TRABAJO DE FIN DE GRADO

Grado en Estadística

Título: Machine learning mediante Microsoft Azure: una aplicación sobre real-state.

Autor: Daniel Montoya Guirado

Director: Salvador Torra

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Resumen y palabras clave

Resumen

Este trabajo explica de forma resumida el funcionamiento de la plataforma "Microsoft Azure". Como veremos, se trata de una plataforma muy potente con diversas aplicaciones donde la característica principal es que todos sus servicios se encuentran en la nube, por lo que no es necesario ningún tipo de infraestructura previa para llevar a cabo proyectos. En concreto, el trabajo se centra en crear varios modelos de machine learning con la finalidad de explicar el funcionamiento del software.

Analizaremos, mediante el caso empírico basado en una inmobiliaria de estados unidos, como sería implementar mediante la aplicación del servicio "Azure Machine Learning" un problema real para predecir y clasificar el valor de las casas en función de sus características. Y veremos si los resultados obtenidos son factibles contrastándolos con otros métodos de clasificación y predicción.

Palabras clave:

Machine Learning, Microsoft Azure, Predicción, Big data, Nube.

Abstract:

The following report is about "Microsoft Azure" platform. As we will see, it is a very powerful software with much applications for the business. The main feature is that has all services in the cloud. That is a big advantage because it is not necessary any previous infrastructure to carry out projects. We will focuses on creating several models of machine learning in order to explain the operation with the software.

We will analyze, through one empirical case based on a real-state agency located in city of Ames, United States of America. We will implement through Azure Machine Learning module, a real problem about predict and classify the value of houses according to their characteristics. And we will see if the results obtained are feasible by contrasting them with other methods of classification and prediction.

Keywords:

Machine Learning, Microsoft Azure, Prediction, Big Data, Cloud.

Clasificación AMS (American Mathematical Society)

62-07: Data analysis

97P30 System software.

Índice

1. Introducción

1.1 Concepto "Big Data"

Hoy en día, el mundo genera más de 2.5 trillones[1] de bytes de datos diariamente y se predice que en el 2020 el universo digital podrá llegar a albergar 40 ZB (10^{21} bytes). Basándonos en esta premisa, podemos sacar multitud de conclusiones. Desde un punto de vista estadístico, estos datos son oportunidades de aprender de la población que los genera. Saber manejar la información que los datos nos proporcionan nos da una herramienta muy útil para poder predecir comportamientos de la sociedad o población de la que obtenemos las muestras.

La gran cantidad de datos que se genera cada día hace que sea cada vez más necesarias herramientas más potentes para poder tratar con grandes volúmenes de información. Esto es un problema que afecta a escala mundial y en todos los ámbitos, ya que estos datos debidamente analizados pueden servir para mejorar procesos, adelantar comportamientos, personalizar campañas de publicidad un largo etcétera de aplicaciones.

Figura 1. Infografía sobre el Big Data.
Gay Terán, M. (2017) "Big data": Optimizando los procesos de selección del personal [infografía]. Recuperado de: https://confilegal.com/20170228-estrategia-reptiliana/

Para llevar a cabo un exitoso proyecto de Big Data, es necesario facilitar al máximo que los usuarios puedan ir "experimentando" con la información. Por lo tanto, debe ser de alta importancia el desarrollo de una interfaz sencilla, donde se presenten los datos de manera que los usuarios no necesiten grandes conocimientos científicos o tecnológicos para probar sus hipótesis. Debido a que en general, se esperan resultados rápidos o instantáneos de las aplicaciones que desarrollan Big data, a pesar de que las consultas puedan ser muy complejas y abarquen millones de registros de datos, nunca se debería

[1] Datos obtenidos de la entrada del diario ABC con fecha 23/02/2018

olvidar que para ello es necesario un gran gasto computacional que no todos los equipos disponen.

Con la llegada de las nuevas tecnologías, las empresas se han modernizado con el objetivo de agilizar el trabajo en la administración de los departamentos y reducir costes de infraestructuras. Por lo que no tardó en llegar la idea de utilizar la denominada "nube" para almacenar la gran cantidad de datos generada por las empresas.

1.2 Concepto "Nube"

La nube, es un concepto proveniente del inglés y significa "Cloud Computing", es el nombre que se le dio al procesamiento y almacenamiento masivo de datos en servidores que alojen la información del usuario. Es decir, son capaces de guardar externamente (en la red) la información y los archivos de los usuarios.

Este concepto se ha convertido en algo habitual y le damos un uso diario ya que los servicios que hacen uso de esta tecnología son muy fáciles de utilizar, la mayoría de usuarios no son conscientes que por ejemplo, todas las aplicaciones de google se gestionan a través de una "nube" en la que todos los datos quedan almacenados.

Una de las principales ventajas de la nube es el acceso desde cualquier lugar y cualquier momento a tus datos, y también hay aplicaciones como Evernote[2] que además de permitir guardar imágenes, utilizan el procesamiento de los servidores de Google para hacer funcionar sus programas y ahorrar coste computacional de nuestro PC.

1.3 Beneficios de la "nube"

En un nivel profesional, podemos ver las ventajas que proporciona manera de trabajar:

1. Fin del almacenamiento local: Con la nube podemos almacenar toda la información de la empresa de forma externa para poder tenerla accesible desde cualquier lugar.
2. Protección de datos: El usuario que podrá tener acceso a alterar y editar un determinado documento será definido mediante una contraseña, por este motivo se puede proteger los datos que se almacenan en la red.
3. Disminución de los costos de almacenamiento e infraestructura: Mediante nubes privadas y públicas será posible realizar una combinación para construir una nube híbrida. De esta forma se puede dividir la información confidencial de los archivos a los que puede tener acceso el personal de la empresa.
4. Comportamiento de la información sencillo: Como los datos se encuentran en la red, son accesibles desde cualquier parte del mundo, por lo que resultará mucho más sencillo actualizar o modificar la información y hacerlo a tiempo real, para que esté disponible a los demás usuarios con acceso.

[2] https://www.evernote.com/Login.action

5. Capacidad de almacenamiento flexible: La nube permite aumentar o disminuir el espacio destinado al almacenamiento, todo dependerá de la tarifa que contratemos y cantidad de archivos que queramos tener en la red.

A pesar de todas estas ventajas, todavía hay empresas que no se han decantado a hacer la migración a la nube como almacenamiento para sus datos ya que es necesario invertir en planeamiento, esfuerzo, recursos y tiempo. El cambio a la nube debe ser visto como una oportunidad para dejar atrás sistemas pesados por versiones rápidas y económicas.

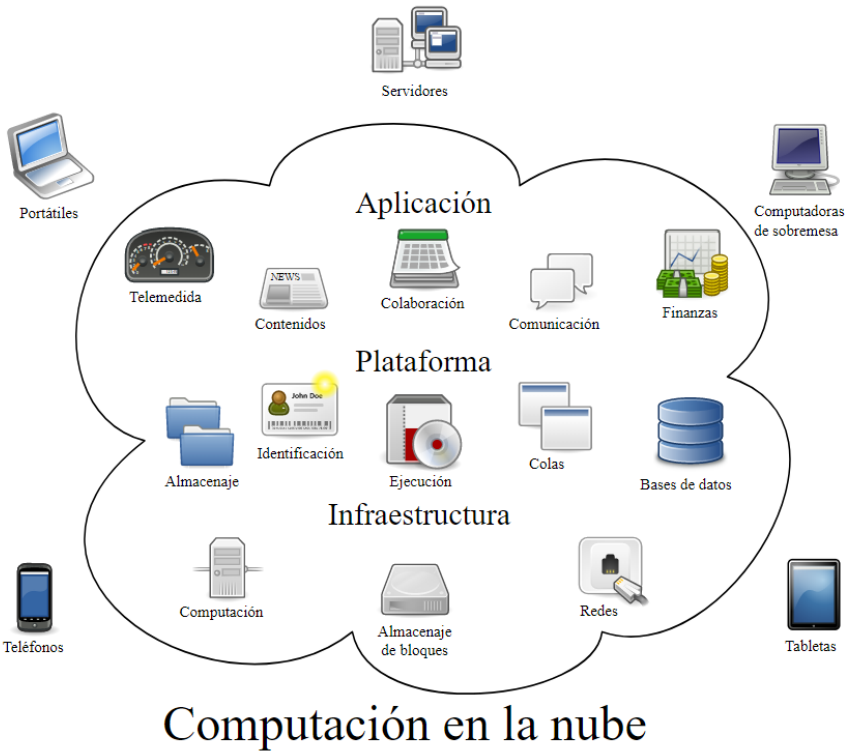


Figura 2. Computación en la nube. Wikipedia (2018). Recuperado de: <https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube>

1.4 Como ha afectado este fenómeno al Big Data

En un sector basado en el tratamiento de datos, la nube ha sido toda una revolución. Poder almacenar los millones de bytes que se generan sin necesidad de infraestructura (ni software ni hardware) y además poder tratarlos para conseguir mejoras para los usuarios o empresas dedicadas a analizar grandes volúmenes de datos. También la nube proporciona mayor rapidez y acceso a la información ya que dependiendo de qué servicio utilicemos no debemos preocuparnos de molestas actualizaciones. Además estos servicios están accesibles desde cualquier ubicación por lo que es muy cómodo para trabajar desde distintos lugares. Por último y quizá lo más relevante, es la seguridad que proporciona la nube en la protección de sus datos, ya que cuenta con varios protocolos y la posibilidad de almacenar copias de seguridad un número ilimitado de veces.

1.5 Machine Learning en la nube

En primer lugar, ya que el término está siendo muy utilizado, definiremos correctamente el significado del término machine learning: se trata de una rama de la inteligencia artificial con el objetivo de desarrollar métodos que permitan a las computadoras aprender por sí solas. De esta forma, en estadística hablamos de machine learning como la capacidad de aprender de los datos recopilados durante el tiempo para poder tratar de predecir acontecimientos futuros.

Para que estos datos recopilados sean manejables, han de ser debidamente procesados, limpiados, filtrados, transformados y modelados. Si el conjunto de datos no es demasiado grande, se pueden tratar, analizar y modelar en un único servidor. Pero cuando el tamaño de la muestra aumenta, se requerirá del uso de varios servidores para que el tiempo de procesado no sea demasiado elevado. Esto no quiere decir que nos olvidemos de los métodos de tratamiento de datos convencionales, ya que las plataformas que trabajan con la nube, permiten implementar una parte de desarrollo de software, como por ejemplo lenguaje de programación en R o Phyton. Por lo que en este *nuevo mundo* también tienen cabida las empresas más tradicionales.

Ahora bien, ¿cómo podemos acceder a la nube?, ¿es muy costosa y difícil la implementación? Preguntas como estas se las hacen grandes empresas que se plantean trasladar sus servicios a la nube. Lo primero que hay que conocer son las tres empresas pioneras en tecnología que proporcionan servicios de Big Data en la nube. Estas empresas son: Amazon Web Services (AWS), Google Cloud y Microsoft Azure.

Figura 3. Amazon Aws vs Microsoft Azure vs Google Cloud, ¿cuál elegir?. Recuperado de <https://www.openinnova.es/amazon-aws-vs-microsoft-azure-vs-google-cloud-cual-elegir>

1.6 ¿Cómo escoger entre las tres grandes compañías que proporcionan servicios en la nube?

Como hemos comentado al comienzo de la introducción, para las empresas implementar la infraestructura física necesaria para realizar las propias operaciones es necesario una cantidad de técnicos, presupuesto y tiempo para realizar pruebas.

En cambio, en la nube se podría evitar gastar mucho presupuesto configurando entornos de desarrollo ya que accediendo a una de las tres plataformas que proporcionan los servicios en la nube, accedes a la infraestructura necesaria para trabajar desde el principio, sin coste ni pruebas. Este concepto no sólo está enfocado a las grandes empresas, sino que cualquier usuario, desde su casa puede acceder a estos servicios de primer nivel.

Esto es posible porque tanto Microsoft Azure como Amazon Aws y Google Cloud ofrecen sus servicios mínimos gratuitos y la cantidad que se paga varía mucho en función de la cantidad, potencia demandada y número de servidores virtuales que se quieren implantar. Y en los tres casos se ofrecen ventajas de pago que en resumen son: No hay costes iniciales ni cargos por cancelación, se paga solo por el uso que se le da a los servicios y la facturación es por minuto.

Ventajas de Azure respecto de AWS y Google Cloud:

- Azure tiene la cobertura de cumplimento normativo más completa respecto a la competencia. Esto indica más seguridad, por lo tanto más confianza en la nube.
- Proporciona coherencia hibrida en el desarrollo de aplicaciones, administración, seguridad y administración de identidades. De esta forma ayuda a la portabilidad de aplicaciones y cargas de trabajo a la nube de Azure.
- Es la plataforma que tiene el mayor número de regiones con servicios en la nube, con más de 50 regiones. Además es la plataforma líder en la nube, con varios reconocimientos por sus soluciones vanguardistas en este sector.
- Azure te permite incorporar cualquier herramienta o lenguaje de desarrollo, permite la incorporación de código abierto a sus soluciones.
- Inteligencia incomparable, Azure presume de una gran capacidad de proceso, basado en GPU que permite agilizar el aprendizaje, hacer simulaciones de alto rendimiento y llevar a cabo análisis de datos en tiempo real.
- Investigación y análisis de negocios mediante soluciones como la previsión de la demanda y optimización de inventario para obtener una ventaja competitiva. Además de tratar con escenarios sobre internet de las cosas como supervisión remota o mantenimiento predictivo mediante la aplicación IOT de Azure.
- Administrar y optimizar el gasto en la nube mediante la herramienta Azure cost management, que permite controlar la asignación de costos, elegir el tamaño de las máquinas virtuales y visualizar el beneficio económico que se va obteniendo.

1.7 Planteamiento del trabajo

En este trabajo se explicará mediante información y un caso empírico el funcionamiento de la plataforma Microsoft Azure con el objetivo de determinar si es una buena herramienta para el análisis, predicción y clasificación de datos.

La estructura que seguiremos será una primera comparativa entre las tres empresas en la nube que proporcionan los servicios de machine learning, posteriormente se explicará el funcionamiento de la plataforma Microsoft Azure, haciendo zoom en las herramientas que necesitamos para la predicción de las bases de datos.

Por último, se detallará como sería la implementación de una solución de machine learning en Azure desde cero, utilizando la versión gratuita del software. Finalmente, sacaremos conclusiones de los resultados obtenidos, veremos si son fiables y como se valoraría una posible implantación comercial de un modelo de machine learning general para la utilización en distintas bases de datos.

2. Metodología

En la realización de este trabajo hemos tenido que tener una sólida base de conocimientos teóricos sobre el tratamiento y modelos de predicción de los datos. En este apartado se explican todos los métodos y modelos que se han utilizado para llevar a cabo el proyecto.

2.1 Pre-procesado de datos

En el apartado de pre-procesado de la base datos, nos encontramos missings que hemos de sustituir por valores artificiales que sean probables y lógicos. Para ello, en la función *clean missing data*, utilizamos el método probabilistic PCA. Seguidamente, con los datos limpios y sin valores perdidos, tendremos que llevar a cabo un análisis de las variables independientes con más potencia predictiva. Para ello utilizaremos la función ***Filter Based Feature selection***, es una función en la cual podemos escoger distintos métodos para conocer las variables que más influyen en la variable dependiente. Los métodos a utilizar son: Correlación de kendall, Spearman y Pearson, método de información mutua, puntuación de fisher y prueba chi-cuadrado. A continuación, entraremos en detalle de todos los métodos utilizados y explicaremos en que consiste cada uno y la forma de interpretarlos.

2.1.1 *Data scrubbing (limpieza de datos)*

2.1.1.1 Probabilistic Principal Component Analysist (Probabilistic PCA):

Esta función reemplaza los valores perdidos mediante el uso de un modelo lineal que analiza las correlaciones entre las variables y estima una aproximación de bajas dimensiones de los datos, a partir de la cual se reconstruye la información completa. La reducción de las dimensiones tiene su base en una forma probabilística del análisis de componentes principales (PCA)

Si comparamos con otras opciones, como por ejemplo la imputación múltiple mediante ecuaciones encadenadas (MICE), la opción basada en las componentes principales tiene la ventaja de no necesitar la aplicación de variables predictoras para cada columna. En cambio, sí que utiliza la aproximación a la covarianza para el conjunto de datos completo. Por lo tanto, podría ofrecer un mejor rendimiento para los conjuntos de datos que tienen valores perdidos en muchas columnas.

Sin embargo tenemos una serie de limitaciones a la hora de implementar este método. La principal limitación es la expansión de columnas categóricas en indicadores numéricos, por lo que calcula una matriz de covarianza de los datos resultantes, que no resulta tan exacta como otros métodos. La otra gran limitación es que no está optimizada para representaciones dispersas, por lo cual los conjuntos de datos con grandes cantidades de columnas o grandes dominios categóricos (decenas de miles) no son compatibles ya que consumirían demasiado espacio.

2.1.2 Selección de variables con más poder predictivo

2.1.2.1 Correlación de Pearson

El coeficiente de correlación de Pearson o r-valor, es una medida de la relación lineal entre dos variables aleatorias numéricas. Devuelve un valor que indica la fuerza de la correlación entre dos variables.

Se calcula tomando la covarianza de dos variables y dividiendo por el producto de sus desviaciones estándar. El coeficiente no se ve afectado por los cambios de escala en las dos variables.

2.1.2.2 Método de información mutua

El valor del método de información mutua mide la contribución o dependencia de una variable para reducir la incertidumbre (entropía[3]) sobre el valor de otra variable. En el caso de selección de variables sobre la respuesta del modelo.

El valor del método de información mutua ($I\left(x_i; y_j\right)$) es muy útil en la selección de variables con potencia predictiva ya que maximiza la dependencia entre la distribución conjunta y la variable objetivo en los conjuntos de datos con muchas dimensiones. Se calcula de la siguiente forma:

$$I(x_i; y_j) = \log \frac{P(x_i|y_j)}{P(x_i)}$$

2.1.2.3 Valor del estadístico de Fisher

El valor del estadístico de Fisher representa la cantidad de información que proporciona una variable sobre algún parámetro del que depende y desconocido.

Se calcula midiendo la varianza entre el valor esperado de la información y el valor observado. Cuando la varianza se minimiza, la información se maximiza, por lo que se puede decir que el valor de Fisher representa la varianza del error.

Este valor se puede utilizar directamente para la selección de variables, seleccionando las variables que obtengan una mayor puntuación discriminante.

2.1.2.4 Prueba Chi-Cuadrado

Se trata de un método estadístico que mide que la proximidad entre los valores esperados de los resultados reales. Supone que las variables son aleatorias y se extraen de una muestra adecuada de variables independientes. El valor del test Chi-cuadrado indica cuanta diferencia existe entre los resultados obtenidos del resultado esperado (aleatorio). De esta forma, para la selección de variables necesitaremos ordenar los resultados de mayor a menor y seleccionar los *K* primeros, ya que se tratarán de las variables más significativas.

[3] Entropía: Cantidad de información que contiene un símbolo. Es una medida de incertidumbre.

2.2 Modelos de regresión

2.2.1 Modelo de regresión lineal

La regresión lineal es quizás el método estadístico más común cuando se quiere llevar a cabo una predicción simple, recientemente ha adoptado el aprendizaje automático y se ha mejorado con métodos nuevos para lograr el ajuste de la línea de regresión. Este método suele funcionar bien en conjuntos de datos escasos y de gran dimensión no demasiado complejos.

Existen dos tipos de regresiones lineales: simple y múltiple. La regresión lineal simple implica simplemente una variable independiente y una dependiente, en cambio la múltiple implica dos o más variables independientes que contribuyen a una sola variable respuesta.

En Azure, el módulo de la regresión lineal admite dos métodos para medir el error y ajustar la línea de regresión: método de mínimos cuadrados ordinarios y descenso de gradiente.

El descenso de gradiente es un método que minimiza la cantidad de errores en cada paso del proceso del modelo. Hay muchas variaciones en el descenso del gradiente. Por ejemplo, si seleccionamos este método podemos establecer una variedad de parámetros para controlar el tamaño del paso, la velocidad de aprendizaje, etc.

El método de mínimos cuadrados ordinarios es una de las técnicas más utilizadas en la regresión lineal. Calcula el error como la suma del cuadrado de la distancia desde el valor real a la línea predicha, y se ajusta al modelo minimizando el error al cuadrado. Este método supone una fuerte relación lineal entre las entradas y la variable dependiente.

2.2.2 Regresión bayesiana

Es un método basado en la regresión lineal, pero partimos de un análisis estadístico realizado a partir de la inferencia bayesiana. Este enfoque bayesiano a menudo se suele contrastar con el enfoque frecuencialista[4].

El enfoque bayesiano utiliza la regresión lineal complementada con información adicional en forma de una distribución de probabilidad previa. La información previa sobre los parámetros se combina con una función de verosimilitud[5] para generar estimaciones para los parámetros. En cambio, el enfoque frecuencialista, viene representado por la regresión lineal de mínimos cuadrados estándar.

La inferencia bayesiana se compone de tres partes: Definir la distribución “a priori” para los parámetros, determinar la verosimilitud de los datos y finalmente aplicar el teorema de Bayes para actualizar la distribución a posteriori.

[4] Número ideal al que converge la frecuencia relativa cuando la frecuencia total tiende a infinito.
[5] Función de los parámetros de un modelo estadístico que permite realizar inferencias sobre su valor a partir de un conjunto de datos.

2.2.3 Decision Forest Regression

Se trata de un modelo de aprendizaje automático utilizado en la predicción de datos que consiste en la construcción de diagramas lógicos que sirven para representar unas condiciones que ocurren de forma sucesiva hasta lograr la resolución del problema. Es un método no paramétrico en el que se recorre una estructura de datos en forma de árbol binario hasta que se alcanza el nodo de decisión. Este modelo está formado por nodos, vectores de números, flechas y etiquetas.

Los nodos son el punto en el que se ha de tomar una decisión entre varias opciones posibles. Si existen muchos nodos, aumenta el número de posibles valores a los que puede acceder un individuo. Los vectores de números son la solución que se obtiene al llegar al final de un árbol de decisión. Las flechas son las uniones que se crean entre un nodo y otro que representa la acción que se lleva a cabo después de una decisión. Y por último, las etiquetas se encargan de dar nombre a cada acción, se encuentran en cada nodo y flecha.

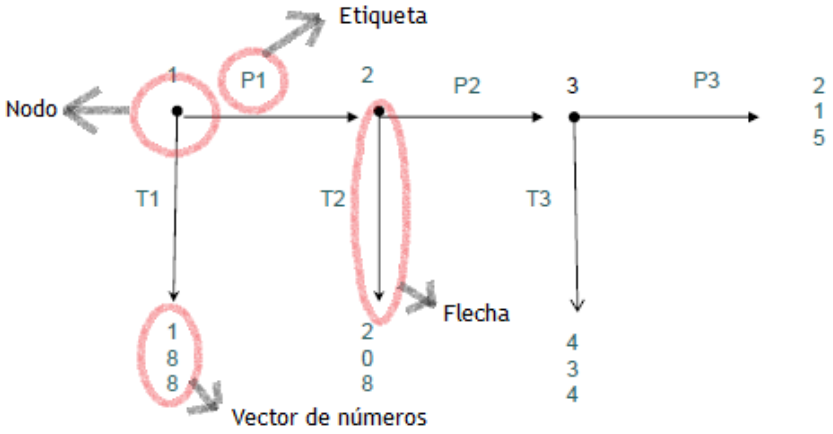


Figura 4: Elementos de un árbol de decisión. Recuperado de <https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n>

Este modelo de predicción consiste en un conjunto de árboles que se encuentran en un bosque de decisión. Cada árbol genera una distribución gaussiana como predicción para las observaciones. Se trata de ir añadiendo árboles al bosque para tratar de ajustar la distribución gaussiana final generada por el conjunto de árboles del bosque de decisión.

2.2.4 Boosted Decision Tree Regression

A diferencia del método anterior, el aprendizaje automático basado en árboles de decisión reforzados consiste en generar una secuencia de árboles donde cada uno se construye a partir de los residuos de predicción del árbol anterior.

Este método se basa en la construcción de árboles binarios (divide los datos en dos muestras dentro de cada nodo) y en cada paso del algoritmo de refuerzo se determina una partición (óptima) de los datos, se calculan las desviaciones estándar de los residuos de cada partición y en el siguiente nodo se ajustará a los residuos previamente generados con la finalidad de encontrar otra partición de datos que reduzca más el error (varianza residual) una vez obtenida una secuencia de árboles.

En Azure Machine Learning, los árboles de decisión reforzados utilizan el algoritmo de mejora de gradiente MART (Multiple additive regression tres). Se trata de una técnica de aprendizaje automático para problemas de regresión, construye cada árbol de regresión de forma escalonada, utilizando una pérdida predefinida. Por lo tanto, el modelo de predicción (reforzado) es un conjunto de modelos de predicción más débiles. Este método genera una serie de árboles de forma escalonada y después selecciona el árbol óptimo utilizando una función de pérdida diferenciable arbitraria.

2.3 Redes Neuronales

Este concepto está basado en intentar imitar el funcionamiento de las redes neuronales de los organismos vivos. Es decir, tratar de conectar un conjunto de neuronas sin que haya asignada una tarea concreta para cada una, sino que ellas mismas van creando y reforzando sus conexiones para tratar de "aprender". Este es el concepto genético de redes neuronales, pero ahora hablamos de redes neuronales a partir de las matemáticas y estadística, que se basa en dados unos parámetros, encontrar la forma de combinarlos para predecir un cierto resultado.

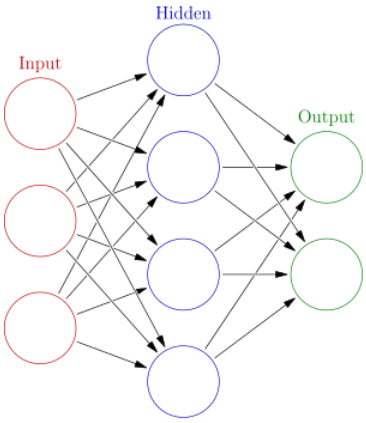


Figura 5. Wikipedia (2018). Redes neuronales. Recuperado de :<https://es.wikipedia.org/Red_neuronal_artificial>

Estos modelos son sistemas que aprenden y se forman a sí mismos, en vez de ser programados explícitamente con unos datos determinados. Se crearon para detectar soluciones o características donde los modelos tradicionales no eran capaces de llegar. Las neuronas suelen tener varias capas y la señal atraviesa de delante hacia atrás, es decir hacia donde se encuentra el siguiente estímulo y en dirección al siguiente nodo.

Las redes neuronales son conocidas por su uso en el aprendizaje profundo y modelan problemas complejos como el usual problema del reconocimiento de imágenes, se adaptan fácilmente a los problemas de regresión. Cualquier clase de modelos estadísticos se puede denominar una red neuronal si usan pesos adaptativos y pueden aproximar las funciones no lineales de sus entradas. Por lo tanto, la regresión de la red neuronal es adecuada para los problemas donde un modelo de regresión más tradicional no puede adaptarse a una solución.

La regresión de la red neuronal es un método de aprendizaje supervisado y, por lo tanto, requiere un conjunto de datos que se etiquetan en función del nodo en que se encuentran. Debido a que un modelo de regresión predice un valor numérico, la columna de etiquetas debe ser un tipo de datos numéricos.

2.4 Cross-validation

Se trata de una técnica para evaluar los resultados de un análisis o modelo estadístico. Este método permite evaluar los resultados de forma que se tiene en cuenta el sobreajuste del modelo. Para ello, supone que hay independencia entre los datos de entrenamiento y de prueba. Se trata de hacer iteraciones calculando la media aritmética de los coeficientes de determinación obtenidos de dividir los datos en varias carpetas, crear un modelo para cada una y analizar los ajustes a los datos reales por separado.

Cada carpeta devuelve un conjunto de estadísticos de precisión que se pueden interpretar por separado y en conjunto, para saber si el modelo está sobreajustado[6] a los datos de entrenamiento.

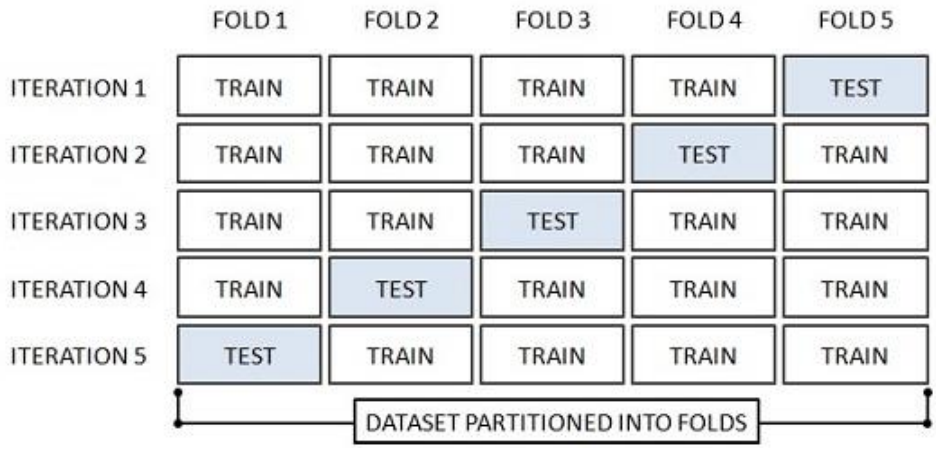


Figura 6: Cross-Validation in machine learning. Recuperado de: <https://www.dummies.com/programming/big-data/data-science/resorting-cross-validation-machine-learning/>

[6] Modelo susceptible a variación en los datos, valido solo para el conjunto de datos de entrenamiento

3. Caso empírico

3.1 Iniciar Azure Machine Learning

Para iniciar el módulo de azure machine learning, debemos dirigirnos directamente a la web del módulo: https://azure.microsoft.com/es-es/services/machine-learning-studio/

Es importante dirigirnos directamente al módulo de machine learning ya que si entramos en la aplicación global (https://azure.microsoft.com) no podremos redirigirnos y seguramente necesitemos pagar para movernos por los diferentes menús. Por tanto, es necesario dirigirnos a la primera dirección web para entrar de forma gratuita y directa.

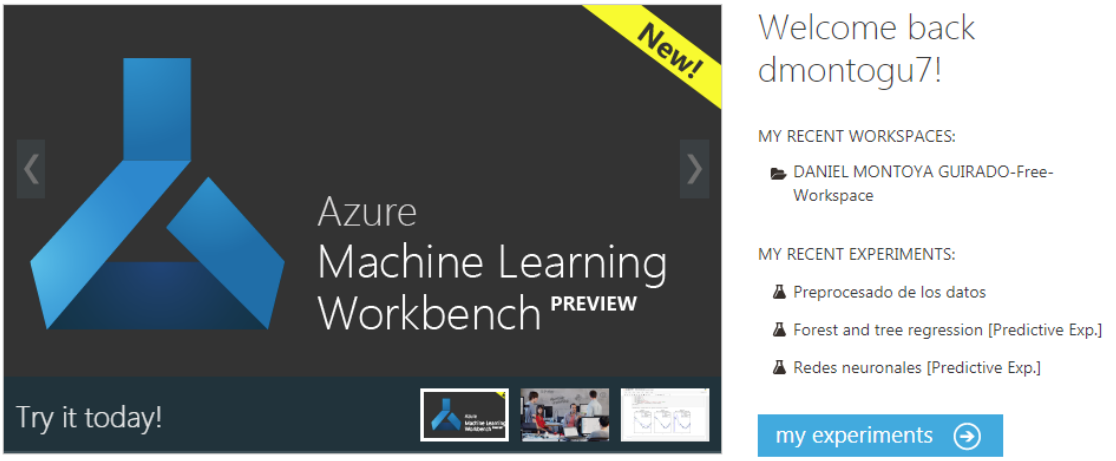
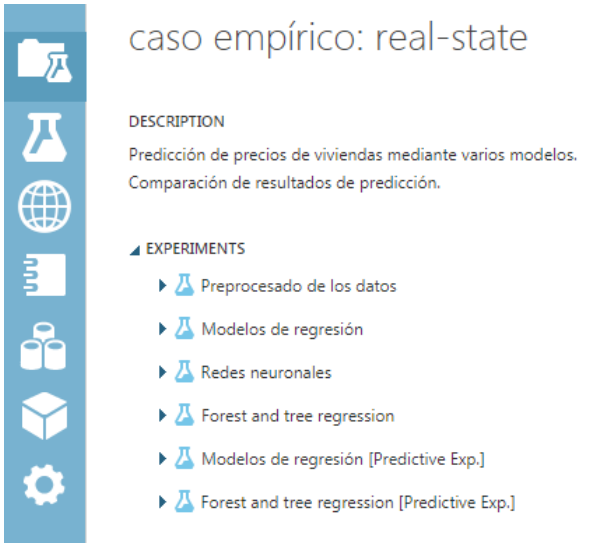


Figura 7. Inicio Azure Machine Learning. Extraido de:
< https://azure.microsoft.com/es-es/services/machine-learning-studio/>

Una vez hemos entrado en la plataforma, nos encontramos con el menú principal, el cual se compone de un menú lateral en el que se encuentran las funciones principales de Azure (en azul). Lo primero que nos aparece al entrar son los experimentos en los que hemos estado trabajando recientemente:

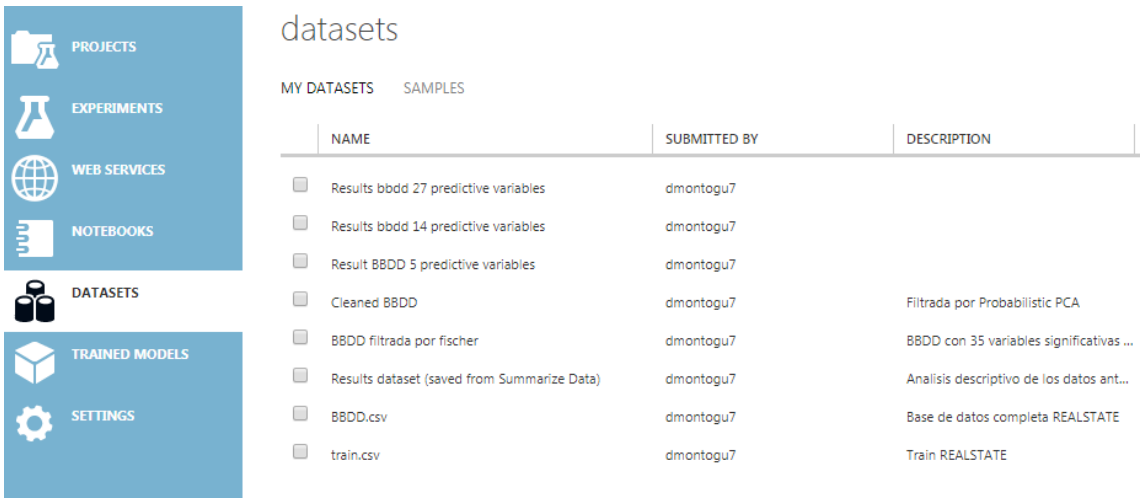


En nuestro caso tenemos cinco experimentos de prueba, uno de pre-procesado de datos y los otros tres son los modelos de machine learning implementados.

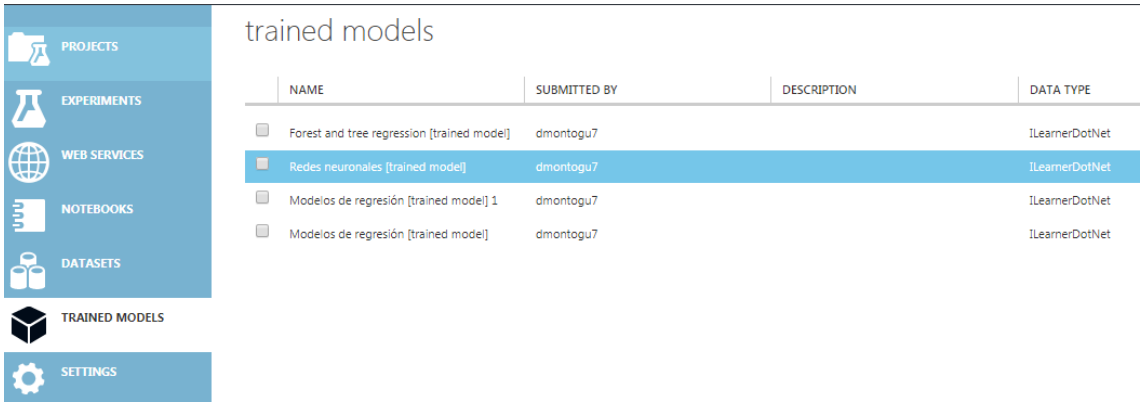
Se han asignado todos los experimentos al proyecto del trabajo: “caso empírico: real-state”, de esta forma podemos tener en un mismo proyecto las bases de datos y modelos relacionados con el mismo proyecto.

Ahora entramos al menú de proyectos. El primer apartado del menú principal, en él podemos ver los proyectos que se han creado. En este caso se muestran dos, uno en el que están asignados los experimentos necesarios para llevar a cabo el trabajo como hemos visto en el apartado anterior. Y en el segundo proyecto están recopilados todos los experimentos necesarios para probar los diferentes parámetros en los modelos hasta conseguir el mejor ajuste a los datos.

El tercer apartado del menú principal: *web services*, se encuentran los modelos que han sido subidos a internet con tal de poder hacer una estimación del precio introduciendo las características de ésta.

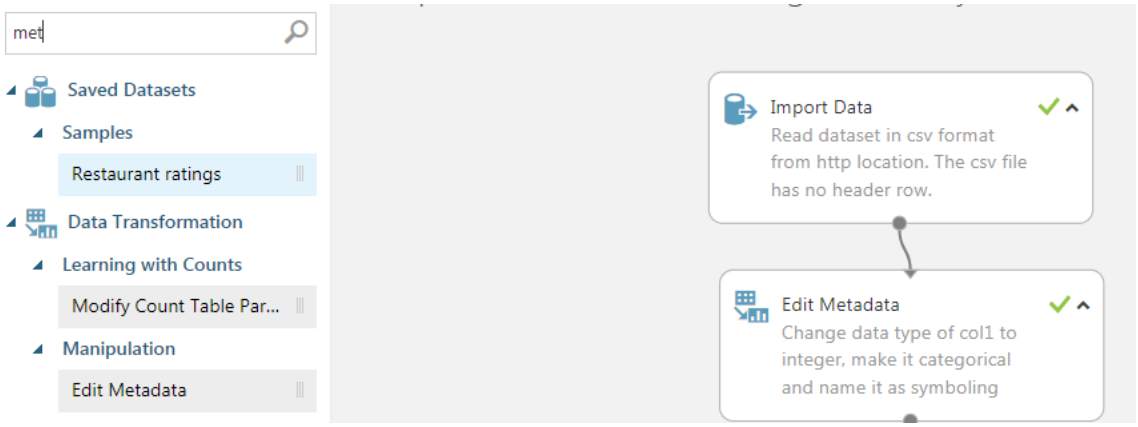


Seguidamente, en el menú nos encontramos la pestaña Notebooks, en élla podemos introducir codigo de R y Phyton y ejecutarlo como si lo hicieramos en el propio programa. En la pestaña datasets vemos todas las bases de datos introducidas o modificadas que hemos guardado en azure.



Finalmente, encontramos la pestaña trained models, aquí podemos encontrar los modelos de machine learning que hemos creado y validado. Y por último las opciones, para modificar aspectos de la interfaz de Azure.

Iniciar un experimento de machine learning es muy sencillo, simplemente tenemos que buscar la función que deseamos y arrastrarla hasta el *workspace* dentro de un experimento. Luego podemos ir uniendo estas funciones con flechas y modificando los parámetros de dichas funciones para conseguir los resultados que esperamos:



Además, cada módulo disponible, tiene una ayuda asociada donde se explica la utilización correcta, que imputs tienen que entrar y que es lo que devuelve la función. Además de un poco de teoría sobre lo que se está haciendo.

Es una ayuda muy completa que proporciona información de calidad y rápida de utilizar, ya que solo picando en el botón “read more” de debajo de los parámetros se abrirá el explorador de internet con la página que hace referencia al módulo seleccionado.

3.2 Base de datos

3.2.1 Contexto

La Base de Datos fue extraída de la web *Kaggle*, se trata de una BBDD sobre real-state. Se trata de unos datos sobre una inmobiliaria de la ciudad de Ames, situada en el condado de Story, estado de Iowa, Estados Unidos. Según el censo de 2000 tenía una población de 50.731 habitantes.

Nos encontramos una base de datos con 81 variables y 1460 viviendas de la ciudad. Trabajaremos con 79 variables explicativas (excluimos la variable ID) para predecir el precio final de la vivienda (en dólares).

A continuación, una lista con todas las variables explicativas y la variable respuesta con una breve descripción de cada una (en el anexo se amplía esta información):

MSSubClass: Clase de construcción
MSZoning: la clasificación general de zonificación
LotFrontage: Pies lineales de la calle conectados a la propiedad
LotArea: Tamaño del lote en pies cuadrados
Calle: Tipo de acceso por carretera
Callejón: tipo de acceso a callejones
LotShape: forma general de la propiedad
LandContour: Planitud de la propiedad
Utilidades: Tipo de utilidades disponibles
LotConfig: configuración del lote
LandSlope: Pendiente de la propiedad
Vecindario: ubicaciones físicas dentro de los límites de la ciudad de Ames
Condición 1: proximidad a la carretera principal o ferrocarril
Condición2: proximidad a la carretera principal o ferrocarril (si hay un segundo presente)
BldgType: tipo de vivienda
HouseStyle: estilo de la vivienda
OverallQual: material general y calidad de acabado
OverallCond: calificación de la condición general
Año de construcción: fecha de construcción original
YearRemodAdd: fecha de remodelación
RoofStyle: tipo de techo
RoofMatl: material de techo
Exterior1: revestimiento exterior en la casa
Exterior2nd: Cubierta exterior en la casa (si hay más de un material)
MasVnrType: Tipo de chapa de mampostería
MasVnrArea: Área de chapa de la mampostería en pies cuadrados
ExterQual: calidad del material exterior
ExterCond: estado actual del material en el exterior
Fundación: tipo de fundación
BsmtQual: Altura del sótano
BsmtCond: estado general del sótano
BsmtExposure: muros de sótano a ras de suelo o de jardín
BsmtFinType1: Calidad del área acabada del sótano
BsmtFinSF1: Tipo 1 pies cuadrados terminados
BsmtFinType2: calificación del área de sótano terminado (si hay múltiples tipos)

BsmtFinSF2: Tipo 2 pies cuadrados terminados
BsmtUnfSF: Pies cuadrados sin terminar del área del sótano
TotalBsmtSF: pies cuadrados totales del área del sótano
Calefacción: tipo de calefacción
HeatingQC: Calidad y condición de la calefacción
CentralAir: Aire acondicionado central
Eléctrico: sistema eléctrico
1stFlrSF: primer piso pies cuadrados
2ndFlrSF: segundo piso pies cuadrados
LowQualFinSF: Pies cuadrados terminados de baja calidad (todos los pisos)
GrLivArea: pies cuadrados del área habitable sobre el nivel del suelo
BsmtFullBath: baños completos en el sótano
BsmtHalfBath: medio baño en el sótano
FullBath: baños completos por encima del nivel de tierra
HalfBath: medio baño por encima del nivel de tierra
Dormitorio: Número de habitaciones sobre el nivel del sótano
Cocina: Número de cocinas
KitchenQual: calidad de la cocina
TotRmsAbvGrd: Total de habitaciones por encima del grado (no incluye baños)
Funcional: calificación de la funcionalidad del hogar
Chimeneas: cantidad de chimeneas
FireplaceQu: calidad de la chimenea
GarageType: ubicación del garaje
GarageYrBlt: año de garaje en el que fue construido
GarageFinish: acabado interior del garaje
GarageCars: tamaño del garaje en la capacidad del automóvil
GarageArea: Tamaño del garaje en pies cuadrados
GarageQual: calidad de garaje
GarageCond: condición de garaje
PavedDrive: calzada pavimentada
WoodDeckSF: área de cubierta de madera en pies cuadrados
OpenPorchSF: área de porche abierto en pies cuadrados
EnclosedPorch: área de porche cerrado en pies cuadrados
3SsnPorch: área del porche de tres estaciones en pies cuadrados
ScreenPorch: área del porche de la pantalla en pies cuadrados
PoolArea: área de la piscina en pies cuadrados
PoolQC: calidad de la piscina
Valla: calidad del cercado
MiscFeature: característica miscelánea no cubierta en otras categorías
MiscVal: $ Valor de la función miscelánea
MoSold: Mes vendido
YrSold: Año de venta
SaleType: Tipo de venta
SaleCondition: Condiciones de venta
SalePrice: Precio de venta (Variable respuesta)

3.3 Análisis descriptivo

Seguidamente se muestra un primer análisis descriptivo de las variables numéricas. Destacar que las variables de medida de área se encuentran en pies cuadrados, posteriormente en el pre-procesado se cambiará a metros cuadrados para lograr una mejor comprensión de los datos.

3.3.1 Tabla descriptiva variables numéricas

Feature	Count	Unique Value Count	Missing Value Count	Min	Max	Mean	Mean Deviation
MSSubClass	1460	15	0	20	190	56,90	31,28
LotArea	1460	1073	0	1300	215245	10516,83	3758,81
OverallQual	1460	10	0	1	10	6,10	1,10
OverallCond	1460	9	0	1	9	5,58	0,89
YearBuilt	1460	112	0	1872	2010	1971,27	25,07
YearRemodAdd	1460	61	0	1950	2010	1984,87	18,62
BsmtFinSF1	1460	637	0	0	5644	443,64	367,37
BsmtFinSF2	1460	144	0	0	1474	46,55	82,54
BsmtUnfSF	1460	780	0	0	2336	567,24	353,28
TotalBsmtSF	1460	721	0	0	6110	1057,43	321,28
1stFlrSF	1460	753	0	334	4692	1162,63	300,58
2ndFlrSF	1460	417	0	0	2065	346,99	396,48
LowQualFinSF	1460	24	0	0	572	5,84	11,48
GrLivArea	1460	861	0	334	5642	1515,46	397,32
BsmtFullBath	1460	4	0	0	3	0,43	0,50
BsmtHalfBath	1460	3	0	0	2	0,06	0,11
FullBath	1460	4	0	0	3	1,57	0,52
HalfBath	1460	3	0	0	2	0,38	0,48
BedroomAbvGr	1460	8	0	0	8	2,87	0,58
KitchenAbvGr	1460	4	0	0	3	1,05	0,09
TotRmsAbvGrd	1460	12	0	2	14	6,52	1,28
Fireplaces	1460	4	0	0	3	0,61	0,58
GarageCars	1460	5	0	0	4	1,77	0,58
GarageArea	1460	441	0	0	1418	472,98	160,02
WoodDeckSF	1460	274	0	0	857	94,24	102,00
OpenPorchSF	1460	202	0	0	547	46,66	47,68
EnclosedPorch	1460	120	0	0	552	21,95	37,66
3SsnPorch	1460	20	0	0	508	3,41	6,71
ScreenPorch	1460	76	0	0	480	15,06	27,73

PoolArea	1460	8	0	0	738	2,76	5,49
MiscVal	1460	21	0	0	15500	43,49	83,88
MoSold	1460	12	0	1	12	6,32	2,14
YrSold	1460	5	0	2006	2010	2007,82	1,15
SalePrice	1460	663	0	34900	755000	180921,20	57434,77

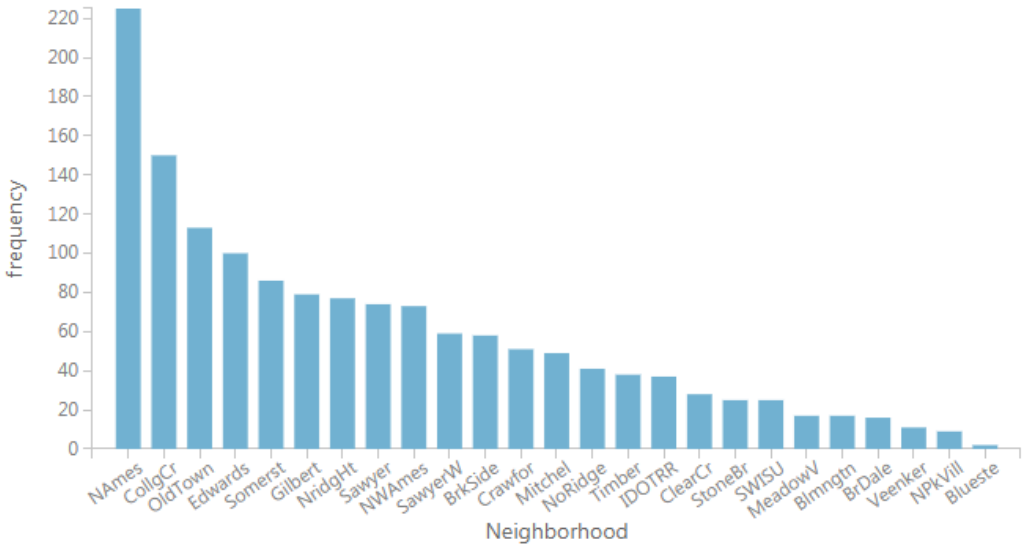
Como podemos ver en la tabla anterior, las variables numéricas no tienen ningún valor faltante. Además, la mayoría de variables tienen el valor mínimo en cero debido a que se tratan de unidades de medida (metros cuadrados), y por lo tanto se tratan de viviendas sin esa característica.

Otro valor llamativo es el máximo de las variables que hacen referencia a baños, habitaciones, extintores y coches en el garaje (*BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageCars*) ya que los valores son inferiores a 8, para estos casos en los que las variables no son puramente numéricas las categorizaremos en valores de uno en uno hasta el valor máximo de cada variable.

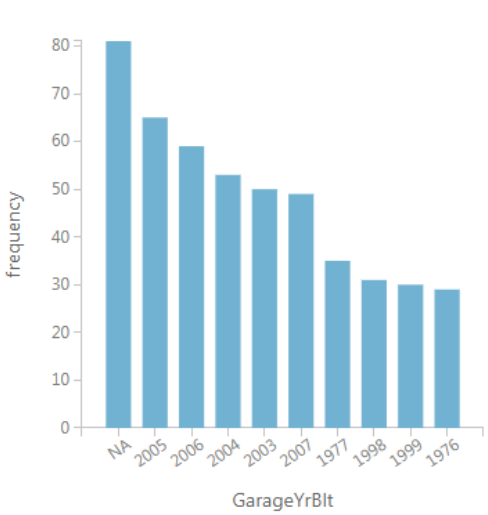
3.3.2 Gráficos de las variables categóricas

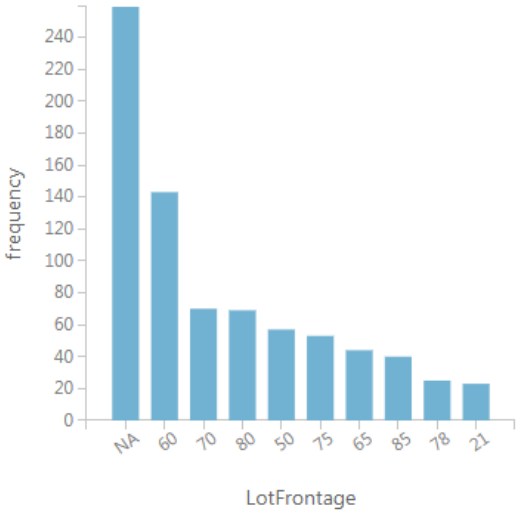
Ahora pasamos a representar las variables categóricas más representativas para ver las diferentes categorías que adquieren y la forma en la que se distribuyen los datos para ver si hay valores o categorías extrañas que puedan influir posteriormente en los modelos predictivos. Los representaremos a través de los gráficos que genera Azure.

- Variable **Barrio de la vivienda:** Destaca por encima del resto que la mayoría de viviendas se sitúan en el barrio de "Names" y en cambio hay muy poca representación en el barrio "Blueste".

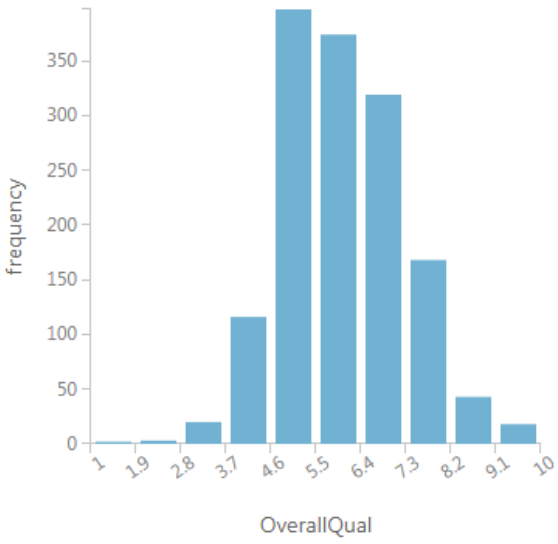


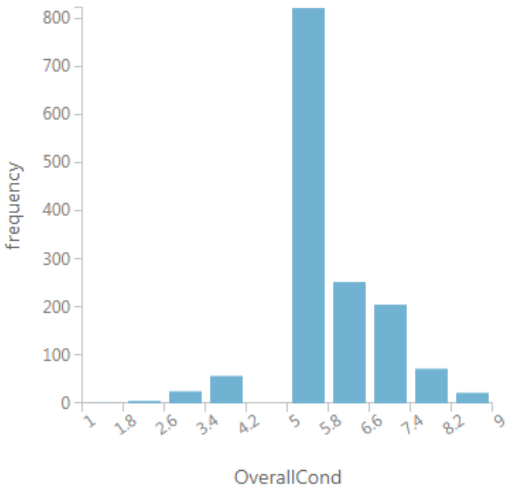
- Variables **año de construcción del garaje (izquierda) y pies lineales conectados a la vivienda (derecha)**: Podemos ver en las dos variables cómo la categoría con más valores es "NA", deberemos tenerlo en cuenta a la hora de escogerla como predictoras ya que podemos encontrarnos datos que hayan sido generados de forma artificial para solventar el problema de los missings.



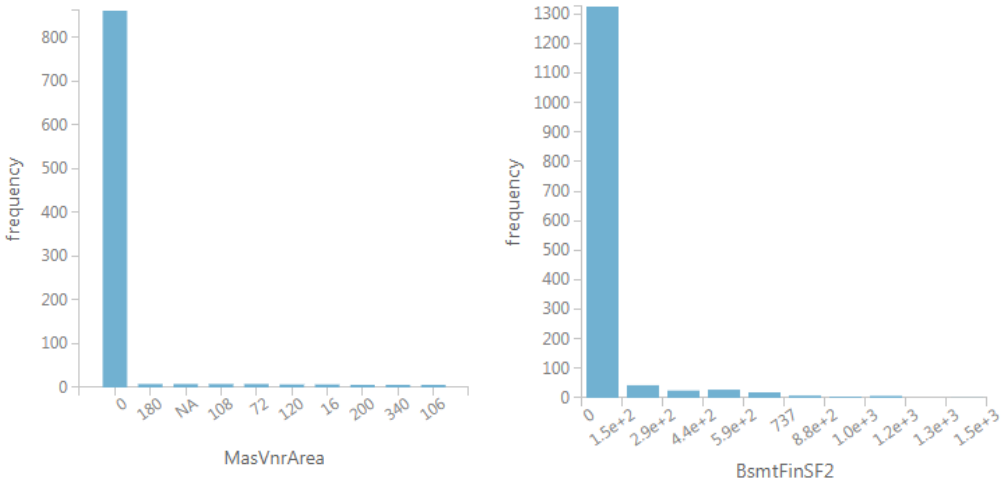


- Variables de **puntuaciones de calidad del acabado y materiales (izquierda) y condición general (derecha)**: Podemos ver como se reparten las puntuaciones entre 0 y 10, por lo que es valorable distribuir los datos en categorías.





- Variables **tamaño del area de mampostería (izquierda) y area terminada del tipo 2 (derecha):** La mayoría de datos tienen un tamaño de 0 pies cuadrados, probablemente no tengan dicho área, estas variable no deben ser influyentes en los posteriores modelos ya que dependen de otras variables que nos dicen si tienen estas características.

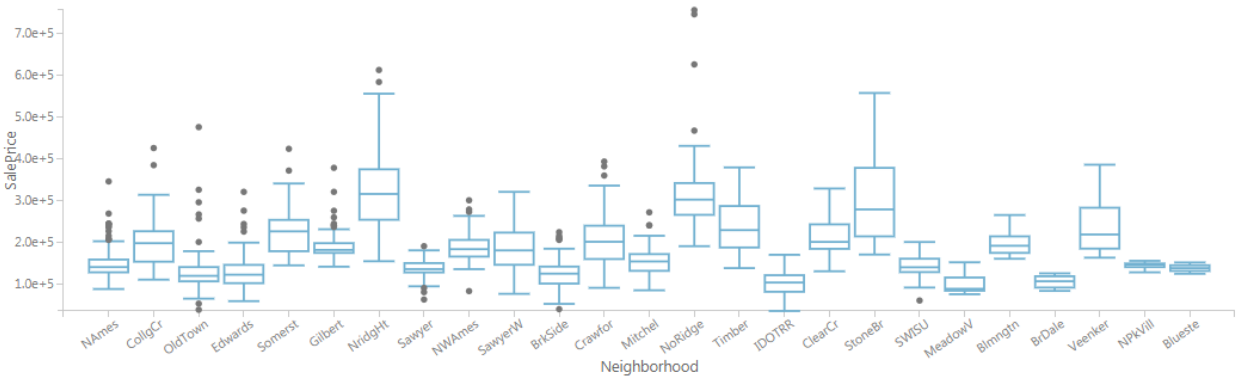


3.3.3 Gráficos bivariantes con la variable respuesta

En este apartado podemos observar el comportamiento de las variables explicativas tanto numéricas como categóricas en la variable respuesta. De esta forma si las variables independientes actúan de forma esperada en relación al precio de la vivienda.

Para ver estos gráficos en Azure, se abre la base de datos con el botón derecho en cualquier función que la muestre y seguidamente indicaremos una columna de la variable explicativa que nos gustaría comparar. Cuando la seleccionamos, directamente nos aparece un gráfico de la distribución de sus datos (histograma si es categórica, boxplot si es numérica), encima del gráfico hay un menú desplegable de comparación. Seleccionamos la variable respuesta y directamente se crea el gráfico bivariante más adecuado para poder ver cómo actúa la iteración de las variables.

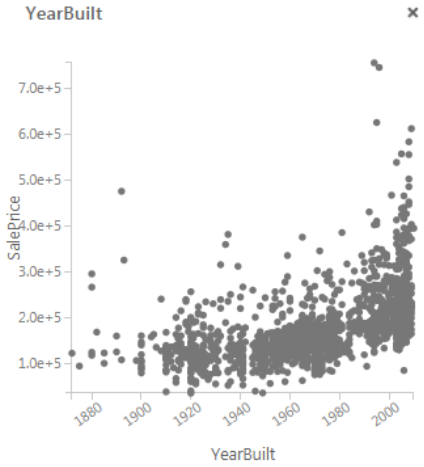
Barrio vs precio de venta



Este gráfico es muy interesante, ya que podemos apreciar en que barrios son más caras las viviendas. En un primer vistazo nos encontramos que en los barrios *NridgHt* y *StoneBr* tenemos el 25% de las viviendas con un valor muy elevado, destacar también el barrio de *NoRidge* ya que tiene las casas más valiosas de toda la ciudad, podemos intuir que es un barrio bastante caro ya que tiene la mediana de los datos superior a los demás y vivienda más barata en ese barrio está por encima de la mediana de los demás barrios. Por el otro lado podemos observar los barrios más pobres, los que tienen la mediana de

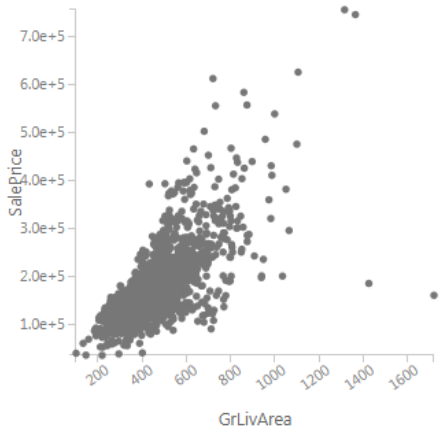
sus datos inferior al resto y además cuentan con casas de muy poco valor, estos barrios son: *OldTown*, *Edwards* y *IDOTRR*.

Año de construcción vs precio de venta



En este gráfico se ve claramente como las viviendas más nuevas tienen un precio más elevado. Es algo esperado ya que el tiempo es un elemento que hace decrecer su valor.
Nos encontramos alguna vivienda en años de construcción muy antiguos que tiene un valor elevado, no podemos decidir sobre ella ya que nos falta información (puede tener muchos metros cuadrados construidos).

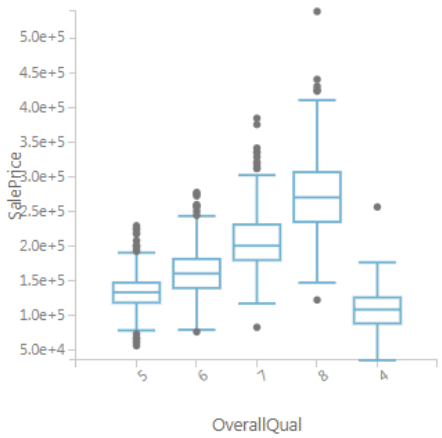
Metros cuadrados de vivienda vs precio de venta



En este grafico podemos ver claramente que los metros cuadrados de vivienda están linealmente relacionados con el precio de venta. A mayor área, mayor precio.

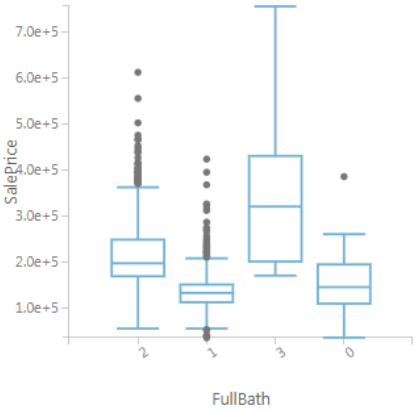
Tenemos dos viviendas que se encuentran lejos de ésta "recta" de puntos, se trata de dos viviendas que tienen un tamaño muy grande pero su precio es muy bajo. Esto puede ser debido a que se encuentren en una zona que en la que los metros cuadrados de vivienda no tengan un precio elevado.

Calidad de los materiales vs precio de venta



En este gráfico podemos apreciar claramente la importancia de esta variable en el precio de la vivienda. Tenemos que las viviendas con una puntuación de 4 son las que tienen un menor precio y las que tienen una puntuación de 8 en general son las más caras.

Baños completos vs precio de venta

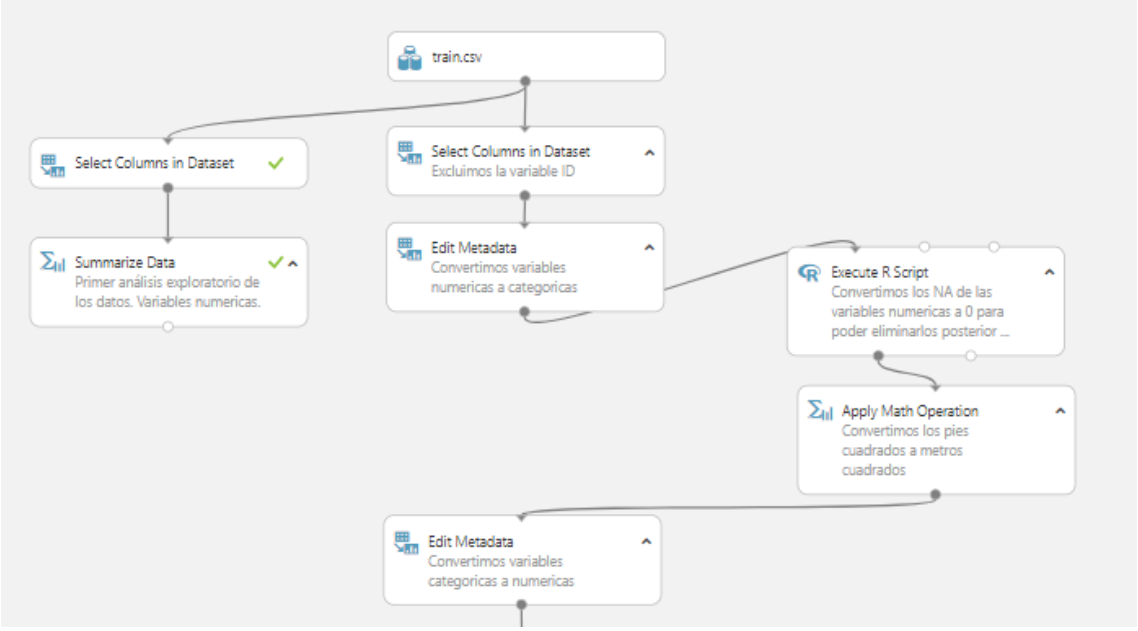


En este gráfico de boxplots, igual que en el gráfico anterior, obtenemos que a mayor número de baños en la vivienda, mayor es el precio.

Destacar que esta variable se trata de baños completos sobre el nivel del suelo (sin contar el sótano). Por lo que puede ser 0 si la vivienda tiene "medios baños", es decir sin bañera.

3.4 Pre-procesado de los datos

Para poder trabajar con la base de datos, lo primero es tenerla depurada, sin valores faltantes o datos atípicos. Para ello es esencial la parte de pre procesado de los datos, el en que trataremos de dejarla lista para poder predecir el precio de las viviendas de la ciudad de Ames según sus características. Para ello no tendremos en cuenta todas las variables, sino que haremos un análisis de las que tienen mejor potencia predictiva en relación a la variable respuesta.



Empezamos el pre-procesado convirtiendo las variables numéricas que hemos visto en el apartado anterior que deberían asignarse a categóricas mediante la función "edit metadata". Seleccionamos las variables *MSSubClass, MSZoning, OverallQual, OverallCond, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageCars, MoSold* y las asignamos "make categorical"

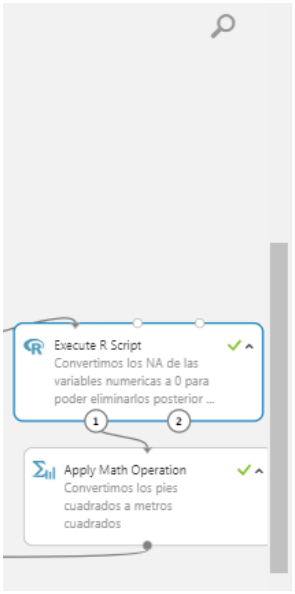
Además, vamos a transformar los datos que tenemos en pies cuadrados a metros cuadrados aplicando una función matemática. Seleccionamos las columnas de la base de datos que queremos transformar y les aplicamos una operación matemática, multiplicar por una constante k=0.3048 y remplazamos los datos anteriores con los nuevos datos.

Seguidamente, seleccionamos las variables numéricas de tipo año y las convertimos a enteros, ya que nos interesa tener años concretos y no decimales en las fechas de construcción y venta.

3.4.1 Tratamiento de missings

Como hemos comentado anteriormente, nuestra base de datos a priori no tiene ningún valor faltante, es decir, todas las celdas están llenas. Pero si la analizamos detenidamente, vemos cómo hay varias variables categóricas que tienen una categoría denominada "NA" que forma parte de sus niveles de datos, en cambio esto también ocurre para distintas variables numéricas (si son missings). Por lo que nos encontramos un problema añadido ya que hemos de analizar cada variable en y ver si el valor "NA" es una categoría esperada o un valor faltante.

Para tratar estos casos, una vez hemos analizado las variables que realmente tienen missings (LotFrontage, MasVnrArea y GarageYrBlt), creamos un pequeño script en R donde seleccionamos todos los "NA" de estas variables y los pasamos a valores vacíos. Seguidamente asignamos que sean variables numéricas, ya que con los "NA" se había asignado el valor string a los datos de estas columnas, y no se podían tratar.



R Script

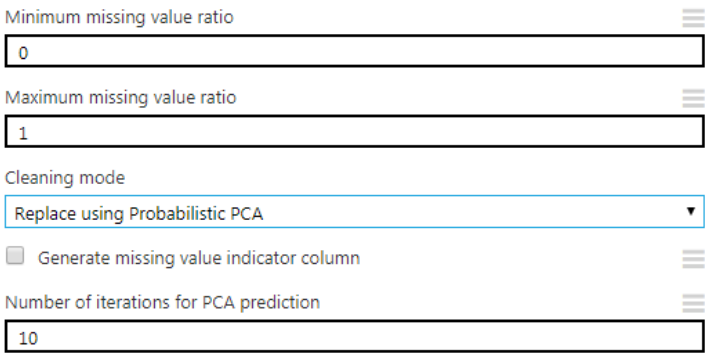
```r
# Map 1-based optional input ports to variables
Dataset1 <- maml.mapInputPort(1) # class: data.frame
summary(Dataset1$LotFrontage)
summary(Dataset1$MasVnrArea)

Dataset1$LotFrontage[Dataset1$LotFrontage == "NA"] <-""
Dataset1$MasVnrArea[Dataset1$MasVnrArea == "NA"] <-""
Dataset1$GarageYrBlt[Dataset1$GarageYrBlt=="NA"] <-""
Dataset1$LotFrontage <- as.numeric(Dataset1$LotFrontage)
Dataset1$MasVnrArea <- as.numeric(Dataset1$MasVnrArea)
Dataset1$GarageYrBlt <- as.numeric(Dataset1$GarageYrBlt)

plot(Dataset1$LotFrontage)
plot(Dataset1$MasVnrArea)

# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("Dataset1");
```

Una vez hemos encontrado las variables categóricas a las que les corresponde los "NA" y las variables numéricas que tienen missings vamos a crear valores artificiales para estas celdas vacías. Azure nos permite utilizar varios métodos para el tratamiento de missings, en nuestro caso vamos a utilizar el método probabilistic PCA. Para ello implementaremos la función Clean Missing Data y la configuraremos para que haga 10 iteraciones en las variables numéricas seleccionadas.



Una vez ejecutada la función vemos como los valores que vacíos se llenan con valores probables teniendo en cuenta las demás variables, con tal de ajustarlo al valor que podría tener esa vivienda si fuese conocido. A continuación podemos ver un ejemplo:

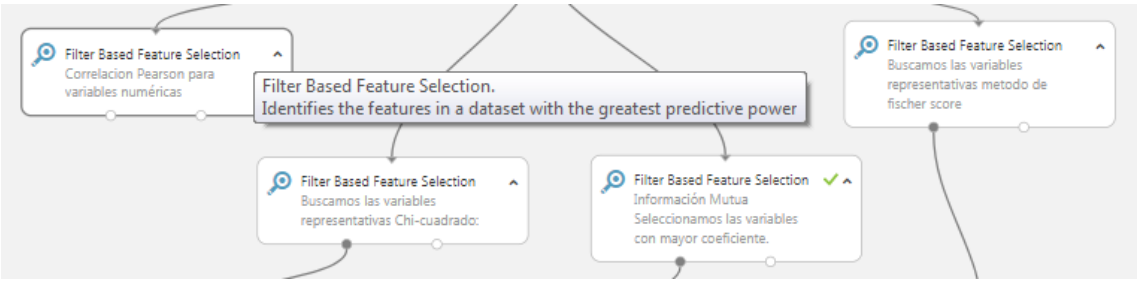
MS Zoning	LotArea	Street		MS Zoning	LotArea	Street
60	19.812	2575.560		60	19.812	2575.560
20	24.384	2926.080		20	24.384	2926.080
60	20.7264	3429.000		60	20.726	3429.000
70	18.288	2910.840		70	18.288	2910.840
60	25.6032	4346.448		60	25.603	4346.448
50	25.908	4302.252		50	25.908	4302.252
20	22.86	3073.603		20	22.860	3073.603
60		3164.434		60	21.181	3164.434
50	15.5448	1865.376		50	15.545	1865.376
190	15.24	2261.616		190	15.240	2261.616
20	21.336	3413.760		20	21.336	3413.760
60	25.908	3634.435		60	25.908	3634.435
20		3952.646		20	17.834	3952.646
20	27.7368	3246.730		20	27.737	3246.730
20		3328.416		20	22.207	3328.416
45	15.5448	1865.376		45	15.545	1865.376
20		3426.257		20	22.001	3426.257
90	21.9456	3289.097		90	21.946	3289.097

3.4.2 Selección de variables con más capacidad predictiva

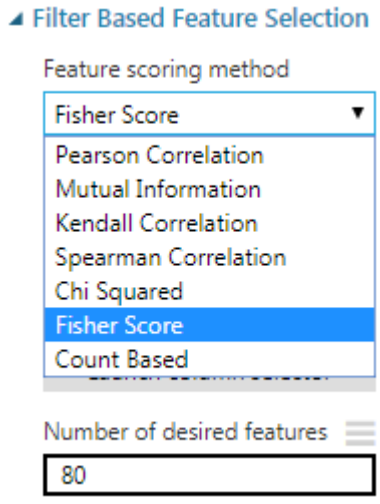
Como hemos visto, tenemos una base de datos con 80 variables explicativas, hemos de analizar cuales son las que tienen una mayor potencia predictiva para la variable respuesta. En la parte de pre-procesado de los datos hemos visto que teníamos variables con muchos valores perdidos (ahora son valores artificiales) y otras variables con la mayoría de sus valores en una sola categoría. Por lo que ya nos indica que no pueden influir demasaido en la predicción de la variable dependiente.

En este apartado, hemos utilizado tres de los siete métodos que dispone azure para tratar de encontrar las variables con más poder predictivo. Para ello se escoge la función "Filter Based Feature Selection" y dentro de ella se puede seleccionar el tipo de método que queremos aplicar.

Además de los tres métodos de selección de variables, añadimos la función de correlación de pearson para ver que variables están mas relacionadas entre ellas.



Puntaje de Fisher



En este caso de ejemplo, seleccionamos el método del puntaje de Fisher, para las 80 variables explicativas.
Nos devuelve dos tablas, una con todas las columnas de las variables seleccionadas y otra con solo una fila con el valor del puntaje de fisher de las variables explicativas con la variable respuesta (gráfico inferior).

SalePrice	MiscVal	GrLivArea	LotArea	GarageArea	1stFlrSF
1	3.119997	2.853253	2.729277	2.145761	1.942493

De todas las variables a las que se les aplica el método, seleccionaremos las que tienen una puntuación por encima de 1.90, y las guardaremos en una nueva base de datos que la llamaremos: 7 variables → fisher score.

Las variables seleccionadas son: *MiscVal, GrLivArea, LotArea, GarageArea, 1stFlrSF, TotalBsmtSF, YearBuilt*

Información Mútua

Con el método de la información mutua nos encontramos que las variables con mayor potencia predictiva con la variable respuesta han cambiado respecto el método anterior.

Ahora las variables seleccionadas serán las que tengan una puntuacion por encima de 0.20, esto quiere decir que seleccionamos las siguientes 19 variables: *OverallQual, Neighborhood, GrLivArea, GarageCars, GarageArea, YearBuilt, TotalBsmtSF, BsmtQual, ExterQual, KitchenQual, GarageFinish, 1stFlrSF, FullBath, MSSubClass, GarageYrBlt, YearRemodAdd, GarageType, FireplaceQu.* Guardaremos estas columnas en la base de datos llamada 19 variables → Información mutua.

OverallQual	Neighborhood	GrLivArea	GarageCars	GarageArea	YearBuilt	TotalBsmtSF
0.462448	0.366018	0.359538	0.33663	0.307121	0.297074	0.296555

Chi Cuadrado

Con este método de selección de variables nos encontramos que "a priori" selecciona las mismas variables que el método de información mutua, aunque no en el mismo orden de importancia.

Las variables seleccionadas serán las que tengan una puntuación por encima de 900 ya que son las consideradas con mayor potencia en comparación con el resto. Nos quedaremos con las siguientes 12 variables: *OverallQual, Neighborhood, GarageCars, GrLivArea, GarageArea, BsmtQual, TotalBsmtSF, YearBuilt, ExterQual, KitchenQual, MSSubClass, 1stFlrSF.* Las guardaremos en la base de datos llamada 12 variables → chi cuadrado

OverallQual	Neighborhood	GarageCars	GrLivArea	GarageArea	BsmtQual
1890.855596	1726.722808	1285.158078	1245.002944	1141.792031	1101.604028

Correlación de Pearson

En este apartado podemos ver las variables que más relacionadas están con la variable respuesta "SalePrice". Como era de esperar, nos encontramos que las variables que hemos encontrado con mayor poder predictivo son las que se encuentran correlacionadas con la respuesta.

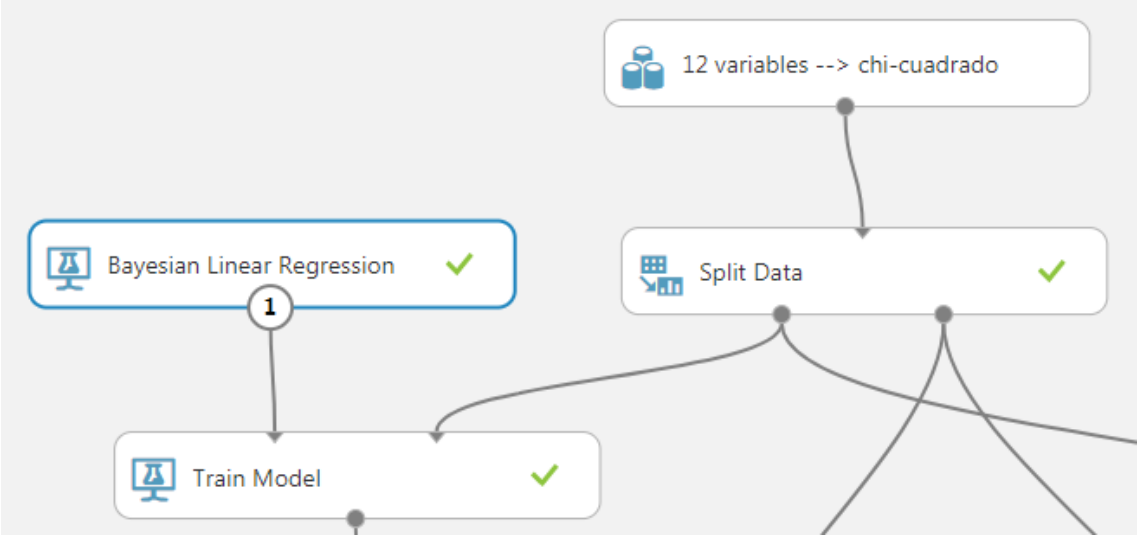
OverallQual	Neighborhood	GrLivArea	GarageCars	ExterQual	BsmtQual	KitchenQual	GarageArea	TotalBsmtSF	1stFlrSF	FullBath
0.827153	0.73863	0.708624	0.700877	0.690933	0.681905	0.675721	0.623431	0.613581	0.605852	0.580028

- La selección de variables es un proceso muy importante, ya que éstas determinaran la información con la que se tratará de predecir la variable respuesta. Por este motivo es necesario que coger las variables que aporten la máxima información y eliminar las que puedan aportar varianza insignificativa, también es importante que las variables seleccionadas no tengan multicolinealidad entre ellas y la respuesta.

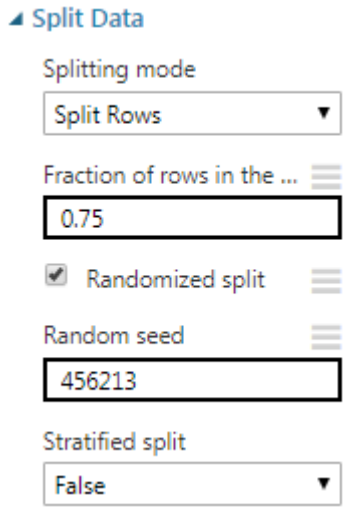
3.5 Modelos predictivos

Una vez tenemos la base de datos preparada y escogidas las variables con mayor potencia predictiva para la variable precio de la vivienda, vamos a implementar los modelos predictivos. Introduciremos cinco modelos machine learning con las tres bases de datos seleccionadas (cada una de ellas con distintas variables) y veremos cuál de los modelos se ajusta mejor al precio real y con qué método de selección de variables. Los modelos que implementaremos son: **regresión bayesiana, regresión lineal, Decision Forest Regression, Boosted Decision Tree Regression y redes neuronales**. Todos los modelos los evaluaremos mediante la cross-validation a excepción de las redes neuronales que haremos una comparativa entre la validación cruzada y la normal.

3.5.1 Modelo de regresión bayesiana



Cogemos una de las bases de datos con la selección de variables realizada, en el ejemplo inferior seleccionamos la que contiene 12 variables mediante el método de selección Chi-cuadrado y la dividimos en dos mediante la función Split data:

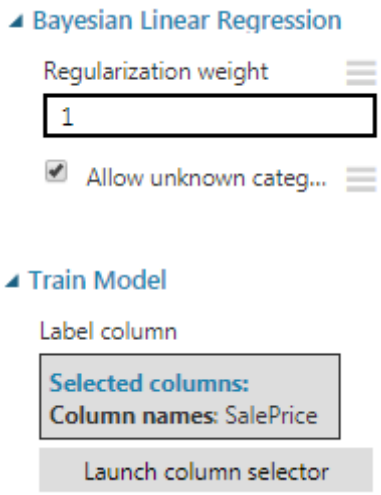


Seleccionamos el modo dividir filas y le aplicamos un porcentaje de 0.75.

Esto nos separará en dos bases de datos diferentes de forma aleatoria el 75% de los datos para "train" y el 25% restante para test. De esta forma, nos evitaremos que haya sobreajuste en el modelo.

Se le añade una semilla para que la partición sea siempre la misma e indicamos falso en la opción de partición estratificada.

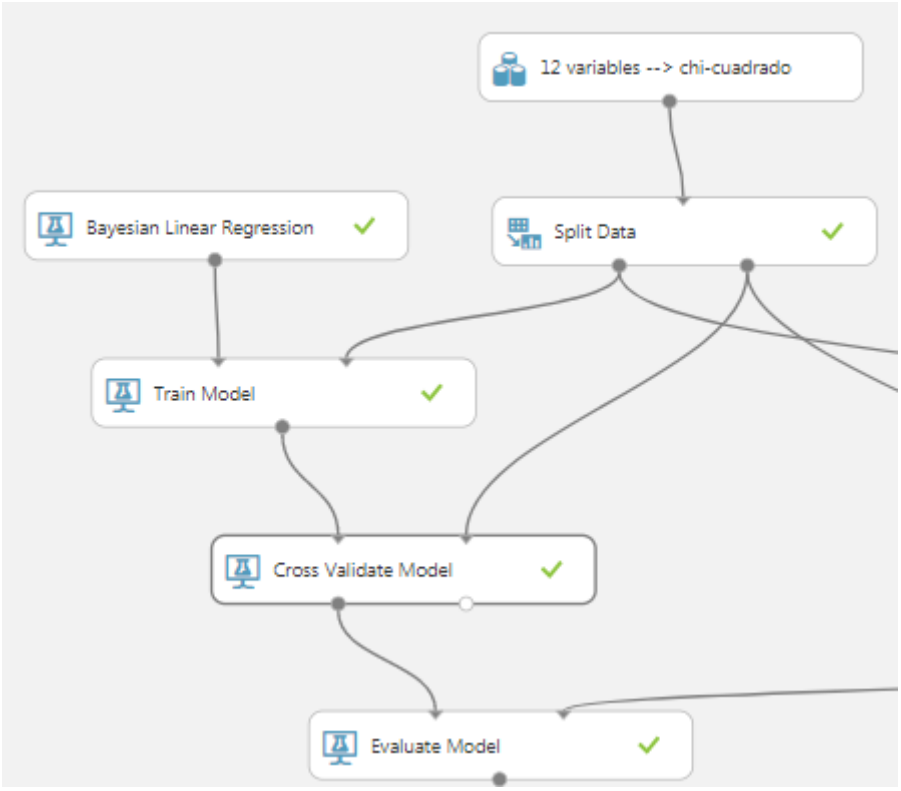
Seguidamente con el conjunto de datos de entrenamiento (train) le introduciremos el modelo de regresión bayesiana, previamente configurado.



Como podemos ver en la imagen de la derecha, es un modelo en el que simplemente nos deja configurar el peso de regularización, nos permite escribir un valor para usar para la regularización, con la finalidad de prevenir el sobreajuste.

En la siguiente imagen, podemos ver como se configura el modelo de entrenamiento. Simplemente hemos de asignar la variable respuesta y Azure se encarga de hacer la regresión con el resto de variables.

Finalmente, nos encontramos con la cross-validation y la evaluación del modelo. En estas funciones ya no podemos configurar nada, simplemente indicar cuál es la variable con la que ha de comparar la predicción (SalePrice) y el resto lo genera el propio software.



Modelo de regresión bayesiana completo.

Ahora vamos a interpretar los resultados de ambas funciones (**Cross-Validate Model y Evaluate Model**). La primera función nos genera dos salidas (los dos nodos de debajo del nombre de la función).

En la tabla inferior podemos ver un ejemplo con nueve filas de la primera salida (scored results), se trata de la predicción para cada individuo en la columna "scored label mean" y también nos muestra la desviación estándar de la predicción, según el modelo de regresión bayesiana.

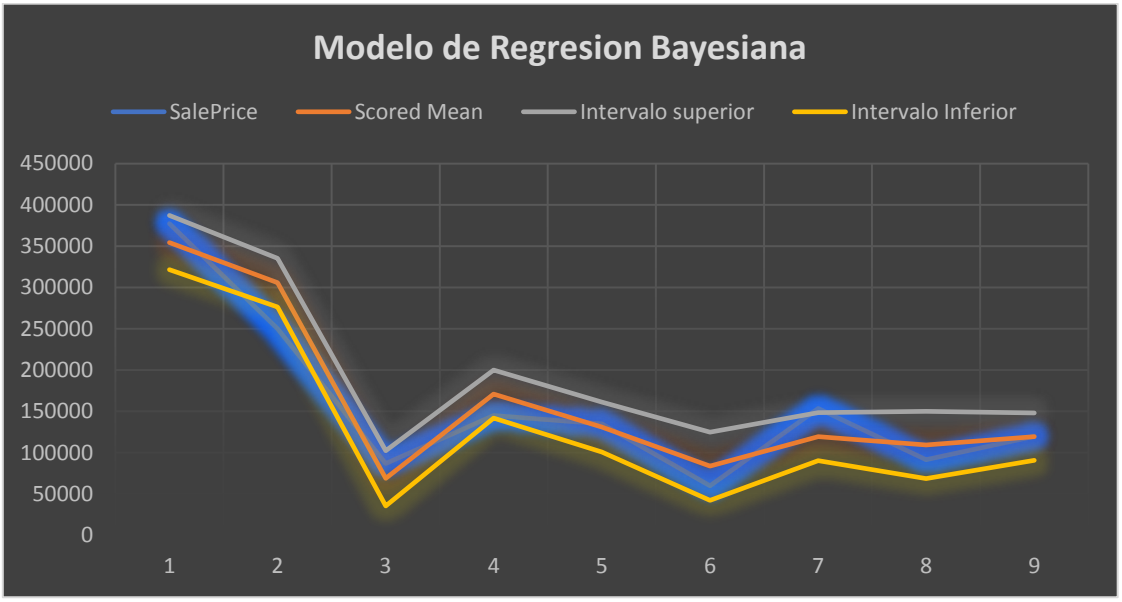
Fold	SalePrice	Overall Qual	Neighborhood	Garage Cars	GrLivArea	Garage Area	TotalBsmtSF	Year Built	Scored Mean	Scored Std.Dev
0	377426	9	StoneBr	3	585.83	206.04	585.83	2005	354458	32831.2
7	250000	8	Somerst	2	607.16	197.51	320.95	2003	305881	29328.72
9	87000	5	OldTown	0	256.03	0	204.83	1930	69034.5	33430.02
5	145000	6	NWAmes	2	383.44	160.93	383.44	1971	170902	29092.89
7	135000	6	Mitchel	2	306.02	153.62	297.18	1977	131128	30416.09
7	60000	2	BrkSide	0	243.84	0	80.47	1936	83776.4	41296.47
8	153575	6	BrkSide	1	425.5	117.04	212.75	1915	119555	29090.01
0	91500	6	BrDale	0	371.25	0	204.83	1971	109423	40569.16
0	119900	5	SawyerW	1	294.13	91.44	252.37	1965	119431	28713.1

La primera columna de la tabla hace referencia a la carpeta a la que ha sido asignada la observación, como hemos comentado anteriormente, se hace de forma aleatoria para evitar el sobreajuste del modelo (método de cross-validación).

En las demás columnas se muestra el valor de las variables explicativas introducidas en el modelo para cada individuo. En la tabla se muestra un ejemplo con siete de las doce variables predictoras.

Finalmente, en las dos últimas columnas podemos ver el valor de la estimación del modelo y su desviación estándar para cada individuo. A simple vista no parece distar mucho del valor real, si cogemos por ejemplo la primera observación obtenemos un valor predicho de 354457.55 con una desviación estándar de 32831.20. Podemos ver como el precio real de la vivienda (377426) está dentro del intervalo estimado mediante la media y la desviación, por lo que ya podemos prever que tendrá un buen ajuste.

Para ver de forma más clara el ajuste del modelo a los datos, se ha creado en Excel un gráfico de las 9 observaciones de la tabla anterior con el precio real y los valores predichos más la desviación estándar (intervalos superiores e inferiores).



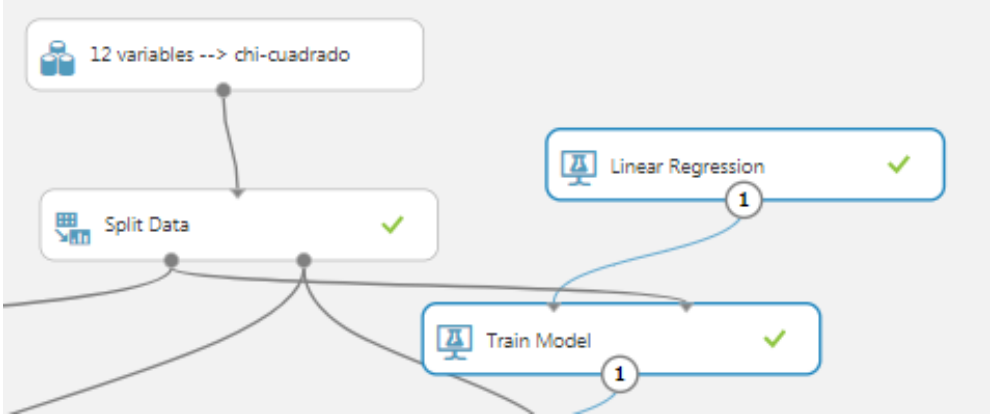
Como podíamos prever, el precio de venta queda englobado por el intervalo formado por la media y la desviación del valor predicho para estas 9 observaciones.

La segunda salida de la función Cross-validación devuelve una tabla con el resumen de las 10 carpetas en las que se han repartido los datos. Las dos primeras columnas nos indican el número de carpeta y cuántas observaciones se han asignado a ella, la siguiente columna es la log verosimilitud negativa, nos indica cuanto probable es la carpeta en el conjunto. Seguidamente encontramos los valores de error puramente estadísticos, y finalmente el R-Cuadrado o Coeficiente de determinación.

Fold Number	Examples in fold	Negative Log Likelihood	Mean Absolute Error	Root MSE	Relative Absolute Error	Relative Squared Error	R-squared
0	36	415.701	18215.856	24215.370	0.256	0.075	0.925
1	37	426.882	19243.277	23917.360	0.315	0.092	0.908
2	37	443.514	21219.659	36903.883	0.284	0.095	0.905
3	37	436.314	22659.627	31704.627	0.515	0.366	0.634
4	37	431.005	23390.483	27738.014	0.575	0.251	0.749
5	36	418.852	18560.027	26047.472	0.368	0.141	0.859
6	36	427.372	22390.944	34407.205	0.390	0.208	0.792
7	36	414.370	18403.732	22712.530	0.323	0.078	0.922
8	37	432.730	21136.641	29994.399	0.258	0.066	0.934
9	36	428.446	25852.019	35250.322	0.431	0.224	0.776
Mean	365	427.519	21107.226	29289.118	0.371	0.160	0.840
St.Dev	365	9.198	2526.067	5129.565	0.107	0.099	0.099

Como podemos ver en la tabla, todas las carpetas tienen un similar valor de verosimilitud, los errores también se encuentran muy relacionados, excepto en la carpeta 3 y 4 que están por encima de los demás. Finalmente observando el coeficiente de determinación vemos como las carpetas con los valores más bajos son las mencionadas con errores elevados. A pesar de ello, obtenemos una media de R-cuadrado de 0.840, todos los valores de las carpetas, a excepción de la tercera se encuentran cercanos y la desviación del R-cuadrado es pequeña, se puede decir que no existe riesgo de sobreajuste en los datos.

3.5.2 Modelo de regresión Lineal



Para explicar el modelo de regresión lineal, seguiremos utilizando la base de datos con las 12 variables seleccionadas por el método de chi-cuadrado. En el siguiente apartado comentaremos las diferencias entre las tres bases de datos escogidas, pero el método de implementar los modelos es el mismo independientemente de las variables seleccionadas, simplemente hay que ajustar los valores de los parámetros.

Igual que en el apartado anterior, y para el resto de modelos, partiremos los datos en dos muestras: train y test. Seleccionamos el 75% de los datos para entrenar el modelo y el 25% restante para probarlo.

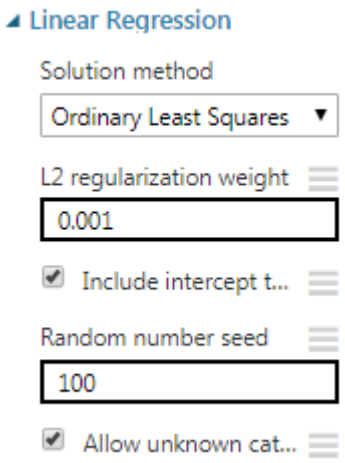


Imagen 1

Si entramos en la función "Linear Regression" nos encontramos con el menú de la imagen 1. El primer desplegable nos deja escoger el método de solución que queremos: "Ordinary Least Squares" o "Online Gradient Descent". En nuestro caso trabajaremos con el primero, ya que es el que se ha estudiado durante el grado.

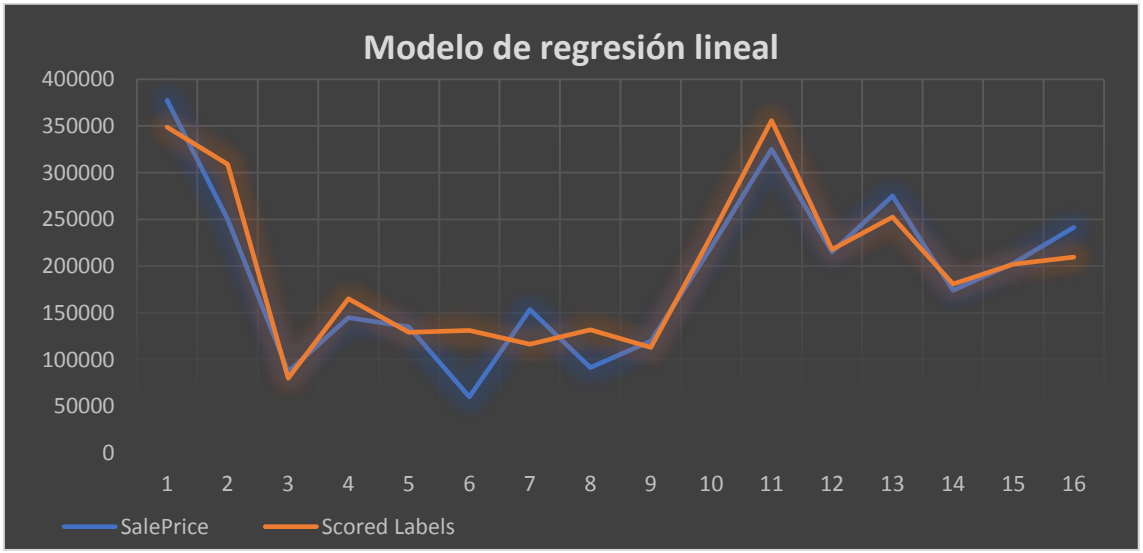
Seguidamente podemos seleccionar el valor de L2, es importante elegir un valor diferente de 0 para evitar el sobreajuste. Y podemos marcar la opción de incluir el termino intercept en el modelo. Finalmente introducimos un número a modo de semilla y ejecutamos.

Una vez tenemos el modelo parametrizado y ejecutado es hora de analizar los resultados obtenidos. Igual que en el anterior modelo, encontramos la tabla de la primera salida de la función cross-validación. En este caso se han seleccionado menos variables para mostrar de ejemplo ya que lo que nos interesa es ver si ajusta bien los precios a la realidad. El modelo lineal ya no nos da una estimación de la desviación estándar de la media estimada, sino que lo engloba todo en una misma predicción (scored labels).

Fold	OverallQual	Neighborhood	GarageCars	GrLivArea	SalePrice	Scored Labels
6	9	StoneBr	3	585.8256	377426	348910.63
3	8	Somerst	2	607.1616	250000	308880.18
9	5	OldTown	0	256.032	87000	79953.43
4	6	NWAmes	2	383.4384	145000	164965.75
9	6	Mitchel	2	306.0192	135000	128983.54
4	2	BrkSide	0	243.84	60000	131060.10
8	6	BrkSide	1	425.5008	153575	116417.08
6	6	BrDale	0	371.2464	91500	131749.58
7	5	SawyerW	1	294.132	119900	113012.82
9	8	SawyerW	2	583.692	220000	231925.16
0	9	StoneBr	3	763.2192	325000	355808.91
0	8	SawyerW	2	582.4728	215000	217811.13
4	8	CollgCr	3	544.6776	275000	252577.65
3	6	NAmes	2	468.4776	174000	180707.83
6	7	CollgCr	2	458.4192	203000	202003.48
3	7	NWAmes	2	570.5856	241500	209780.63

En este modelo vemos como también se ajusta bastante a los precios reales de las viviendas. Si analizamos detalladamente cada observación vemos como por ejemplo la tercera se ajusta mucho al valor real, en cambio en la sexta observación hay una gran diferencia (más del doble del valor) entre el precio real y el predicho. Esto obviamente afectará al ajuste de la carpeta número 4, ya que hará aumentar el error estándar y por lo tanto disminuirá el coeficiente de determinación.

Seguidamente también se ha hecho un gráfico con las observaciones de la tabla anterior para ver de forma visual el comportamiento de los valores predichos del modelo.



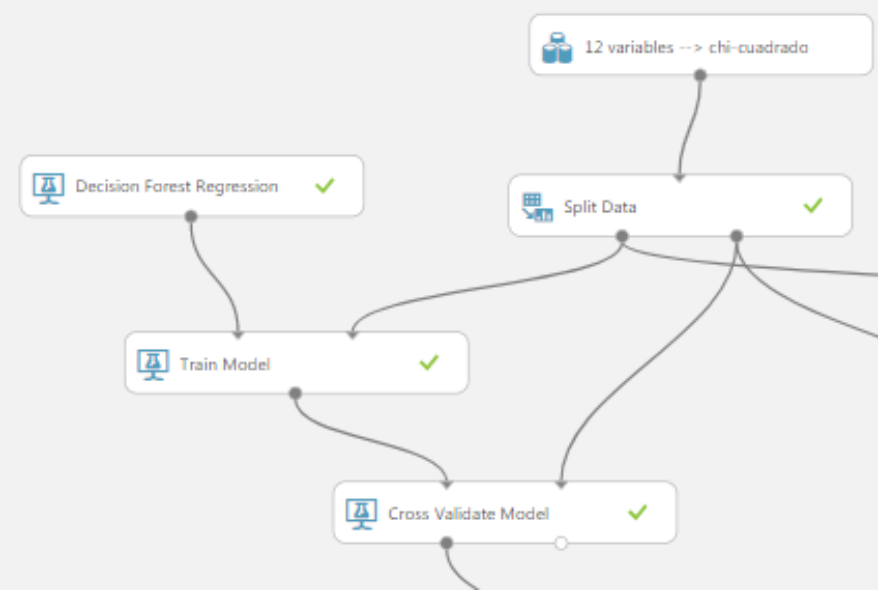
Gráficamente, podemos apreciar como las primeras observaciones, hasta la numero 5 se ajusta bastante al precio real de la vivienda, pero a partir de la sexta y hasta la novena observación hay una gran discrepancia. Deberíamos analizar los valores de cada variable explicativa para tratar de encontrar el motivo por el cual la predicción es tan poco ajustada en esas observaciones.

Por último, vamos a analizar el ajuste de cada carpeta por separado y en conjunto del modelo mediante la segunda salida de la función cross-validación.

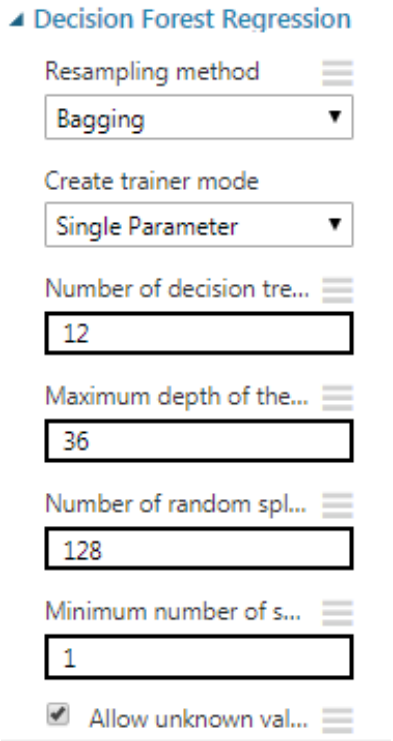
Fold	Num	Mean Absolute Error	Root MSE	Relative Absolute Error	Relative Squared Error	R-squared
0	36	19764.878	33776.313	0.272	0.085	0.915
1	37	21366.125	28760.623	0.430	0.202	0.798
2	36	20053.018	25235.057	0.485	0.254	0.746
3	37	20361.885	28233.911	0.329	0.112	0.888
4	36	27653.351	39021.268	0.535	0.301	0.699
5	36	22727.561	32207.380	0.466	0.282	0.718
6	37	23497.252	34960.016	0.345	0.115	0.885
7	37	23315.651	29313.229	0.340	0.102	0.898
8	36	26166.210	36438.283	0.308	0.117	0.883
9	37	16734.766	26893.809	0.405	0.203	0.797
Mean	365	22164.070	31483.989	0.392	0.177	0.823
St.dev	365	3217.0546	4497.249	0.085982	0.081323	0.081323

Con el modelo de regresión lineal obtenemos un coeficiente de determinación de 0.823 que representa la media de los R-cuadrado de todas las carpetas. Al igual que en la regresión bayesiana, son valores de ajuste muy elevados que nos indican que es un buen modelo. Analizando la Cross-validación vemos como la carpeta con peor ajuste es la numero cuatro, como preveíamos la observación que hemos comentado que no estaba siendo bien predicha puede haber sido el motivo por el cuál esta carpeta tiene la peor puntuación en el R-cuadrado.

3.5.3 Decision Forest Regression



En este apartado explicaremos como crear un modelo de regresión usando el algoritmo de bosque de decisión. Para ello seguiremos los pasos de los modelos anteriores y la misma base de datos (12 variables seleccionadas por el método chi-cuadrado). Separaremos los datos en train y test y parametrizaremos el modelo buscando el mejor ajuste.



Abrimos las propiedades del módulo y la primera opción es seleccionar el método de Resampling. Aquí podemos elegir entre dos opciones para crear los árboles de decisión: Bagging (Embolsado) o Replicate (Replicado).

El método Bagging, o también llamado agregación de bootstrap, hace que cada árbol del bosque de decisión genera una distribución gaussiana a modo de predicción. La agregación consiste en encontrar un árbol gaussiano cuyos primeros dos momentos coinciden con los momentos de la mezcla de gaussianos dada al combinar todos los gaussianos devueltos por árboles individuales. En el método *replicate*, cada árbol está entrenado exactamente con los mismos datos de entrada. La determinación de qué predicado dividido se usa para cada nodo de árbol permanece aleatorio y los árboles serán diversos.

Especificaremos cómo queremos que se entrene el modelo, configurando la opción *Create trainer mode*. En nuestro caso seleccionaremos *Single Parameter* ya que sabemos cómo configurar el árbol de decisión.

En la opción número de árboles de decisión , indicaremos el número total de árboles de decisión para crear en el conjunto. Al crear más árboles de decisión, se puede obtener una mejor cobertura, pero aumentará el tiempo de capacitación. En nuestro caso seleccionaremos 12 que es el mismo número de variables explicativas introducidas.

En el apartado de profundidad máxima de los árboles de decisión, escribiremos un valor para limitar la profundidad de cada árbol de decisión. Si aumentamos la profundidad del árbol puede aumentar la precisión, a riesgo de un exceso de ajuste y un mayor tiempo de entrenamiento. En nuestro caso seleccionamos 36 ya que es 3 veces más que el número de árboles de decisión.

En el apartado número de divisiones aleatorias por nodo, escribimos la cantidad de divisiones que se usarán al construir cada nodo del árbol. Una división significa que las características en cada nivel del árbol (nodo) se dividen aleatoriamente. Este valor lo dejamos por defecto.

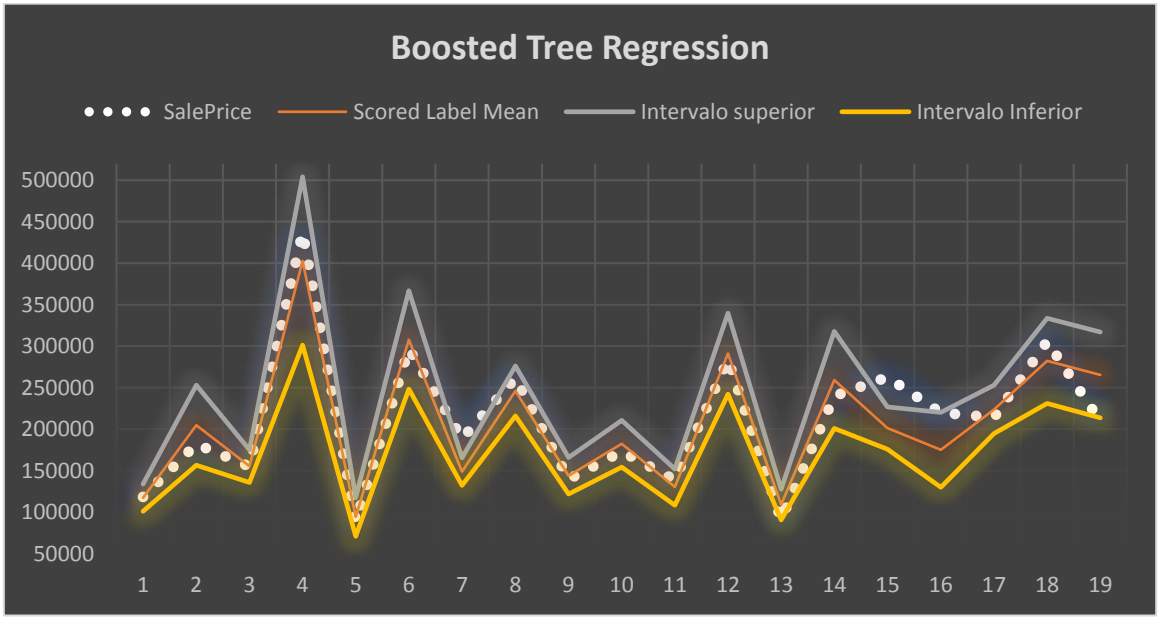
Finalmente, encontramos la opción número mínimo de muestras por nodo hoja, aquí se indica el número mínimo de casos que se requieren para crear cualquier nodo terminal (hoja) en un árbol. Lo dejaremos por defecto ya que al aumentar este valor, aumenta el umbral para crear nuevas reglas. En cambio, con el valor predeterminado de 1, incluso un solo caso puede provocar la creación de una nueva regla. Si aumenta el valor a 5, los datos de entrenamiento deberían contener al menos 5 casos que cumplan las mismas condiciones.

Ahora pasaremos a ver las predicciones puntuales para cada individuo. Usamos también el método de la cross-validation, en la siguiente tabla podemos ver como se han asignado las carpetas a cada vivienda y tres variables explicativas a modo informativo. Finalmente encontramos las predicciones, este método nos permite obtener la media y desviación de la estimaciones.

Fold	YearBuilt	KitchenQual	1stFlrSF	SalePrice	Scored Label Mean	Scored Label Standard Deviation
5	1990	TA	248.72	118500	117376.28	16315.07
2	2007	Gd	435.25	182000	204671.94	48260.87
6	1966	TA	201.17	155000	155481.94	19586.47
9	2005	Ex	563.27	437154	402777.19	101393.20
4	1921	TA	187.76	89000	94251.74	23342.09
3	2007	Gd	523.65	297000	307622.08	59213.66
8	1934	TA	258.78	188700	148854.17	16547.43
4	2001	Ex	395.94	261500	245995.18	30026.65
8	1940	TA	287.73	139500	143854.31	21862.10
8	1990	Gd	370.94	173000	182560.56	28209.29
8	1962	TA	332.84	139000	130168.06	21675.00

8	2003	Gd	306.63	284000	291050.65	48908.03
0	1930	TA	240.79	91000	109163.47	18400.99
0	2005	Gd	274.62	239900	259323.85	58336.29
6	1968	TA	657.15	262500	201347.92	25627.51
6	1932	TA	334.06	220000	175073.82	45144.15
6	2002	Gd	468.17	214000	223855.55	29027.41
2	2004	Gd	609.60	305900	282384.63	51276.20
3	2006	Gd	473.66	209500	265382.24	51667.99

A primera vista, el modelo engloba mediante la media y la desviación estándar el valor real de las viviendas. Para tratar de analizarlo mejor, crearemos un gráfico de líneas en Excel con el valor real y las predicciones con la desviación estándar.



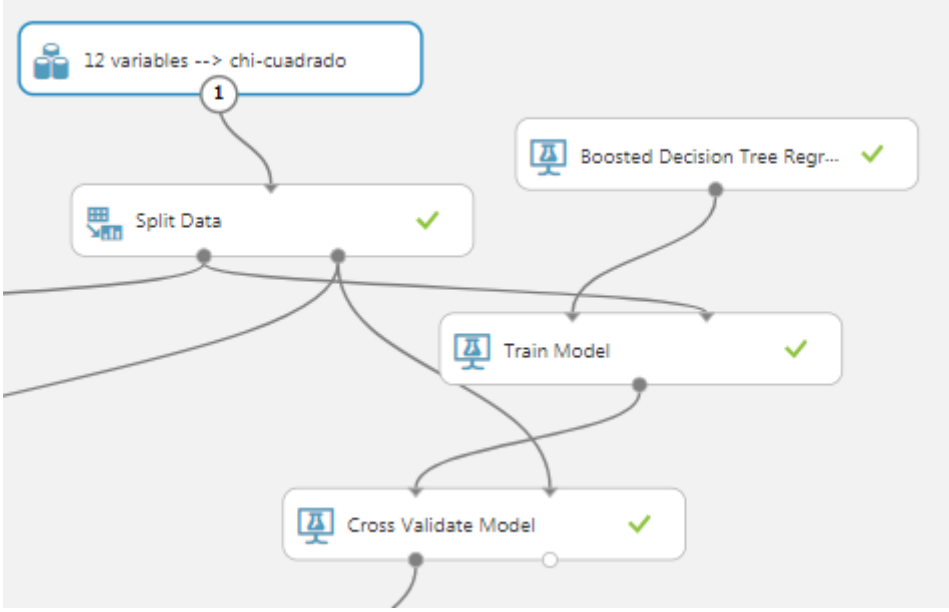
Como preveíamos, tenemos una predicción bastante ajustada dentro de los intervalos de la media más la desviación. En la observación 15 encontramos que se sale del intervalo predicho, se trata de una vivienda del año 1968 y con una primera planta de más de 600 metros cuadrados a la que le estamos prediciendo un precio inferior al real. Tendremos que investigar las demás variables predictoras para ver el motivo de esta diferencia en el ajuste.

Fold	Num	Negative Log Likelihood	Mean Absolute Error	Root MSE	Relative Absolute Error	Relative Squared Error	R-Squared
0	43	489.15	17904.57	23454.56	0.40	0.19	0.81
1	44	520.69	22460.93	31734.22	0.44	0.24	0.76
2	44	517.36	27920.98	44870.43	0.41	0.28	0.72
3	44	513.58	22404.22	30887.77	0.37	0.16	0.84
4	43	492.28	19192.39	27895.75	0.39	0.17	0.83

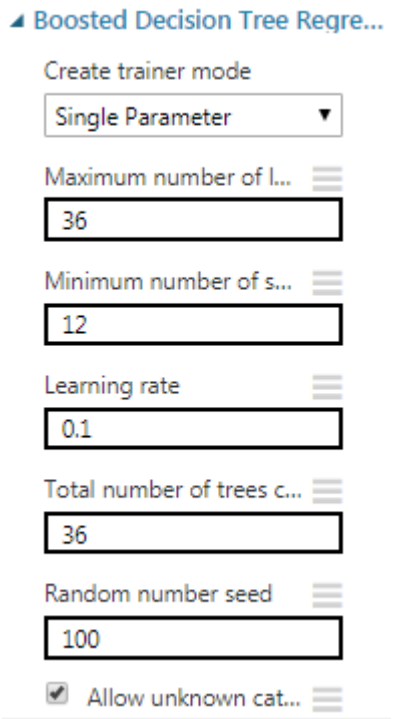
5	44	558.17	27530.10	56431.48	0.43	0.28	0.72
6	44	516.83	21081.25	36315.43	0.36	0.22	0.78
7	44	500.95	14298.20	19146.01	0.25	0.07	0.93
8	44	519.12	28330.03	38304.97	0.59	0.32	0.68
9	44	526.77	23577.91	36321.82	0.39	0.22	0.78
Mean	438	515.49	22470.06	34536.24	0.40	0.22	0.78
St.Dev	438	19.55	4608.96	10706.57	0.09	0.07	0.07

Todas las carpetas tienen un coeficiente de determinación por encima de 0.70, excepto la octava, aunque está muy cerca. Por lo que, además viendo la desviación estándar del r-cuadrado podemos decir que no hay sobreajuste. Mirando la verosimilitud obtenemos valores parecidos en todas las carpetas, junto con el 0.78 de media, es otro síntoma de que es un buen modelo.

3.5.4 Boosted Decision Tree Regression



En este apartado explicaremos como crear un modelo de regresión usando el algoritmo del árbol de decisión reforzado. Para ello seguiremos los pasos de los modelos anteriores y la misma base de datos (12 variables seleccionadas por el método chi-cuadrado). Separaremos los datos en train y test y parametrizaremos el modelo buscando el mejor ajuste.



Para la parametrización del modelo deberemos configurar los apartados que aparecen en la imagen. El primero, igual que en el modelo anterior, deberemos configurar el modo de entrenador, en este caso también seleccionaremos “Single Parameter” ya que conocemos los parámetros que queremos asignarle al modelo, en caso que no lo supiésemos escogeríamos la opción “Parameter Range”, entonces el porpio algoritmo buscaría en función de un rango de parámetros que le indicamos el que funcionaria mejor para el modelo. Esto es un gasto computacional elevado en comparación a introducirlo directamente nosotros.

La siguiente opción es el número máximo de hojas por árbol, indica la cantidad máxima de nodos terminales (hojas) que se pueden crear en cualquier árbol. Si aumentamos este valor, posiblemente aumente el tamaño del árbol y obtenga una mayor precisión, a riesgo de un ajuste excesivo y un tiempo de entrenamiento más prolongado. Nosotros configuramos 36, que es el triple de variables que tenemos en el modelo.

Seguidamente deberemos configurar el número mínimo de muestras por nodo hoja, indica el número mínimo de casos necesarios para crear cualquier nodo terminal (hoja)

en un árbol. Si aumentamos este valor, aumenta el umbral para crear nuevas reglas. Por ejemplo, con el valor predeterminado de 1, incluso un solo caso puede provocar la creación de una nueva regla. Si aumenta el valor a 5, los datos de entrenamiento deberían contener al menos 5 casos que cumplan las mismas condiciones. En nuestro caso asignamos 12 ya que es el valor que proporciona un mayor ajuste al modelo.

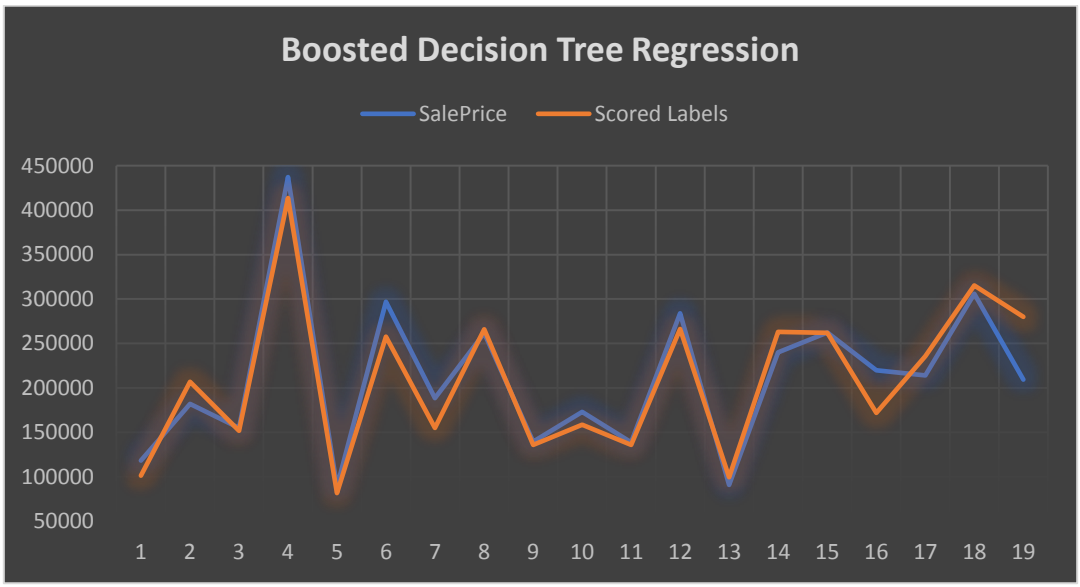
Tasa de aprendizaje: escribiremos un número entre 0 y 1 que defina el tamaño del paso mientras aprende. La tasa de aprendizaje determina qué tan rápido o lento converge el modelo en la solución óptima. Si el tamaño del paso es demasiado grande, puede sobrepasar la solución óptima. Si el tamaño del paso es demasiado pequeño, el entrenamiento tarda más en encontrar la mejor solución. Configuramos el valor en 0.1 ya que es lo suficientemente ajustado para encontrar la solución óptima.

Número de árboles construidos: indicaremos el número total de árboles de decisión para crear en el conjunto. Al crear más árboles de decisión, se puede obtener una mejor cobertura, pero el tiempo de ejecución aumenta. Este valor también controla la cantidad de árboles que se muestran al visualizar el modelo entrenado. si quiere ver o imprimir un árbol de un solo nodo, se puede establecer el valor en 1. Sin embargo, esto significa que solo se produce un árbol (el árbol con el conjunto inicial de parámetros) y no se realizan iteraciones adicionales. Nosotros hemos configurado 36 que es el triple de variables explicativas que se le asignan al modelo.

Ahora pasamos a ver la predicción para cada vivienda mediante la primera salida de la función de cross-validación. Seleccionamos 4 variables explicativas de forma informativa, la variable respuesta SalePrice y la predicción en la columna Scored Labels.

Fold	Neighborhood	KitchenQual	1stFlrSF	GarageFinish	SalePrice	Scored Labels
5	Edwards	TA	248.7168	NA	118500	101477.24
2	Somerst	Gd	435.2544	Fin	182000	206832.80
6	NAmes	TA	201.168	RFn	155000	151916.81
9	NridgHt	Ex	563.2704	Fin	437154	413613.84
4	BrkSide	TA	187.7568	Unf	89000	81833.18
3	CollgCr	Gd	523.6464	RFn	297000	257617.84
8	Crawfor	TA	258.7752	Unf	188700	155136.88
4	CollgCr	Ex	395.9352	RFn	261500	266025.63
8	SWISU	TA	287.7312	Unf	139500	135961.16
8	Mitchel	Gd	370.9416	Unf	173000	158574.52
8	Sawyer	TA	332.8416	Unf	139000	136180.75
8	NridgHt	Gd	306.6288	RFn	284000	266366.34
0	SWISU	TA	240.792	Unf	91000	99758.02
0	Gilbert	Gd	274.6248	Fin	239900	263052.56
6	NWAmes	TA	657.1488	RFn	262500	262088.92
6	Crawfor	TA	334.0608	Unf	220000	172150.75
6	CollgCr	Gd	468.1728	RFn	214000	235613.31
2	CollgCr	Gd	609.6	Fin	305900	315116.41
3	NridgHt	Gd	473.6592	RFn	209500	279869.88

En este modelo de árboles de decisión reforzado, nos devuelve la predicción puntual sin la desviación estándar, ya que se tiene en cuenta en el modelo. Para poder observarlo mejor, lo vemos de forma visual generando un gráfico en Excel con las predicciones y el valor real de las observaciones de ejemplo.



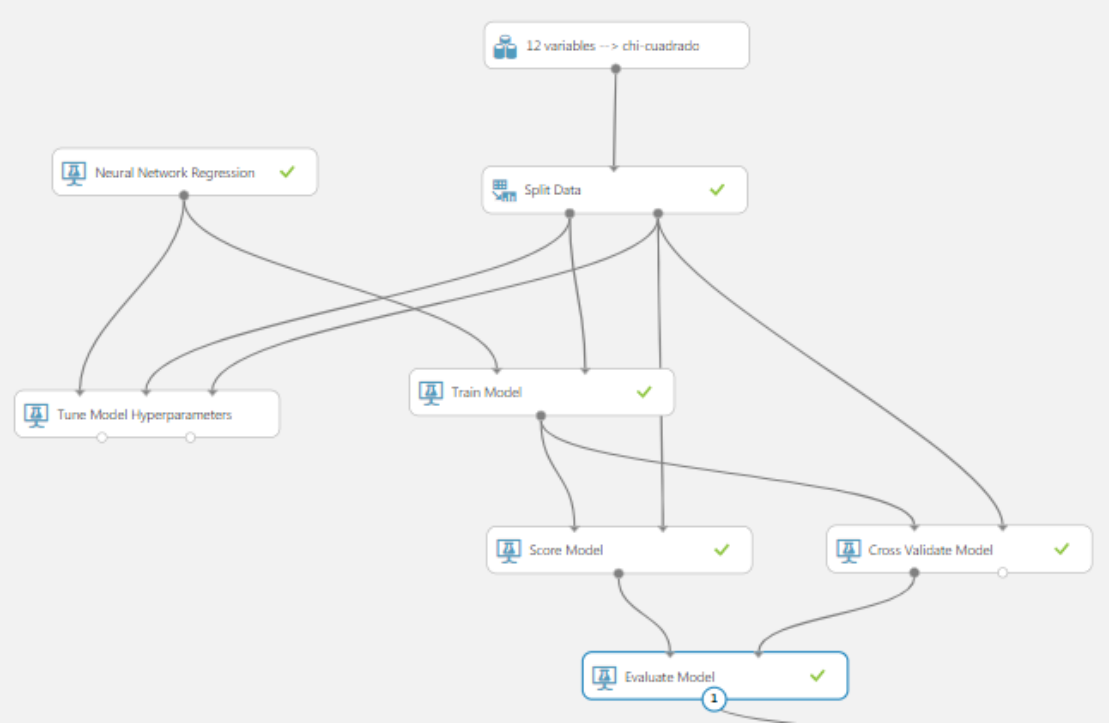
Este modelo utiliza los mismos datos que el modelo de árboles de decisión. A diferencia del modelo anterior, este corrige la predicción de la observación 15, pero dista mucho en la observación 19. Deberemos comprobar el valor del coeficiente de determinación para saber si globalmente es mejor modelo que el anterior.

En el siguiente apartado se analiza mediante el método cross-validation el ajuste del modelo a los datos, vemos las diez carpetas y el número de observaciones en cada una de ellas. Y seguidamente la suma de errores de cada una con su coeficiente de determinación.

Fold	Num	Mean Absolute Error	Root MSE	Relative Absolute Error	Relative Squared Error	R-Squared
0	43	14958.664	22532.015	0.337	0.177	0.823
1	44	25263.173	35425.045	0.495	0.295	0.705
2	44	27644.353	44176.092	0.404	0.271	0.729
3	44	24563.336	30133.976	0.405	0.156	0.844
4	43	19881.639	28112.855	0.405	0.176	0.824
5	44	25705.277	63444.563	0.404	0.351	0.649
6	44	20180.206	33349.552	0.342	0.182	0.818
7	44	15000.380	19308.655	0.257	0.071	0.929
8	44	26387.582	34482.531	0.548	0.262	0.738
9	44	23231.567	44252.812	0.381	0.326	0.674
Mean	438	22281.618	35521.810	0.398	0.227	0.773
St.Dev	438	4584.461	12702.309	0.081	0.088	0.088

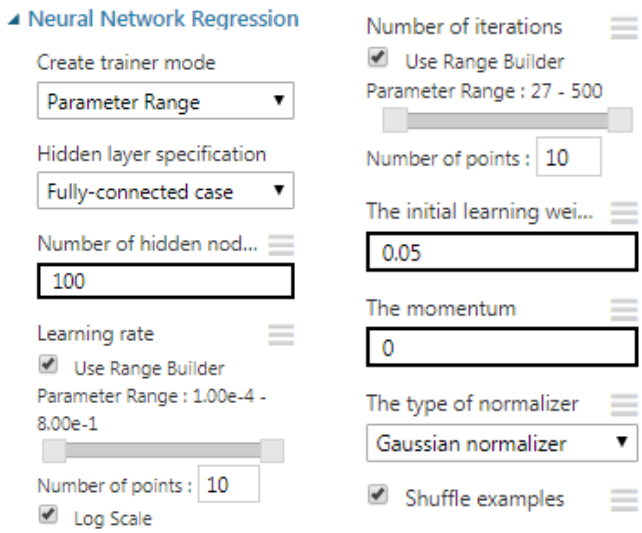
Con el modelo de Arboles de decisión reforzado obtenemos un ajuste de 0.773, es un valor elevado que nos indica que es un buen modelo. Si miramos los coeficientes de determinación de todas las carpetas vemos como hay dos con un valor inferior a 0.7, no es preocupante ya que no parece afectar demasiado a la media, la desviación estándar de todas las carpetas es de 0.088, lo que nos indica que el modelo no está sobreajustado a los datos.

3.5.5 Redes Neuronales



Por último, vamos a implementar el modelo de redes neuronales. Se trata de un modelo capaz de aprender de los datos y autoajustarse para crear predicciones. Utilizaremos, como en los modelos anteriores, la base de datos con 12 variables seleccionadas por el método chi-cuadrado, evaluaremos el modelo mediante la cross-validación y lo compararemos el ajuste del modelo por el método de r-cuadrado tradicional, sin hacerlo por carpetas.

Primeramente, configuramos el modo de entrenador de la función *Neural Network regression* de forma que acepte un rango de parámetros. De esta forma le indicamos que ha de hacer la selección de parámetros entre un rango previamente indicado, es un coste computacional mayor, pero nos ayuda a elegir entre la mejor selección de parámetros para la red neuronal.



Si le indicamos que nos haga una selección de parámetros es necesario preconfigurar los rangos entre los que se van a seleccionar los parámetros. Para ello configuramos la especifiación de capa oculta (Hidden layer specification) en modo completamente conectado (Fully-connected case). Esta opción crea un modelo que utiliza la arquitectura de red neuronal predeterminada, que para un modelo de regresión de red neuronal contiene estos atributos:

1. La red tiene exactamente una capa oculta.
2. La capa de salida está completamente conectada a la capa oculta y la capa oculta está completamente conectada a la capa de entrada.
3. Se pueden establecer la cantidad de nodos en la capa oculta (valor por defecto es 100).

Debido a que la cantidad de nodos en la capa de entrada está determinada por el número de características en los datos de entrenamiento, en un modelo de regresión solo puede haber un nodo en la capa de salida.

Seguidamente configuramos los rangos entre los que queremos que se seleccionen los parámetros. En este caso hemos decidido dejar el valor por defecto de nodos en la capa oculta, y ampliar al máximo los parámetros del ratio de aprendizaje, en escala logarítmica. Hacemos lo mismo para el numero de iteraciones, seleccionando 10 puntos de entre 27 y 500 iteraciones. A continuación, dejamos el valor del peso inicial de aprendizaje por nodo en su valor por defecto (0.05) y le indicamos el valor para aplicar durante el aprendizaje como un peso en los nodos de las iteraciones anteriores, es decir, el momentum a 0.

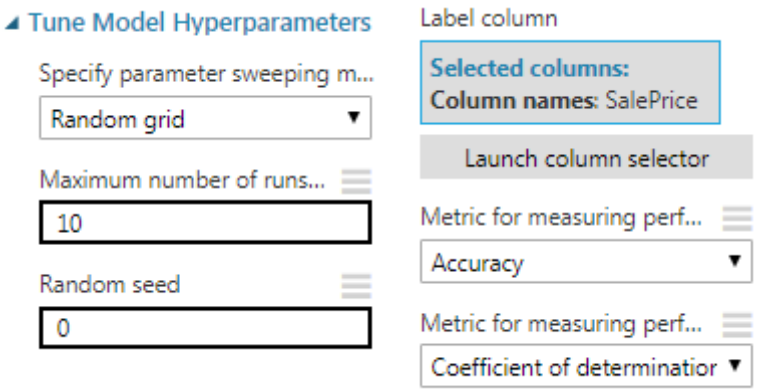
Por último seleccionamos el tipo de normalizador que queremos para la normalización de las características. En este apartado podemos elegir entre:

- Normalización de binning : Binning crea grupos de igual tamaño y luego normaliza cada valor en cada grupo para dividir por el número total de grupos.
- Normalizador gaussiano : la normalización gaussiana vuelve a escalar los valores de cada entidad para que tenga media 0 y varianza 1. Esto se hace calculando la media y la varianza de cada característica, y luego, para cada instancia, restando el valor medio y dividiendo por la raíz cuadrada de la varianza (la desviación estándar).
- Normalizador Min-Max: la normalización mínima maximiza linealmente cada característica al intervalo [0,1].
- No normalizar : no se realiza la normalización.

Para nuestro caso seleccionaremos el normalizador gaussiano, ya que después de varias pruebas es el que más se ajusta a los datos reales.

Una vez configurado el módulo de la red neuronal, es el momento de adjuntarle la función que selecciona los hiperparametros para nuestro modelo. Para ello agregamos el módulo *Tune Model Hyperparamete*. El módulo crea y prueba varios modelos, utilizando diferentes combinaciones de configuraciones, y compara las métricas de

todos los modelos para obtener la combinación de configuraciones. Básicamente, realiza un barrido de parámetros sobre la configuración de parámetros especificada, y aprende un conjunto óptimo de hiperparámetros , que pueden ser diferentes para cada árbol de decisión, conjunto de datos o método de regresión específico. Seleccionaremos la siguiente configuración:



.

En el primer campo, se selecciona como queremos que se encuentren los parámetros. En él podemos escoger entre tres opciones:

- Selección de parámetros aleatoria: esta opción entrena un modelo usando un número determinado de iteraciones. Se especifica un rango de valores para iterar, y el módulo usa un subconjunto elegido al azar de esos valores. Los valores se eligen con reemplazo, lo que significa que los números elegidos previamente al azar no se eliminan del conjunto de números disponibles. Por lo tanto, la posibilidad de que se seleccione cualquier valor permanece igual en todos los pases.
- Selección en forma de cuadrícula: esta opción crea una matriz o cuadrícula que incluye todas las combinaciones de parámetros en el rango de valores que especifique. Cuando comienza a sintonizar con este módulo, se entrena a varios modelos usando combinaciones de estos parámetros.
- Todas las combinaciones: la opción de usar todos los parámetros, significa exactamente eso, se prueban todas y cada una de las combinaciones. Esta opción puede considerarse la más completa, pero requiere más tiempo.
- Cuadrícula aleatoria: si selecciona esta opción, se calcula la matriz de todas las combinaciones y se toman muestras de los valores de la matriz, a lo largo del número de iteraciones que especificó.

Nosotros seleccionamos la opción cuadrícula aleatoria (Random Grid) ya que es un método muy eficiente y seleccionamos 10 iteraciones. Especificamos que la variable respuesta es la columna “SalePrice” y escogemos los métodos para las medidas de clasificación y regresión:

Métricas utilizadas para la clasificación:

- **Acuracy**: La proporción de resultados verdaderos a casos totales.
- Precisión: La proporción de resultados verdaderos a resultados positivos.
- **Recall**: La fracción de todos los resultados correctos sobre todos los resultados.
- **F-score**: Medida que equilibra la precisión y el recuerdo.
- **AUC**: Un valor que representa el área bajo la curva cuando los falsos positivos se trazan en el eje x y los verdaderos positivos se trazan en el eje y.
- **Average log loss**: La diferencia entre dos distribuciones de probabilidad: la verdadera y la del modelo.
- **Train log loss**: La mejora proporcionada por el modelo sobre una predicción aleatoria.

En este ejemplo seleccionaremos la medida de accuracy ya que nos deja ver una medida de proporción que nos interesa para comparar con el resto.

Métricas utilizadas para la regresión:

- **Error absoluto medio**: Promedia todo el error en el modelo, donde error significa la distancia del valor predicho del valor verdadero. A menudo abreviado como MAE .
- **Raíz de error cuadrático medio:** Mide el promedio de los cuadrados de los errores, y luego toma la raíz de ese valor. A menudo abreviado como RMSE
- **Error absoluto relativo:** Representa el error como un porcentaje del valor verdadero.
- **Error cuadrado relativo:** Normaliza el error cuadrado total dividiéndolo entre el error cuadrado total de los valores predichos.
- **Coeficiente de determinación:** Un solo número que indica qué tan bien los datos se ajustan a un modelo. Un valor de 1 significa que el modelo coincide exactamente con los datos; un valor de 0 significa que los datos son aleatorios o no pueden ajustarse al modelo. A menudo se denomina r 2 , R 2 o r-cuadrado .

En esta opción seleccionaremos el método del coeficiente de determinación ya que es el más utilizado para calcular la bondad del ajuste de la regresión.

Una vez configurado el módulo, lo ejecutamos. Nos devuelve dos outputs, el primero se trata de una tabla con las 10 iteraciones buscando los mejores parámetros, en ella podemos ver la tasa de aprendizaje, la función y el número de iteraciones que define azure para el modelo. En las columnas posteriores se detallan los errores de cada iteración y el ajuste mediante el coeficiente de determinación. Obviamente a nosotros nos interesa el que sus errores sean mínimos y el coeficiente mayor.

Learning rate	LossFunction	Number of iterations	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	R-squared
0.001482	CrossEntropy	311	29064.062	40805.696	0.518	0.303	0.697
0.000603	CrossEntropy	453	32856.841	44527.069	0.586	0.361	0.639
0.01	CrossEntropy	216	38955.201	50963.407	0.695	0.473	0.527
0.01	SquaredError	264	38967.008	50968.626	0.695	0.473	0.527
0.01	CrossEntropy	264	38967.008	50968.626	0.695	0.473	0.527
0.01	SquaredError	500	38974.417	50969.751	0.695	0.473	0.527

En el segundo output del módulo, nos presenta directamente el resultado de la mejor iteración de parámetros. Simplemente lo que deberemos hacer es seleccionar los parámetros que podemos modificar e implementarlos como parámetros en la función de red neuronal que teníamos definida. A continuación se muestra la tabla que devuelve el segundo output de la función *Tune Model Hyperparamete.*

Neural Network Regressor

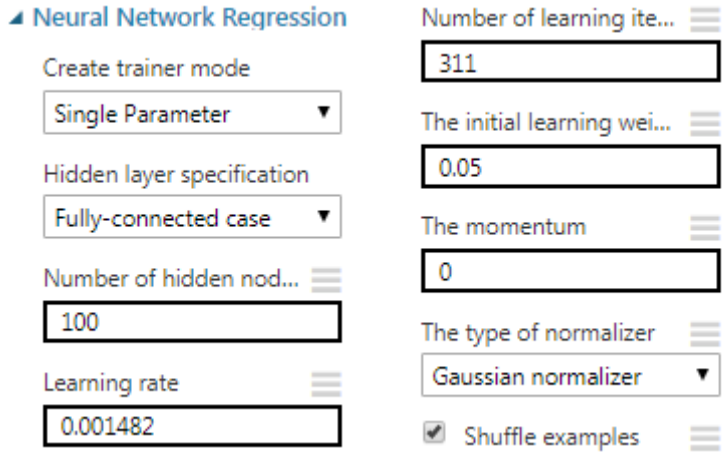
Settings

Setting	Value
Is Initialized From String	False
Is Classification	False
Initial Weights Diameter	0.05
Learning Rate	0.00148228001513336
Loss Function	CrossEntropy
Momentum	0
Neural Network Definition	
Data Normalizer Type	Gaussian
Number Of Input Features	77
Number Of Hidden Nodes	System.Collections.Generic.List`1[System.Int32]
Number Of Iterations	311
Number Of Output Classes	1
Shuffle	True
Allow Unknown Levels	False

Como podemos ver, el número de iteraciones y el valor de la tasa de aprendizaje corresponde con el de mayor ajuste de la tabla anterior. Estos son los parámetros que hemos de definir en nuestra red neuronal ya que son los que mejor pueden representar los datos a través de los nodos de la red neuronal.

También nos indica que el tipo de normalizador que mejor se adapta es el gaussiano, como ya indicábamos en el apartado anterior.

Finalmente, introducimos los parámetros encontrados en el módulo donde se configura la red neuronal y ejecutamos para ver los resultados.



Una vez implementado y ejecutado, pasamos a analizar los resultados obtenidos. Como hemos comentado anteriormente, en las redes neuronales compararemos los ajustes obtenidos mediante el r-cuadrado del modelo y la función cross-validation que divide los datos en carpetas para evitar el sobre ajuste del modelo.

Para llevarlo a cabo, dividimos la salida del modelo entrenado en dos módulos diferentes, Score model y Cross-validation model. La primera nos devuelve una tabla con los resultados de la predicción para cada individuo y la segunda, nos devuelve también la predicción por observación y, además las medidas de ajuste del modelo por carpeta (como hemos visto en los modelos anteriores).

SalePrice	Scored Labels	Scored (Cross-Validation)	Fold	Diferencia
216837	235271.28	228406.56	7	0.03
153000	199344.19	144248.47	3	0.36
149000	190833.33	194096.02	9	-0.02
80000	141316.48	98566.03	4	0.53
110000	140843.95	82216.16	6	0.53
140000	152992.02	129779.73	8	0.17
259000	234437.28	184913.97	2	0.19
159000	139544.95	93841.41	1	0.29
113000	118046.61	97997.42	1	0.18
337500	369105.13	334071.34	7	0.10
135750	149202.28	116472.91	7	0.24
189000	183121.83	204010.50	1	-0.11
136500	177739.86	137618.16	0	0.29
120500	158639.16	78416.11	1	0.67
233170	238499.05	193724.95	1	0.19
148500	130864.38	142151.56	2	-0.08

En la tabla anterior podemos ver un ejemplo de 16 viviendas con las predicciones para cada una mediante los dos métodos de ajuste. La columna *Scored labels* contiene la predicción mediante el método tradicional, en cambio la tercera columna se trata de las predicciones a partir de organizar los datos en 10 carpetas y mezclarlos con los datos de test, en la siguiente columna podemos ver las carpetas en las que se han asignado esas observaciones. Por último se ha creado una columna en Excel con la diferencia entre las dos predicciones y divididas por el precio real de cada vivienda ([scored label – score(cross-validation)] / SalePrice).

A primera vista lo que se puede apreciar es que la mayor parte de los valores en la columna diferencia son positivos, esto quiere decir que el modelo predice valores mayores que cuando hacemos la validación cruzada. En cuanto a comparar los ajustes de las predicciones, de esta forma no podemos sacar ninguna conclusión ya que tenemos observaciones que se ajustan mejor con uno y observaciones que se ajustan mejor con el otro. Por lo que vamos a hacer una comparativa estadística entre los dos modelos, mediante la función *evaluate model*.

Redes neuronales › Evaluate Model › Evaluation results

Metrics

Mean Absolute Error	31030.330308
Root Mean Squared Error	42999.455067
Relative Absolute Error	0.553499
Relative Squared Error	0.33676
Coefficient of Determination	0.66324

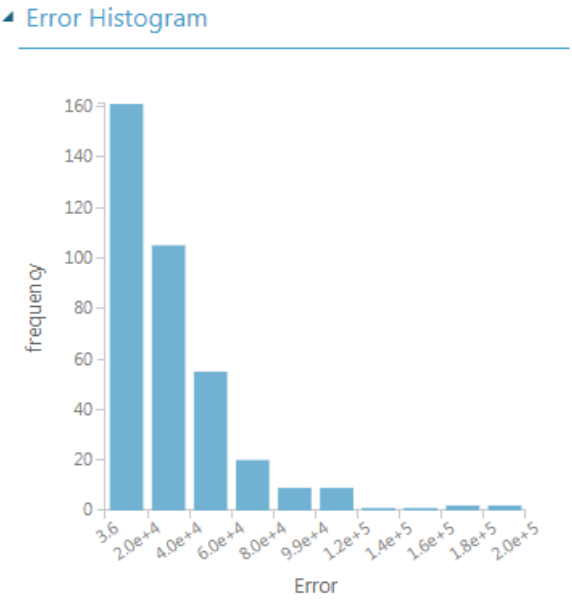
Metrics

Mean Absolute Error	34547.794595
Root Mean Squared Error	47222.60855
Relative Absolute Error	0.616242
Relative Squared Error	0.406157
Coefficient of Determination	0.593843

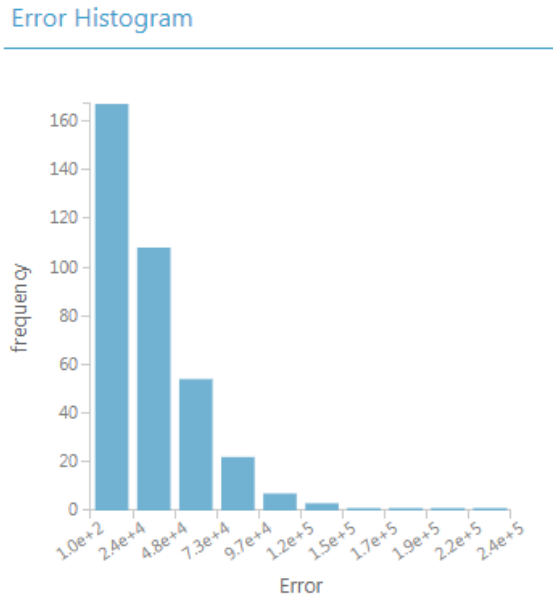
En el cuadro anterior se representa el output de la función *evaluate model,* se puede apreciar una comparativa entre el modelo normal y la cross-validación. En el modelo de puntuación vemos un MAE, Root MSE y los errores relativos y absolutos menores que en el modelo de cross-validación. Es por ello que, a pesar de no haber una gran diferencia entre los dos modelos, el coeficiente de determinación es mejor con el primer método.

A continuación se muestra la segunda parte del output de evaluación de modelos. En él se representa un histograma de los errores de los dos métodos.

Score Model



Cross-Validate Model



Como preveíamos, hay una ligera desviación por en la primera y segunda barra de errores (representan los errores más pequeños), en cambio a partir del tercer grupo de errores esto cambia y hay más frecuencia en el primer método que en el segundo.

Todo esto nos lleva a plantearnos si hay sobre ajuste en el modelo que el método de puntuación no adquiera. Por lo que vamos a analizar el ajuste de las carpetas del método de la cross-validación para ver qué ocurre.

Fold	Num	MAE	Root MSE	Relative Absolute Error	Relative Squared Error	R-squared
0	36	40675.253	57522.084	0.669	0.600	0.400
1	37	32082.589	53477.523	0.534	0.370	0.630
2	37	34567.867	45033.042	0.610	0.376	0.624
3	36	32877.455	41216.997	0.545	0.265	0.735
4	36	37824.511	46291.812	0.782	0.569	0.431
5	37	30065.599	39907.657	0.514	0.278	0.722
6	37	37361.066	51895.667	0.601	0.423	0.577
7	36	36976.232	51134.200	0.698	0.446	0.554
8	36	27913.974	36020.056	0.592	0.403	0.597
9	37	35228.764	45399.080	0.760	0.536	0.464
Mean	365	34557.331	46789.812	0.631	0.427	0.573
St. Dev	365	3880.472	6704.285	0.093	0.114	0.114

En la tabla anterior se ve claramente como hay una diferencia significativa entre las diferentes carpetas. Con una desviación estándar de 0.114, es la mayor de todos los modelos implementados. Esto puede ser debido a utilizar parámetros incorrectos, o simplemente ajustados para los datos de train y cuando se compara el modelo con la base de datos de test no sean correctos.

Con esta comparativa entre los dos métodos de ajuste, resulta evidente que el método de la cross-validación nos proporciona mucha más información acerca de la viabilidad del modelo, ya que podemos asegurarnos de que no esté sobreajustado para los datos de test.

En cambio, si nos fijásemos simplemente en el valor del método de puntuación (score model), probablemente hubiésemos validado el modelo (R-cuadrado de 0.66) y posteriormente si hubiésemos introducido nuevos datos no hubieran sido correctamente predichos.

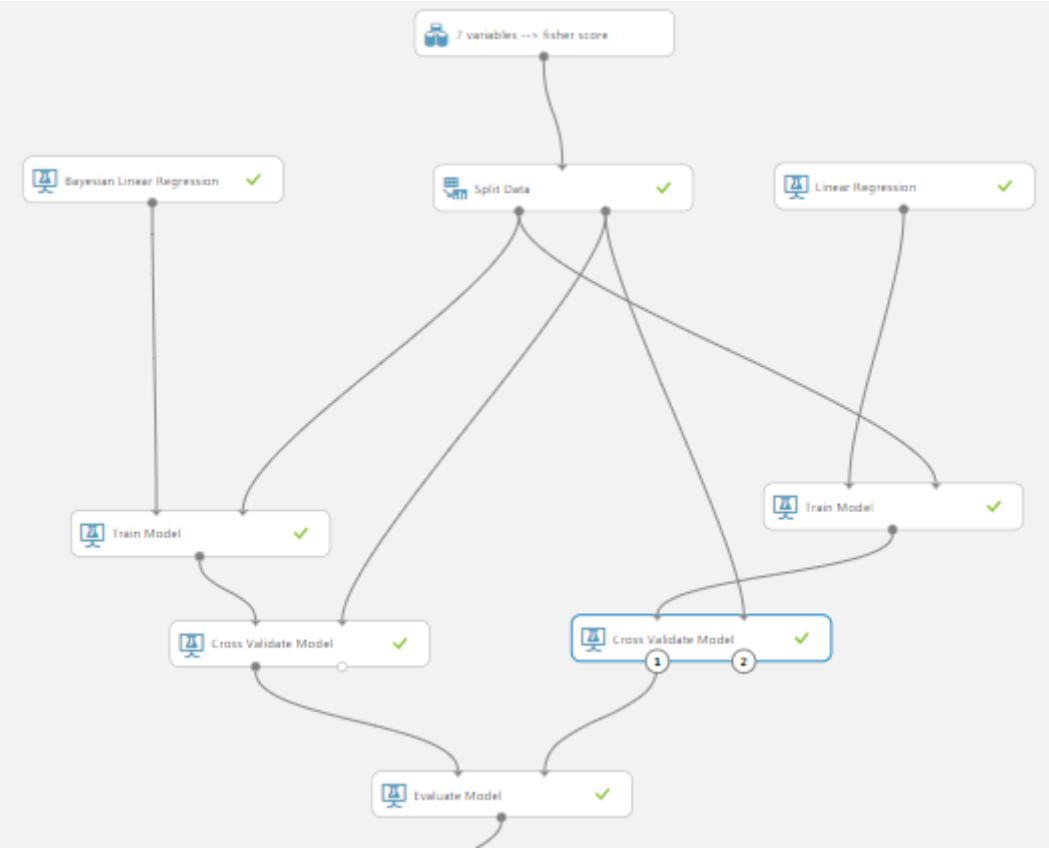
3.6 Análisis de Resultados

Después de realizar todos los modelos, es momento de analizar los resultados obtenidos y compararlos entre sí. Como hemos comentado en el pre-procesado de datos, se han creado tres bases de datos diferentes a partir de los métodos de selección de variables. En los ejemplos de los modelos predictivos simplemente hemos enseñado como implementar las soluciones de Azure a partir de una de ellas (selección de variables mediante el método chi-cuadrado). En este apartado compararemos los resultados obtenidos de la predicción con las otras bases de datos y comentaremos los resultados obtenidos.

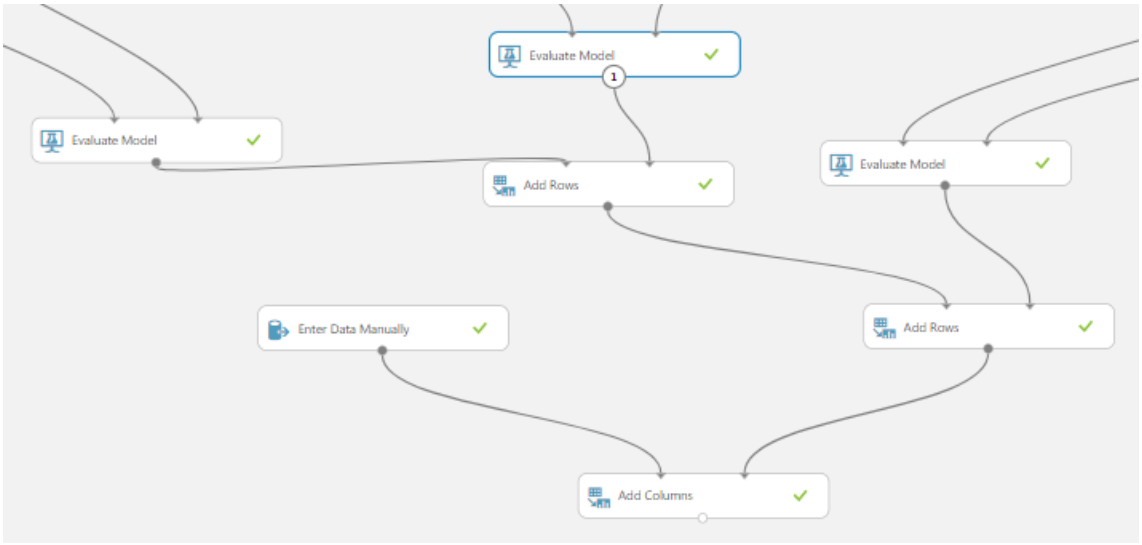
Destacar que las otras bases de datos se han predicho de la misma forma que la que hemos explicado en los ejemplos de los modelos predictivos anteriores, simplemente hemos ajustado los parámetros para conseguir el mayor ajuste posible. En el anexo se adjuntan las imágenes de la creación de los modelos para las distintas bases de datos.

3.6.1 Comparativa entre modelos con distintas variables explicativas

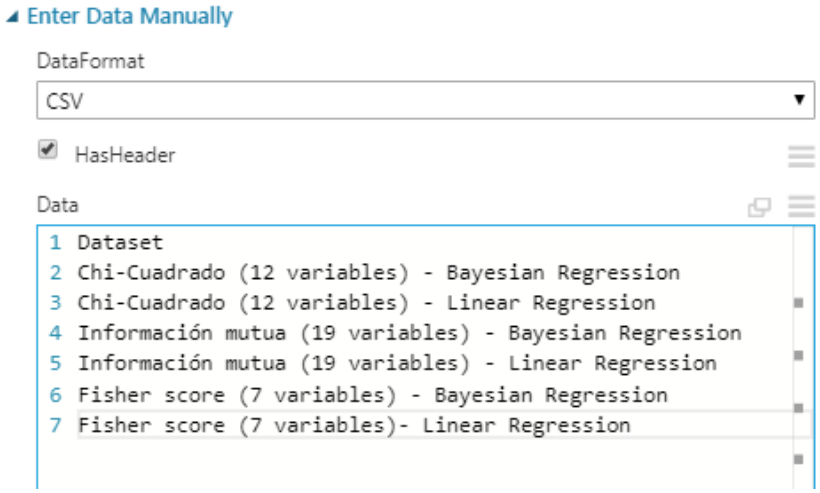
Empezaremos comparando los resultados obtenidos en el modelo de regresión lineal y bayesiana en las tres bases de datos. Se han creado ambos modelos en el mismo *workspace* para facilitar la comparación de resultados, por lo que en este apartado podremos comparar los resultados del modelo lineal y bayesiano con las diferentes variables explicativas con la finalidad de ver cuál se adapta mejor a los datos. A continuación, se muestra como se ha llevado a cabo el proceso con los dos modelos a la vez para la base de datos con siete variables seleccionadas por el método de Fisher.



Seguidamente, se juntan las evaluaciones de los modelos de las tres bases de datos ejecutadas por los métodos de regresión lineal y bayesiana mediante los módulos siguientes:



Como podemos ver en la anterior imagen, utilizamos las funciones “Add Rows” para agregar la información proveniente de las evaluaciones de los modelos. Estas evaluaciones contienen la información de los errores y los coeficientes de determinación para la base de datos en cuestión por ambos modelos. Seguidamente en el módulo “Enter Data Manually”, le agregamos la cabecera para saber la fila a que modelo hace referencia.



Finalmente obtenemos una tabla de comparativa entre los dos modelos de regresión mediante las tres bases de datos que contienen las diferentes variables explicativas.

En los apartados posteriores entraremos a detallar esta comparativa y extraeremos conclusiones sobre cómo afecta el método de selección de variables o los modelos escogidos para predecir los precios de las viviendas.

3.6.2 Modelo regresión bayesiana vs regresión lineal

Una vez hemos visto el procedimiento para llevar a cabo la comparativa entre los modelos y las bases de datos con las distintas variables explicativas, pasamos a analizar los resultados obtenidos.

Dataset	Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	R-squared
Chi-Cuadrado (12 variables) - Bayesian Regression	4275.187	21113.017	29699.882	0.350	0.121	0.879
Chi-Cuadrado (12 variables) - Linear Regression	Infinity	22148.879	31745.090	0.367	0.138	0.862
Información mutua (19 variables) - Bayesian Regression	4257.272	20900.953	28405.624	0.346	0.111	0.889
Información mutua (19 variables) - Linear Regression	Infinity	22699.986	31670.191	0.376	0.137	0.863
Fisher score (7 variables) - Bayesian Regression	4882.144	41614.690	57953.659	0.690	0.460	0.540
Fisher score (7 variables)- Linear Regression	Infinity	38432.491	55468.263	0.637	0.422	0.578

Como podemos ver en la tabla anterior, obtenemos un coeficiente de determinación muy alto en los dos modelos con 12 y 19 variables explicativas. En cambio, si simplemente introducimos las 7 variables predictoras que nos hace referencia el método de Fischer, obtenemos unos errores de casi el doble que con las otras variables, lo que nos lleva a encontrar un ajuste muy bajo del modelo.

Por tanto, obtenemos que el mejor modelo es el de la regresión bayesiana ya que su coeficiente de determinación para ambas variables explicativas es el más elevado. Pero no podemos descartar la regresión lineal, ya que con valores en el r-cuadrado por encima de 0.85, determinamos que es un buen modelo.

3.6.3 Decision forest vs Boosted decision tree regressions

En este apartado vamos a evaluar la capacidad predictiva de los modelos de decisión basados en ramificaciones. Para ello, seguimos los pasos del apartado anterior y extraemos una tabla de resultados con los errores y los coeficientes de determinación de cada uno por las tres bases de datos con las diferentes variables explicativas.

Dataset	Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	R-Squared
Chi Cuadrado (12 var) - decision forest regression	5154.895	22487.963	36038.931	0.397	0.220	0.780
Chi Cuadrado (12 var) - boosted decision tree regresion	Infinity	22303.816	37556.277	0.393	0.239	0.761
Informacion mutua (19 var) - decision forest regression	5370.649	24457.524	38885.667	0.431	0.256	0.744
Informacion mutua (19 var) - boosted decision tree regresion	Infinity	23203.352	39192.412	0.409	0.260	0.740
Fisher Score (7 var) - decision forest regression	5446.729	30091.405	47704.513	0.531	0.385	0.615
Fisher Score (7 var) - boosted decision tree regresion	Infinity	30548.806	47425.684	0.539	0.381	0.619

Como podemos ver en la tabla de resultados, la selección de variables por el método de Fisher nos devuelve el peor ajuste por ambos modelos. En cambio, a diferencia del apartado anterior, ahora encontramos el mejor ajuste en el coeficiente de determinación con la selección por Fisher en el modelo de árboles de decisión. Con un valor de 0.78, es el mayor de esta tabla a pesar que no dista mucho de los otros valores en los que si miramos la selección de variables primera y segunda el menor valor del coeficiente de determinación es 0.74, lo que nos indica que no son malos modelos, pero el más ajustado a los datos se consigue con la selección de 12 variables y mediante el método de árboles de decisión, con los parámetros explicados en el ejemplo del apartado 4.5.3.

3.6.4 Redes neuronales

En este apartado compararemos los resultados obtenidos por el método de las redes neuronales, como hemos comentado en el ejemplo 4.5.5, en este modelo se compara la capacidad de ajuste del modelo de cross-validación que tiene en cuenta el sobreajuste de los datos. Por lo tanto, la tabla posterior representa el ajuste obtenido mediante el modelo de redes neuronales para las tres bases de datos con la selección de variables de cada una.

Dataset	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	R-squared
Chi Cuadrado (12 var) - Neural Network Score model	31030.33031	42999.45507	0.553499	0.33676	0.66324
Chi Cuadrado (12 var) - Neural Network Cross-validation	34547.7946	47222.60855	0.616242	0.406157	0.593843
Informacion mutua (19 var) - Neural Network Score model	35084.85374	46656.02098	0.625821	0.396469	0.603531
Informacion mutua (19 var) - Neural Network Cross-validation	36035.64411	50417.27382	0.642781	0.46297	0.53703
Fisher Score (7 var) - Neural Network Score model	39600.34985	55499.26293	0.706366	0.561007	0.438993
Fisher Score (7 var) - Neural Network Cross-validation	43242.61248	62378.11495	0.771334	0.708694	0.291306

Como podemos ver, claramente hay diferencias en el ajuste del modelo entre el método de cross-validación y el de puntuación. A simple vista, se ve como el método de puntuación siempre indica un mejor ajuste, pero no lo tendremos en cuenta en las conclusiones ya que la diferencia entre los dos métodos nos indica que puede estar sobre ajustado.

Fijándonos en los coeficientes de determinación por el método de cross-validación vemos como la base de datos con 7 variables explicativas, al igual que en los modelos anteriores, obtiene la peor puntuación. Por lo que resulta evidente que las variables seleccionadas no representan completamente la variable respuesta.

Ahora bien, las bases de datos de 12 y 19 variables obtienen simplemente un coeficiente de determinación de 0.59 y 0.53 respectivamente, lo que tampoco es demasiado buen ajuste. De hecho, es el peor de todos los métodos implementados. Como hemos visto en el ejemplo, hemos seleccionado los parámetros a través de la función que nos proporciona Azure de hiperparámetros. Por lo que este pequeño coeficiente de determinación puede ser debido a que la selección de parámetros del propio software no funcione correctamente, o sea necesario introducir más variables explicativas para conseguir explicar mejor la respuesta.

4. Conclusiones

En este trabajo hemos podido comprobar cómo en la nube se pueden implementar soluciones para el análisis y explotación de los datos. En concreto, trabajando con la plataforma Microsoft Azure machine learning hemos visto que no necesitamos una gran infraestructura (a pesar de haber trabajado con una base de datos relativamente pequeña), ni un ordenador muy potente para llevar a cabo predicciones sobre el precio de una vivienda con a través de una serie de variables explicativas.

Por lo tanto, considero que es una herramienta muy útil ya que se permite el uso desde cualquier dispositivo conectado a internet, sin necesidad de tener instalado ningún software en él. Además, tener los datos en la nube es una gran ventaja a la hora de trabajar en diferentes sitios, ya que simplemente accediendo con un usuario y contraseña está disponible para consultar los resultado obtenidos o continuar desarrollando.

Hemos visto, mediante la implementación del caso empírico que la realización de un modelo de machine learning no es complicado en la faceta de *picar código*, a pesar de que la plataforma lo permita (en R y Phyton), si se quiere evitar o no se sabe programar no hay problema, ya que se pueden llevar a cabo experimentos del mismo modo que si introdujéramos el código *a mano.*

Al ser una plataforma tan completa, hemos encontrado todos los modelos que *a priori* queríamos implementar para tratar de predecir el precio de las viviendas. Como estadístico, en mi opinión está un poco limitado a la hora de poder tratar la información que nos producen los outputs. No es posible crear varios modelos de gráficos, sino que Azure mismo selecciona el mejor según sus criterios para representar los datos. Esto se puede conseguir introduciendo el código de R necesario, pero ya dejaría de tener el valor añadido que se le proporciona al tener todo en módulos. De igual forma, si lo que se quiere es llevar a cabo un buen proyecto, sabiendo programar en R o Phyton, Azure proporciona todas las herramientas que necesitaríamos si lo hiciésemos de forma tradicional, pero con los beneficios de tener la solución en la nube.

Comentar también que Microsoft Azure es un software mucho más amplio que solo su parte de machine learning, pero se tratan de elementos separados y por ese motivo en este trabajo solo se habla de dicha parte. También destacar que se ha trabajado en todo momento con la versión gratuita, en la que teníamos un tiempo de procesado y cantidad de almacenamiento limitada, pero para este caso, más que suficiente al tratarse de una base de datos de poco más de mil cuatrocientas filas.

En resumen, después de haber trabajado durante muchas horas con Azure Machine Learning, me parece una innovación muy buena tanto para estudiantes que están aprendiendo los modelos de aprendizaje automático, como para grandes empresas que lo que desean es pasar todas sus operaciones de tratamiento de datos a la nube. Ya que se trata de una herramienta muy fácil de utilizar, que te permite centrarte únicamente en el resultado de tus experimentos, sin tener grandes conocimientos de programación

y por ese motivo, sin temor a equivocarnos en la ejecución de los scripts. Simplemente escoger los parámetros adecuados para el modelo y ejecutar, en unos segundos tenemos una predicción de la variable respuesta a partir del método que hayamos utilizado.

5. Referencias

5.1 Bibliografía

1. Linear regression analysis. GAF Seber, A.J. Lee (2003). Hoboken, N.J
2. Big Data Now: 2012 Edition (2012). O'Reily Media Inc
3. Lior Rokach and Oded Maimon (2008). Data mining with decision trees: theory and applications. World Scientific
4. Theory and Applications of Artificial Neural Networks. Jian-Rong Chen (1991). Durham. Thesis, Durham University. Disponible en: <http://etheses.dur.ac.uk/6240/ >
5. Artificial Neural Networks as Models of Neural Inforation Processing (2017). Marcel van Gerven and Sander Bohte. Research Topic. Disponible en: <https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing>

5.2 Webgrafía

1. Telefónica Digital España (2018). LUCA Data Driven Decisions. [En línea]. Disponible en: <http://data-speaks.luca-d3.com/2017/11/tutorial-azureML-Titanic2.html>
2. Microsoft Azure (n.d.). *Cómo elegir algoritmos para Microsoft Azure Machine Learning.* [En línea]. Disponible en: <https://docs.microsoft.com/es-es/azure/machine-learning/studio/algorithm-choice >
3. Kaggle (n.d.). *Bases de datos para el tratamiento estadístico*. [En línea]. Disponible en: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>
4. Wescon Group (2015). *Beneficios del almacenamiento en la nube.* [En línea]. *Disponible en:* <http://blogmexico.comstor.com/beneficios-del-almacenamiento-en-la-nube>
5. Instituto de ingeniería del conocimiento (2016). El Big Data y la nube: los servicios Cloud. [En línea]. Disponible en: <http://www.iic.uam.es/innovacion/big-data-la-nube-servicios-cloud/>
6. Openinnova (2017). *Amazon Aws, Microsoft Azure, Google Cloud | Cual elegir?* [En linea]. Disponible en: <https://www.openinnova.es/amazon-aws-vs-microsoft-azure-vs-google-cloud-cual-elegir/>
7. Azure (n.d.). *Feature Selection modules*. [En línea]. Disponible en: <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/feature-selection-modules>
8. Rodrigo Perez Burgues (2015). *Analizando los datos y Azure Machine Learning.* [En línea]. Disponible en: <https://rodrigopb.wordpress.com/2015/04/08/analizando-los-datos-y-azure-machine-learning/>

9. Guillermo Julián (2018). *Las redes neuronales: qué son y por qué están volviendo.* [En línea]. Disponible en: <https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo>
10. Joaquín Amat Rodrigo (2017). *Análisis de Componentes Principales.* [En línea]. Disponible en: https://rstudio-pubs-static.s3.amazonaws.com/287787_1c53df3fcf6b432dbc775a91cb2090ce.html

6. Anexo

6.1 Galería de experimentos Azure

Todos los experimentos han sido subidos en la galería de Microsoft Azure, donde están disponibles para ver y modificar por los usuarios. Para ello simplemente se necesita una cuenta Microsoft.

- Pre-procesado de datos:

https://gallery.cortanaintelligence.com/Experiment/Preprocesado-de-los-datos

- Modelos de machine learning:

https://gallery.cortanaintelligence.com/Experiment/Modelos-de-regresi-n-REAL-STATE

https://gallery.cortanaintelligence.com/Experiment/Forest-and-tree-regression

https://gallery.cortanaintelligence.com/Experiment/Redes-neuronales

6.2 Información sobre las variables categóricas del caso empírico

MSSubClass: Identifies the type of dwelling involved in the sale.

20 1-STORY 1946 & NEWER ALL STYLES

30 1-STORY 1945 & OLDER

40 1-STORY W/FINISHED ATTIC ALL AGES

45 1-1/2 STORY - UNFINISHED ALL AGES

50 1-1/2 STORY FINISHED ALL AGES

60 2-STORY 1946 & NEWER

70 2-STORY 1945 & OLDER

75 2-1/2 STORY ALL AGES

80 SPLIT OR MULTI-LEVEL

85 SPLIT FOYER

90 DUPLEX - ALL STYLES AND AGES

120 1-STORY PUD (Planned Unit Development) - 1946 & NEWER

150 1-1/2 STORY PUD - ALL AGES

160 2-STORY PUD - 1946 & NEWER

180 PUD - MULTILEVEL - INCL SPLIT LEV/FOYER

190 2 FAMILY CONVERSION - ALL STYLES AND AGES

Neighborhood: Physical locations within Ames city limits

Blmngtn	Bloomington Heights	NoRidge	Northridge
Blueste	Bluestem	NPkVill	Northpark Villa
BrDale	Briardale	NridgHt	Northridge Heights
BrkSide	Brookside	NWAmes	Northwest Ames
ClearCr	Clear Creek	OldTown	Old Town
CollgCr	College Creek	SWISU	South & West of Iowa State
Crawfor	Crawford	Sawyer	Sawyer
Edwards	Edwards	SawyerW	Sawyer West
Gilbert	Gilbert	Somerst	Somerset
IDOTRR	Iowa DOT and Rail Road	StoneBr	Stone Brook
MeadowV	Meadow Village	Timber	Timberland
Mitchel	Mitchell	Veenker	Veenker
Names	North Ames		

OverallQual: Rates the overall material and finish of the house

10	Very Excellent	5	Average
9	Excellent	4	Below Average
8	Very Good	3	Fair
7	Good	2	Poor
6	Above Average	1	Very Poor

ExterQual: Evaluates the quality of the material on the exterior

Ex	Excellent	Fa	Fair
Gd	Good	Po	Poor
TA	Average/Typical		

BsmtQual: Evaluates the height of the basement

Ex	Excellent (100+ inches)	Fa	Fair (70-79 inches)
Gd	Good (90-99 inches)	Po	Poor (<70 inches
TA	Typical (80-89 inches)	NA	No Basement

KitchenQual: Kitchen quality

Ex	Excellent	Fa	Fair
Gd	Good	Po	Poor
TA	Typical/Average		

FireplaceQu: Fireplace quality

Ex	Excellent - Exceptional Masonry Fireplace
Gd	Good - Masonry Fireplace in main level
TA	Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement
Fa	Fair - Prefabricated Fireplace in basement
Po	Poor - Ben Franklin Stove
NA	No Fireplace

GarageType: Garage location

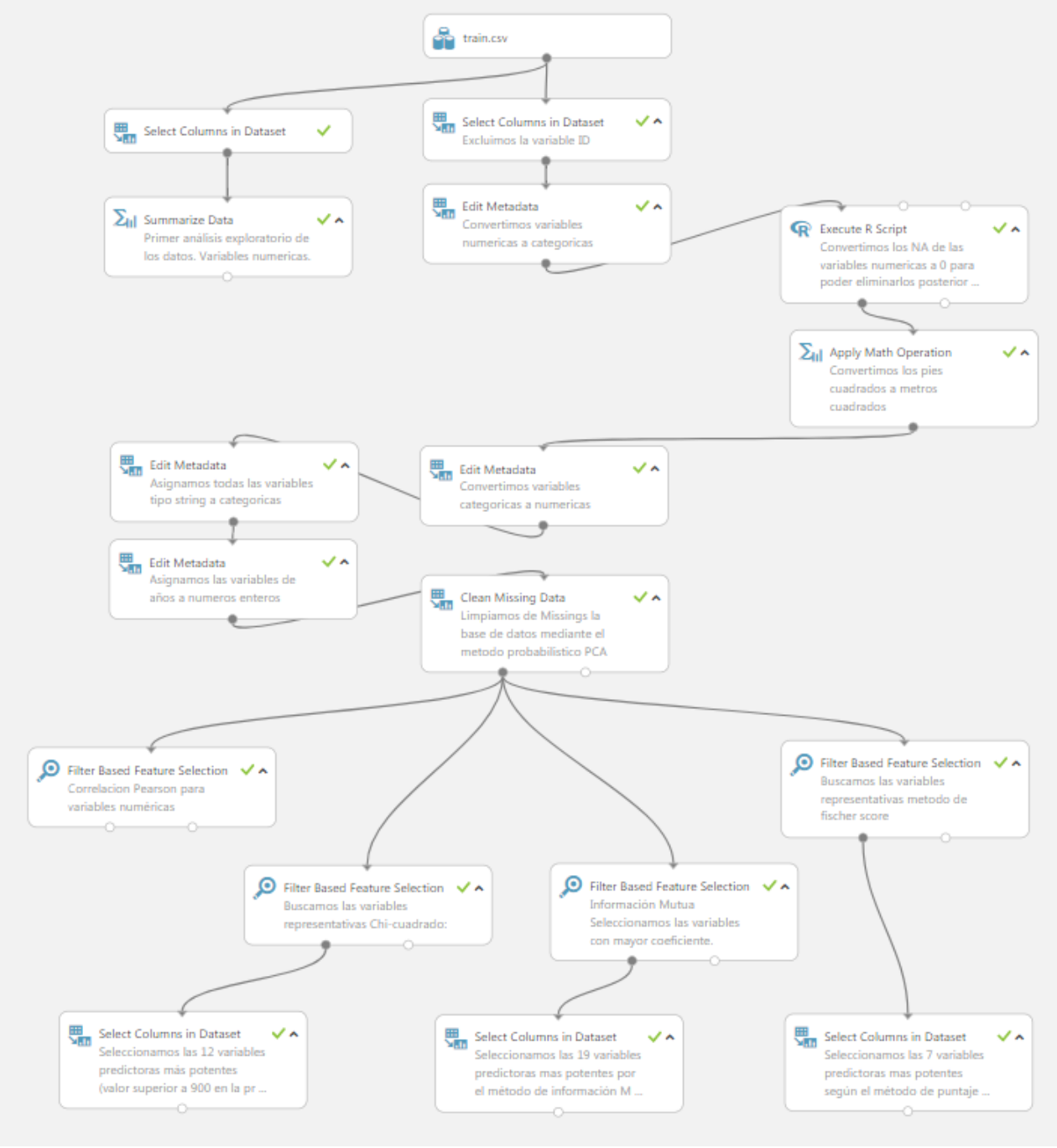
2Types	More than one type of garage	BuiltIn	Built-In (Garage part of house - typically has room above garage)
Attchd	Attached to home	CarPort	Car Port
Basment	Basement Garage	Detchd	Detached from home
		NA	No Garage

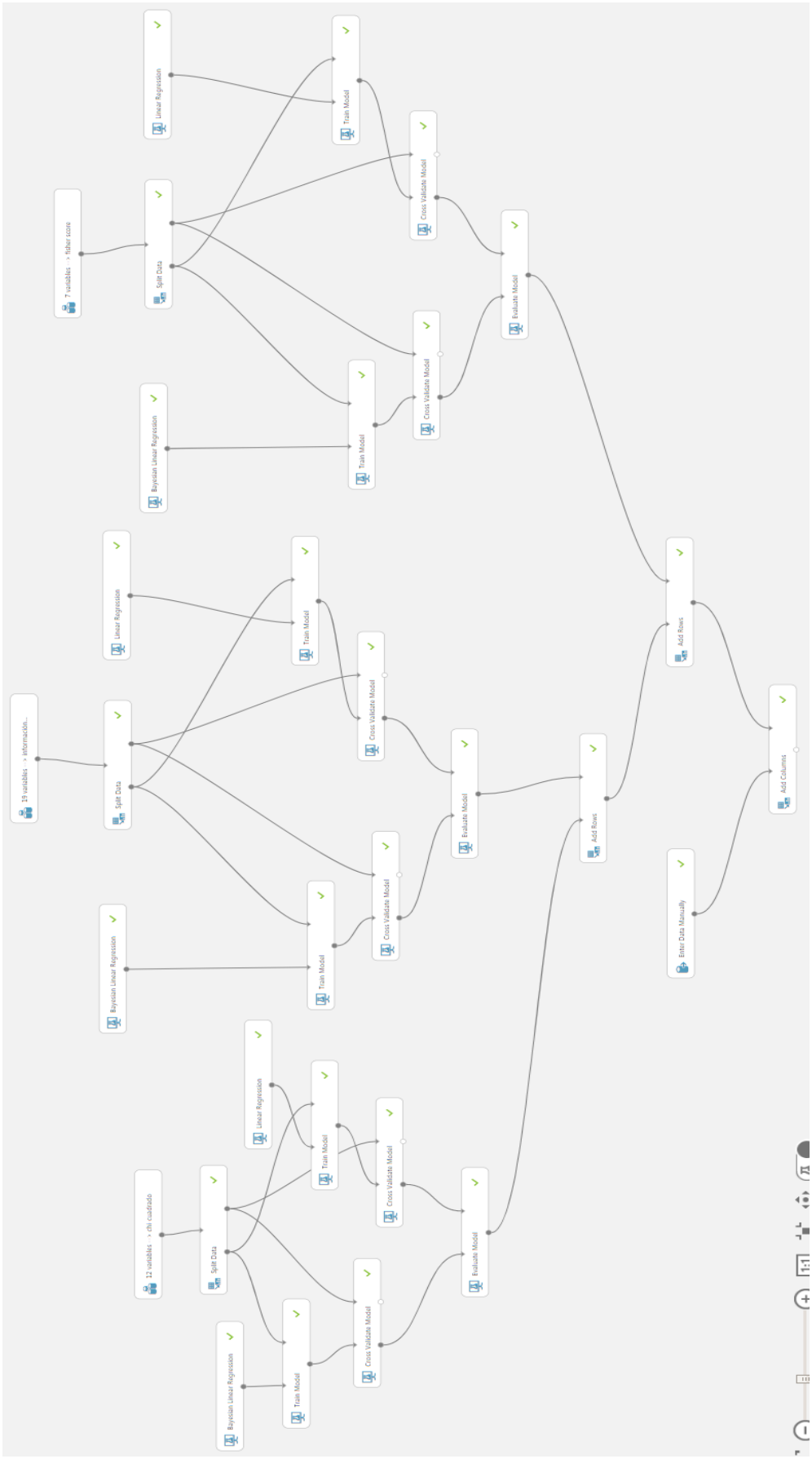
GarageFinish: Interior finish of the garage

Fin	Finished	Unf	Unfinished
RFn	Rough Finished	NA	No Garage

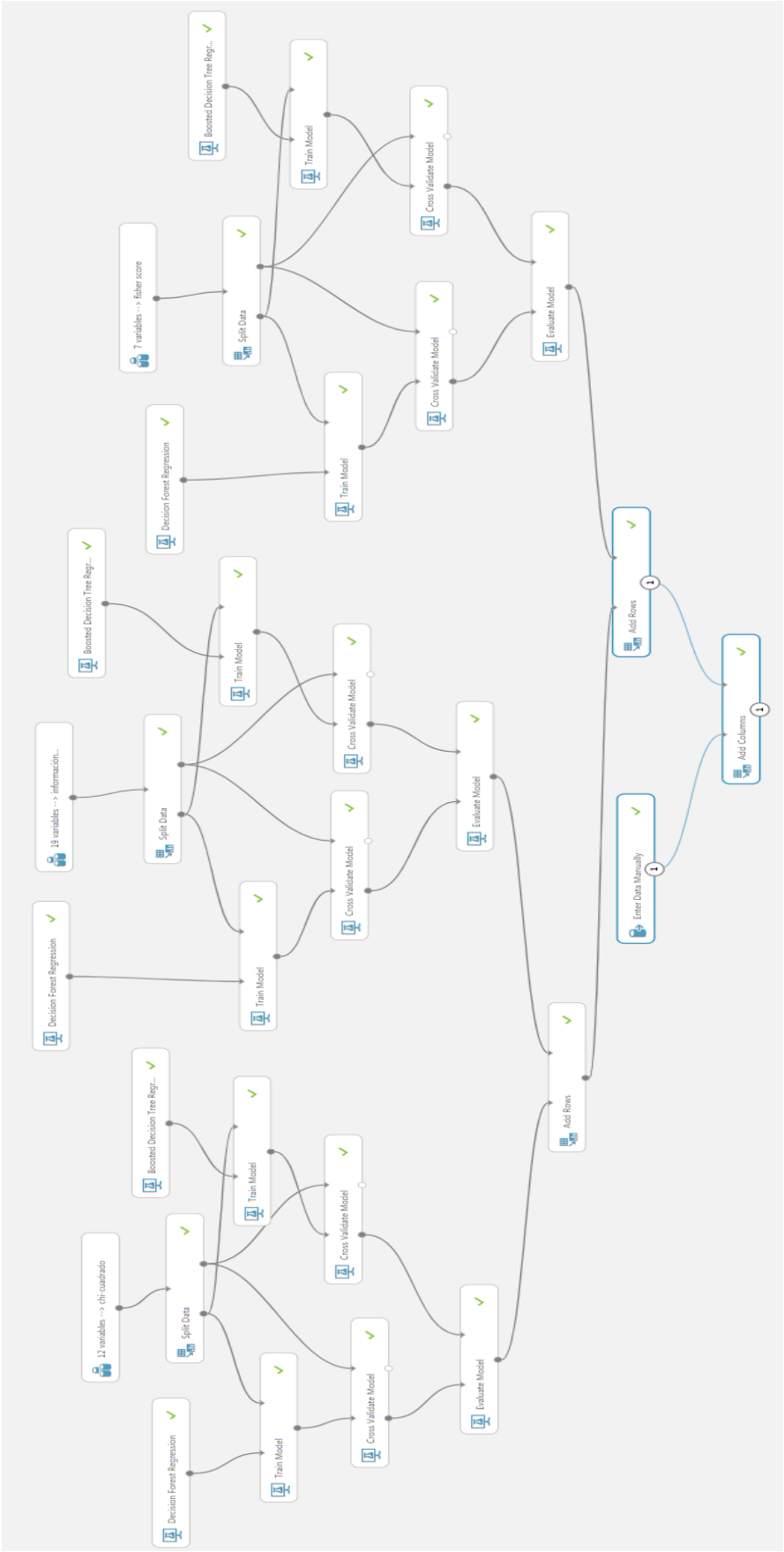
6.3 Workspace de azure

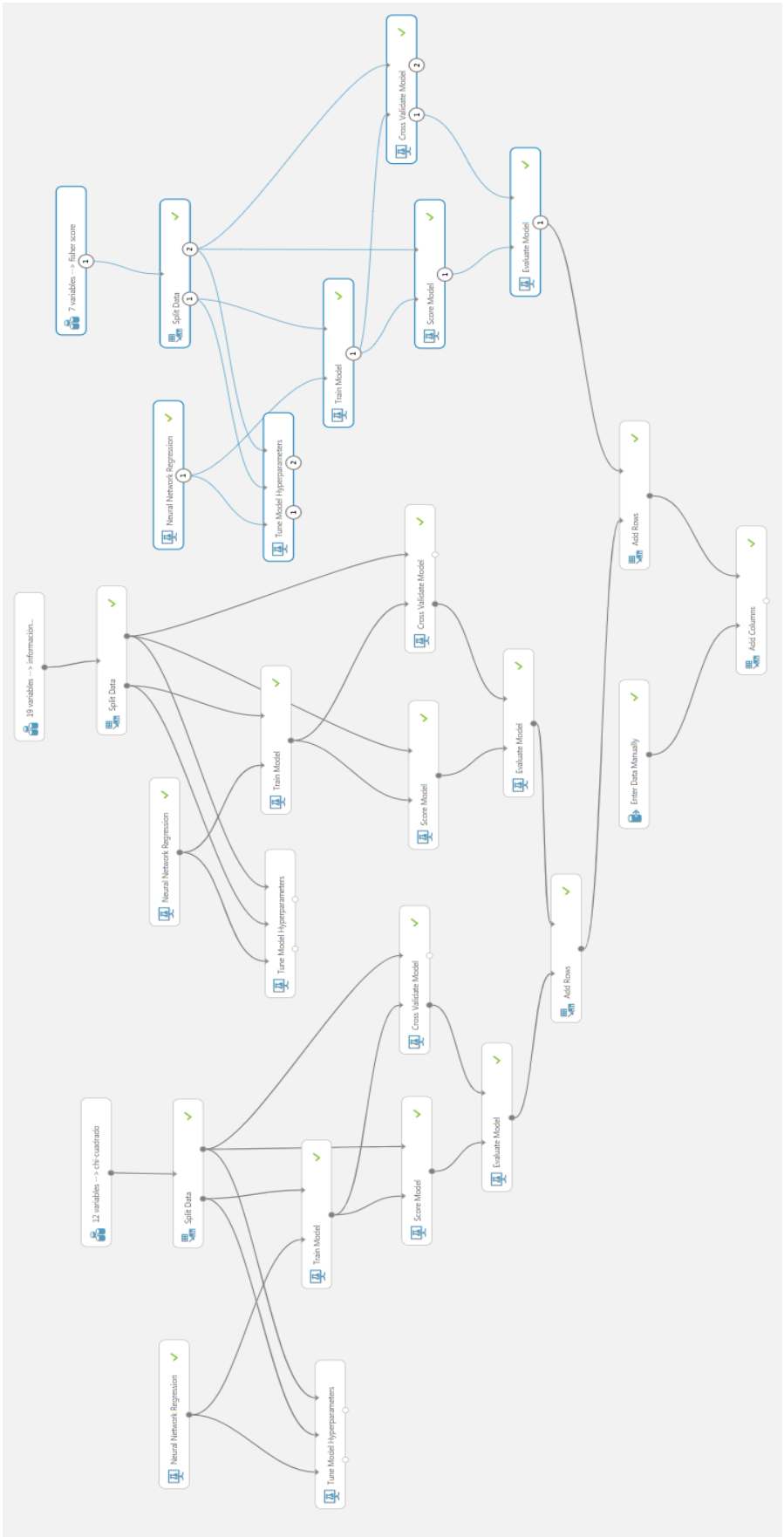
Pre-procesado de datos




Linear Regression
Train Model
7 variables - > Rother score
Split Data
Cross Validate Model
Evaluate Model
Bayesian Linear Regression
Train Model
Cross Validate Model
Linear Regression
Train Model
Add Rows
19 variables - > Información...
Split Data
Cross Validate Model
Evaluate Model
Add Rows
Add Columns
Bayesian Linear Regression
Train Model
Cross Validate Model
Enter Data Manually
Linear Regression
Train Model
12 variables - > chi cuadrado
Split Data
Cross Validate Model
Evaluate Model
Bayesian Linear Regression
Train Model
Cross Validate Model

Decision Forest and boosted decision tree regression




7 variables --> fisher score
Split Data
Cross Validate Model
Train Model
Score Model
Evaluate Model
Neural Network Regression
Tune Model Hyperparameters
19 variables --> información...
Add Rows
Add Columns
Enter Data Manually
12 variables --> chi cuadrado