterativa, los errores cometidos en las iteraciones previas. Como resultado de agregar las predicciones de cada iteración, se genera un modelo más eficiente, que disminuye la varianza y también el sesgo. A diferencia del bagging, no se crean muestras del conjunto de entrenamiento, sino que se trabaja siempre con el conjunto completo de entrada. La idea es que en cada iteración se incremente el peso de los objetos mal clasificados por el predictor en esa iteración, por lo que en la construcción del próximo predictor estos objetos serán más importantes y será más probable clasificarlos bien.



El algoritmo Adaboost es sin duda el algoritmo del boosting más conocido. Este método utiliza, por lo general, arboles de decisión como modelos base. A pesar de que los modelos boosting son más difíciles de optimizar, tienden a obtener mejores resultados que los modelos bagging, ya que, al focalizarse sobre los registros mal clasificados, el error disminuye rápidamente. Otros algoritmos como gradient boosting y stochastic gradient boosting también se basan en este enfoque.

Los algoritmos basados en boosting son susceptibles de overfitting. Una forma de evitarlo es mediante un valor de regularización (learning rate), que limite la influencia de cada modelo. Como consecuencia de este valor, se necesitan más modelos para formar el ensemble pero, a cambio, se consiguen mejores resultados.

TEMA: Aplicaciones Big Data Analytics en el negocio

Esta sección aborda la necesidad de comprender el fenómeno de la fuga de clientes en una compañía de telecomunicaciones. El problema planteado puede ser crítico para muchas compañías, ya que es más complejo atraer a nuevos clientes que retener a los existentes. Para abordar dicho problema, se hace uso de una metodología ágil focalizada en el desarrollo de proyectos Big Data Analytics. El resultado de este proyecto permite identificar a los clientes más propensos a la fuga y, por lo tanto, conocer el público objetivo de las próximas campañas de retención.

4.1. Definición del problema

Es conocido que los activos más valiosos para las organizaciones son sus clientes activos. En las empresas de las telecomunicaciones, los clientes pueden elegir entre varios proveedores y ejercer sus derechos de cambiar de un proveedor de servicios a otro. Este fenómeno, llamado churn, se convierte en un problema financiero muy serio porque generalmente es mucho más costoso atraer a nuevos clientes que retener a los ya existentes. Por lo tanto, la rotación de clientes es importante de gestionar, especialmente en estas industrias caracterizadas por una fuerte competencia, mercados saturados y oportunidades limitadas de crecimiento.

La necesidad de negocio, que se desea solucionar a través de técnicas de aprendizaje automático, es la de comprender el fenómeno churn y reducir los niveles de fuga voluntaria de los clientes. La realización de este proyecto permite a la organización tomar decisiones de negocio basadas en los datos, y así, realizar campañas de retención específicas a los clientes. Estas acciones permiten desarrollar una nueva relación personal con los clientes hasta ahora desconocida.

4.2. Metodología empleada

Los principios de las metodologías ágiles se centran en entregar resultados lo antes posible, reduciendo, así, la exposición al riesgo a medida que el proyecto avanza. Al entregar resultados en pequeños incrementos de trabajo (sprints), todos los requisitos, desarrollos, modelos y resultados se validan a la finalización de cada sprint. Estas validaciones no son el único beneficio que aportan estas metodologías, sino que también permiten mejorar los resultados tras el feedback recibido, permitiendo así asegurar el éxito del proyecto. Las principales fases de la metodología se mencionan a continuación:

- **Business Understanding:** Al comienzo de cada proyecto de analítica avanzada, antes de comenzar el tratamiento de datos y la generación de los modelos analíticos, se debe definir claramente la necesidad del área de negocio. Los sponsors del proyecto desempeñan el papel más crítico en esta fase de la metodología, ya que ellos definen el problema, establecen los objetivos del proyecto y proporcionan los requisitos de la solución desde una perspectiva empresarial. Para ayudar a garantizar el éxito del proyecto, los sponsors deben estar involucrados y participar en todo el proyecto para aportar experiencia sobre el dominio, revisar los resultados intermedios y garantizar el desarrollo del proyecto.

- **Analytical Planning:** Una vez realizada la comprensión de las necesidades de negocio, el siguiente paso es definir un planteamiento analítico y diseñar un plan inicial en base a la información adquirida. Esta fase requiere expresar el problema con técnicas estadísticas y de aprendizaje automático. Además, es necesario identificar los datos necesarios, el volumen, el tipo y el histórico requeridos para resolver el problema. Gracias al diseño de este esquema inicial, es posible obtener una rápida comprensión de todos los posibles factores influyentes y cuellos de botella que pueden afectar más adelante al proyecto.

- **Infrastructure setup:** En la siguiente fase de la metodología se despliega la arquitectura de Big Data que satisface las demandas de negocio y técnicas

ya establecidas. Esta plataforma consta de servicios que abordan todo el ciclo de administración del dato. En concreto, esta plataforma posibilita recoger datos en crudo, almacenar la información en un repositorio seguro, escalable y duradero, transformar los datos a un formato y finalmente, visualizar insights procedentes de los resultados. La infraestructura configurada debe asegurar el concepto de la localidad de datos, en el que las tareas de procesamiento se realizan cerca de la ubicación de datos, evitando la transferencia de estos que es un cuello de botella en el procesamiento a gran escala.

- **Data Acquisition:** El objetivo de la siguiente fase es obtener los datos estructurados, no estructurados y semiestructurados que son útiles para resolver el problema planteado. Estos datos, que pueden proceder de fuentes internas y externas a la organización, deben ingestarse en la plataforma Big Data para su posterior análisis. Concretamente, los datos se ingestan en el Data Lake que permite consolidar todos los datos de la organización en una única visión global, evitando así el problema de los silos de información. Para este proceso de ingesta, se utiliza el término ELT (Extract, Load, Transform) en lugar del tradicional ETL (Extract, Transform, Load). En este nuevo enfoque los datos se extraen sin procesar, se cargan en el Data Lake y se transforman para un objetivo concreto.

- **Data Comprehension:** Antes de continuar con la preparación de datos para el modelado, es necesario entender brevemente la información recopilada y tratar de identificar posibles problemas que puedan surgir. La fase de comprensión de datos comienza con una exploración para familiarizarse con las variables del estudio. Las estadísticas descriptivas y las técnicas de visualización se utilizan para comprender los valores iniciales, descubrir los primeros insights y evaluar la calidad de los datos, entre otros. Tras identificar las tipologías de todas las variables, se realizan análisis univariante y multivariante sobre las variables.

Los métodos seleccionados para realizar el análisis univariante dependen de la tipología de la variable estudiada. En el caso de variables continuas, las medidas de centralidad y dispersión deben analizarse utilizando diagramas de caja, histogramas y gráficos de densidad, entre otros; mientras que, para las variables categóricas, las tablas de frecuencia y los gráficos de barras son utilizados para entender la distribución de cada categoría. El análisis univariante sirve para lograr una primera comprensión de los patrones en cada una de las variables y ayuda a desarrollar bocetos mentales de lo que los datos pueden mostrar en un análisis más detallado.

Por otro lado, el análisis multivariante estudia la relación entre variables. En el caso de dos variables continuas, los gráficos de dispersión y las matrices de correlación analizan la existencia de relación entre variables y la potencia de dicha relación. Para las variables categóricas se utilizan matrices bidimensionales, gráficos de barras apiladas y test de chi-cuadrado, entre otros. Finalmente, los gráficos de caja, junto con otras técnicas, pueden ser usados para explorar la relación entre variables categóricas y continuas.

- **Data Preparation:** Las fases de comprensión y preparación de los datos son las que consumen más tiempo en un proyecto, ya que tienen una importancia crucial sobre el resultado final. Esta fase cubre las tareas relacionadas con la transformación de datos y la ingeniería de variables. La primera de ellas se centra en transformar los datos en variables más útiles utilizando técnicas como agrupación, normalización, funciones logarítmicas y muchas otras. Estas técnicas se usan cuando es necesario cambiar la escala de una variable, estandarizar los valores de una variable, transformar relaciones no lineales en lineales o modificar de distribuciones sesgadas en simétricas, entre otras. En cambio, la ingeniería de variables genera nuevas variables predictoras derivadas de la información ya existente, pero aportando mucho conocimiento del dominio del problema.

Esta tarea se puede considerar como el arte de extraer información sobre los datos existentes con el objetivo de que esas nuevas variables creadas tengan un impacto notable en el poder de la predicción. Otras técnicas como la reducción de la dimensionalidad y la selección de variables se utilizan para producir el mejor subconjunto mínimo de variables no correlacionadas que mejor explica la varianza de los datos. Como resultado de las dos tareas descritas, se genera el conjunto de datos final que alimenta la fase de modelado analítico.

- **Model Planning:** Durante la fase de planificación del modelo se deben definir los métodos, técnicas y flujos para su construcción. Normalmente, existen varios enfoques para resolver el mismo problema analítico, así que antes de decidir la estrategia seleccionada es necesario analizar si el conjunto de datos cumple con los requisitos de los algoritmos y si los algoritmos cumplen con las restricciones de negocio. En esta fase también se definen las métricas y métodos empleados en la evaluación de los modelos.

- **Model Building:** Esta fase se focaliza en el desarrollo de los modelos según la estrategia definida previamente. Durante el proceso de construcción se recomienda generar una primera versión rápida del modelo para que sus resultados puedan ser utilizados de referencia para las siguientes iteraciones. El feedback de los sponsors, junto con la incorporación de nueva información y nuevas técnicas, será útil para refinar el modelo durante los siguientes sprints. Para el aprendizaje supervisado, los algoritmos ensamblados suelen construir modelos estables y de gran precisión. La idea tras estos algoritmos es fusionar múltiples modelos débiles para producir un modelo fuerte. Por otro lado, los algoritmos de clustering probabilístico son recomendados para el aprendizaje no supervisado. Estos algoritmos proporcionan para cada elemento una probabilidad de pertenencia a cada clúster, aportando mucho valor desde el punto de vista de negocio.

- **Model Evaluation:** La evaluación del desempeño de los modelos es un paso clave en cualquier problema de aprendizaje supervisado, ya que su objetivo es identificar como de buenas son las predicciones para los nuevos datos no observados. Durante esta fase se obtiene la matriz de confusión y a partir de ella, se obtienen las métricas de evaluación. El aprendizaje no supervisado también requiere la evaluación de sus resultados. Para ello, se utilizan un conjunto de métricas de validez internas y externas. Las métricas internas se basan en la información intrínseca del conjunto de datos, mientras que las métricas externas requieren de información previa sobre el conjunto de datos para verificar la calidad de los resultados del modelo.

- **Results communication:** Una vez que el modelo desarrollado ha cumplido su objetivo desde las perspectivas analítica y de negocio, el siguiente paso es interpretar formalmente sus resultados a los sponsors de negocio. En este contexto, la visualización de datos es una técnica muy poderosa para transmitir los insights de valor aportados y generar resultados accionables. Las reuniones de presentación de resultados deben estar lideradas por perfiles próximos a negocio y con conocimientos técnicos.

- **Deployment:** Cualquier proyecto de analítica avanzada tiene el propósito de convertir los resultados de sus modelos en acciones de negocio. Una vez que el modelo desarrollado ha sido aprobado por los sponsors, se procede a su implementación en el entorno de producción. Generalmente, la puesta en producción se realiza parcialmente hasta que su rendimiento haya sido evaluado por completo. Dependiendo de los requerimientos, la fase de implementación puede ser tan simple como generar un informe periódico con recomendaciones o un dashboard interactivo, o tan difícil como su integración en workflows complejos, o el desarrollo de nuevos procesos específicos.

- **Documentation:** Esta fase de la metodología se centra en compartir el código desarrollado para las ingestas de datos, el tratamiento de las variables, la construcción del dataset, la generación de los modelos analíticos, los informes finales del proyecto y toda la documentación técnica relevante. Toda la documentación aportada hace que todo el proceso sea reproducible y compatible con desarrollos posteriores. Una forma muy común de compartir esta información es a través del repositorio de código disponible dentro de la organización.

- **Feedback:** Y, por último, pero no menos importante, esta etapa puede aportar mucho valor cuando se realiza dentro del proceso iterativo. Los modelos desarrollados durante las fases previas no deben implantarse en los sistemas de producción y olvidarse de ellos, sino que deben adaptarse continuamente a las nuevas necesidades utilizando el feedback de la organización. En este contexto, tras la recoger los resultados del modelo en el entorno de producción, los sponsors proporcionan un importante feedback con el que es posible refinar y volver a implementar el modelo. Esta nueva iteración puede aumentar la precisión del modelo, proporcionando así un valor adicional a la organización

4.3. Experimentos y resultados

Construir un mecanismo de predicción capaz de monitorear el abandono de los clientes es un paso importante para el desarrollo de las empresas. Esta necesidad de negocio se resuelve utilizando un modelo analítico que explota la información detectada en los comportamientos de los clientes fugados. En particular, este proyecto se aborda como un problema de clasificación supervisada mediante el uso de información relacionada con los meses anteriores de la fuga del cliente. El modelo propuesto es capaz de proporcionar una puntuación de abandono para cada usuario. Gracias a este score, los usuarios con mayor puntuación será los más propensos a la fuga y, por lo tanto, el objetivo de las próximas campañas de retención.

Con el objetivo de desarrollar el anterior experimento, es necesario el despliegue de una plataforma Apache Hadoop. Esta plataforma debe contar con componentes como Sqoop, Hive y Spark, para diferentes propósitos. Apache Sqoop es utilizado para realizar la ingesta de la información al Data Lake. Este componente permite importar grandes volúmenes de datos entre las bases de datos de origen y el sistema de archivos distribuidos HDFS de la plataforma Hadoop. Por otro lado, Apache Hive es el componente utilizado para el tratamiento de los datos, mientras que Apache Spark es el componente utilizado para la construcción de los modelos analíticos sobre la plataforma Hadoop.

Tras el despliegue de la plataforma Big Data, se realiza la ingesta de las fuentes de datos relacionadas con el tráfico de voz, mensajes y datos móviles, los planes contratados, la información del dispositivo y variables sociodemográficas, entre otras. A partir de estas fuentes, se genera un conjunto de variables predictoras, en cada mes, tales como número de minutos de uso (entrada y salida), número de sms (entrada y salida), número de megabytes de uso (ascendente y descendente), porcentaje de cobertura 4G/3G/2G en el hogar/trabajo, número de llamadas completadas/incompletas (entrada y salida), número de llamadas onnet/offnet (entrada y salida) y número de llamadas desde el contact center de la competencia, entre otras.

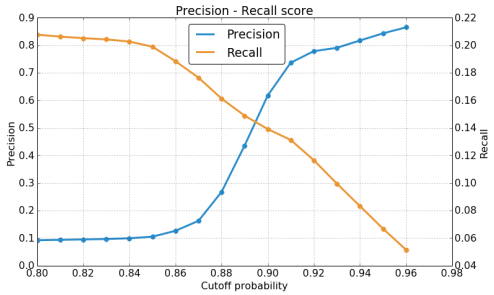
El conjunto de datos inicial está compuesto por más de 15 millones de clientes distintos y cientos de variables predictoras. Tras la aplicación de algunos filtros relacionados con la profundidad histórica de los datos, el porcentaje de valores vacíos o inválidos y la detección de valores atípicos, un total de 9 millones de clientes con más de 300 variables predictoras asociadas son finalmente analizados.

Debido al excesivo número de variables predictoras, la selección de variables juega un pale importante en este problema. Varias técnicas como random forest, boruta y mRMR han sido probadas sobre el conjunto de datos. El método que proporcionó resultados más interesantes fue mRMR cuyo objetivo es seleccionar variables que tengan redundancia mínima y máxima relevancia. Finalmente, las variables que fueron seleccionadas para construir el conjunto de datos final fueron: la edad del cliente, la antigüedad del cliente, el plan contratado, los minutos empleados, las llamadas incompletas e interrumpidas, los megabytes utilizados, el exceso de megabytes contratados, la proporción de cobertura 3G capturada, y la posibilidad de disponer de cobertura 4G en el chip, en el dispositivo y en el área.

Con el objetivo de entrenar el modelo, el conjunto de datos se divide de tal manera que, el 70% de los clientes fugados se asocian al conjunto de entrenamiento y el 30% restante al conjunto de test. Como es de esperar en este tipo de problemas, la clase minoritaria está muy desbalanceada (0,4% clientes fugados) respecto a la mayoritaria. Tras aplicar técnicas de undersampling, el conjunto de entrenamiento consigue una proporción más equilibrada, un cliente fugado por cada tres clientes no fugados. En cambio, en el conjunto de test se mantiene con la proporción real de clases para obtener métricas honestas.

Después de configurar los conjuntos de entrenamiento y de test, se entrenan varios modelos con los algoritmos decision trees, gradient boosted trees y random forest, pero es este último el seleccionado por sus buenos resultados. En concreto, la configuración de parámetros seleccionada para este problema es 300 árboles, con una profundidad de 8 niveles y utilizando la entropía como la métrica de división del árbol en los siguientes subniveles. El siguiente gráfico

muestra los principales resultados obtenidos por el modelo. Concretamente, refleja la relación existente entre las métricas precision y recall calculadas sobre los clientes del conjunto de test con mayor puntuación. Se puede observar que a medida que aumenta una métrica, disminuye la otra.



Estableciendo un punto de corte en la probabilidad proporcionada por el modelo, por ejemplo 0.95, se puede observar en la siguiente tabla que, de los 974 clientes con esa probabilidad o superior, 821 son clientes fugados, por lo que la métrica precisión para ese subconjunto de clientes es del 84%. A medida que el punto de corte disminuye, se incluyen más falsos positivos en el conjunto de clientes seleccionados, lo que hace disminuir la métrica precisión.

Probability	Users	Churners	Precision
>0.95	974	821	0.8429
>0.93	1547	1223	0.7906
>0.90	2771	1710	0.6171