

Modelos de regresión

1. Algoritmos: regresión lineal, árboles, k-vecinos, ensembles
2. Métricas específicas de regresión
3. Introducción a Big ML
4. Caso en Big ML

Notación ML supervisado

El *dataset* consistirá en una tabla **X** con la información de los predictores y un vector **y** con los valores de la variable objetivo.

- La tabla **X** tiene:
 - **Filas:** cada fila representa un ejemplo del *dataset*
 - **Columnas:** cada columna es un atributo ó predictor que se considera útil para explicar **y**
- El vector **y** es:
 - **En regresión:** una variable que toma números reales ó en un continuo. *Ejemplo: 0.123, 2.34, 232.21 , ...*
 - **En clasificación:**
 - **Binaria:** los valores 0 y 1
 - **Multiclase:** los valores 1,2,3,..., K donde K es el número de clases

Algoritmos paramétricos y no paramétricos

Los **algoritmos paramétricos** son aquellos que **encuentran el mejor valor posible de los parámetros** usando el conjunto de datos de entrenamiento y una vez encontrados no utilizan el conjunto de datos de entrenamiento para realizar las predicciones.

Algoritmos no paramétricos son aquellos algoritmos de aprendizaje que necesitan el conjunto de datos para realizar una nueva predicción:

- Necesitan tener el conjunto de datos de entrenamiento disponible
- La memoria crece linealmente a cómo crecen los datos en el conjunto de datos de entrenamiento.

Regresión lineal

En matemáticas y estadística, los modelos más simples son los lineales, tanto por motivos teóricos de diseño como de eficiencia y realizabilidad computacional. Es por esto que suelen ser la primera aproximación recomendable.

En el modelo de regresión lineal intentamos explicar la variable objetivo como una combinación lineal de los valores de los predictores:

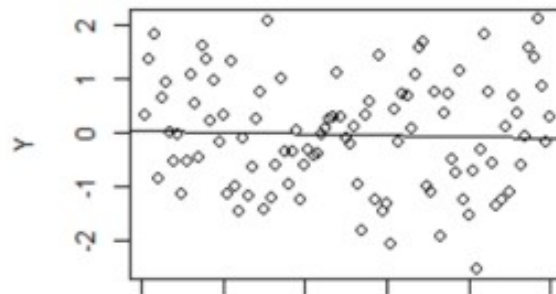
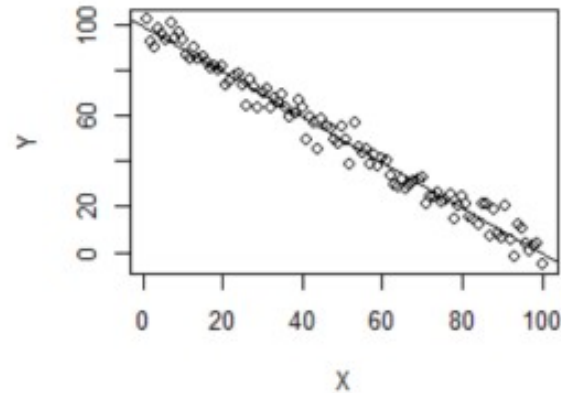
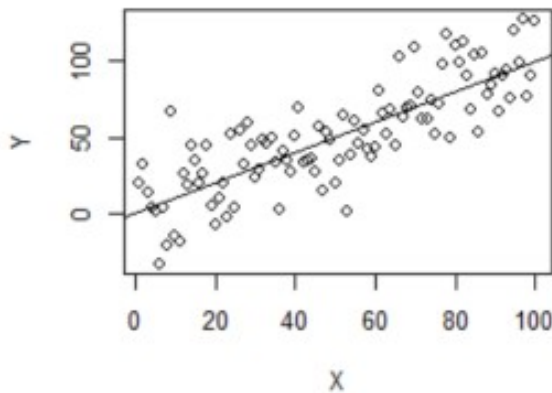
$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$$

La regresión lineal es un modelo paramétrico se entrena en el conjunto de entrenamiento aprendiendo los valores de los coeficientes *alfa* que minimizan la función de error. Los patrones de los datos quedan **grabados** en los valores de los coeficientes.

Una ventaja de la regresión lineal es que podemos dar interpretación de la influencia de los predictores a través del valor de los coeficientes. Al ser uno de los modelos más simples es de los que más interpretabilidad aporta.

Regresión lineal: correlación

$$R = \frac{\text{Cov}[X, Y]}{\sqrt{\text{Var}[X] \text{Var}[Y]}} \in [-1, 1], \quad R^2 \in [0, 1]$$



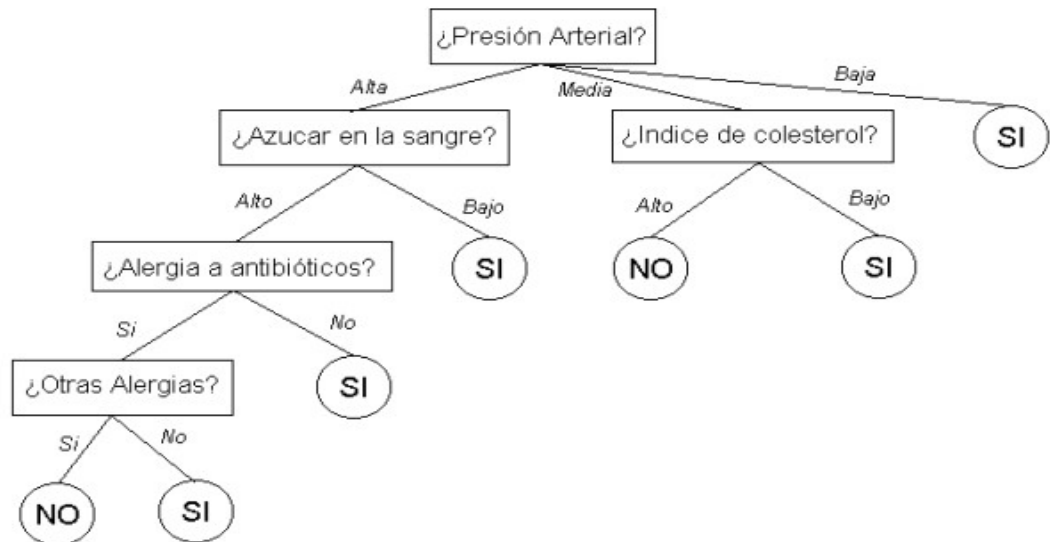
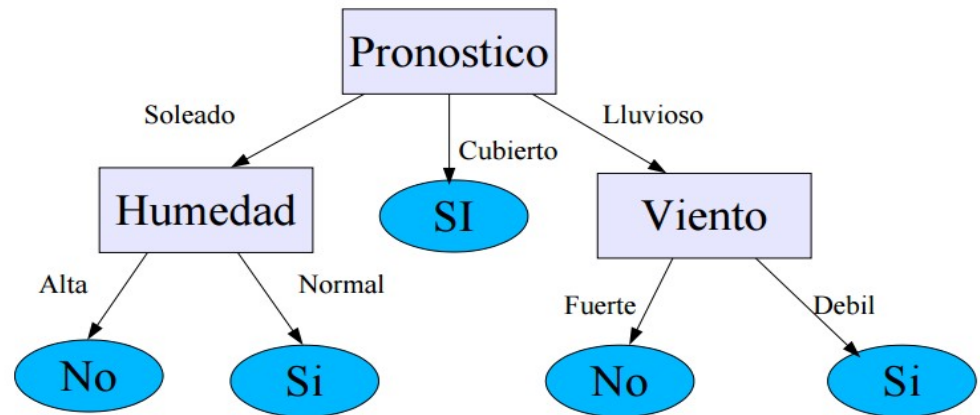
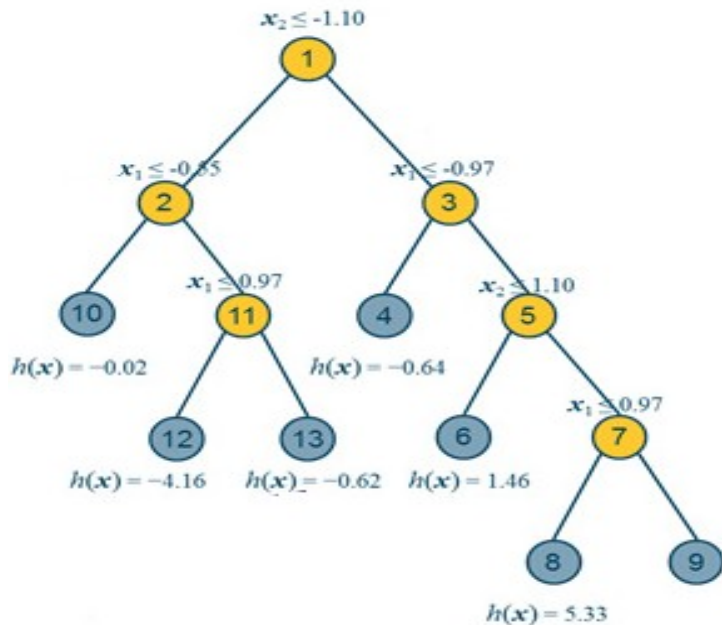
Árboles de decisión

Los árboles de decisión son modelos que generan bifurcaciones (ramas) en función de los valores de los predictores hasta llegar a una hoja final con el valor predictivo de una entrada.

Los arboles de decisión están formados por tres tipos de nodos:

- **Nodo raíz.** El nodo raíz del árbol del que cuelga el resto de nodos
- **Nodo interno.** Un nodo interno está asociado con uno de los atributos y de el salen 2 o más nodos (pueden ser intermedios o nodos hoja), cada una de estas ramas tiene asociado un criterio que ‘particionan’ o clasifica los datos en este nodo interno hacia sus nodos hijos a través de las distintas ramas (cada rama establece un criterio con respecto al atributo)
- **Nodo hoja.** Son los que nos devuelven la decisión del árbol

Árboles de decisión



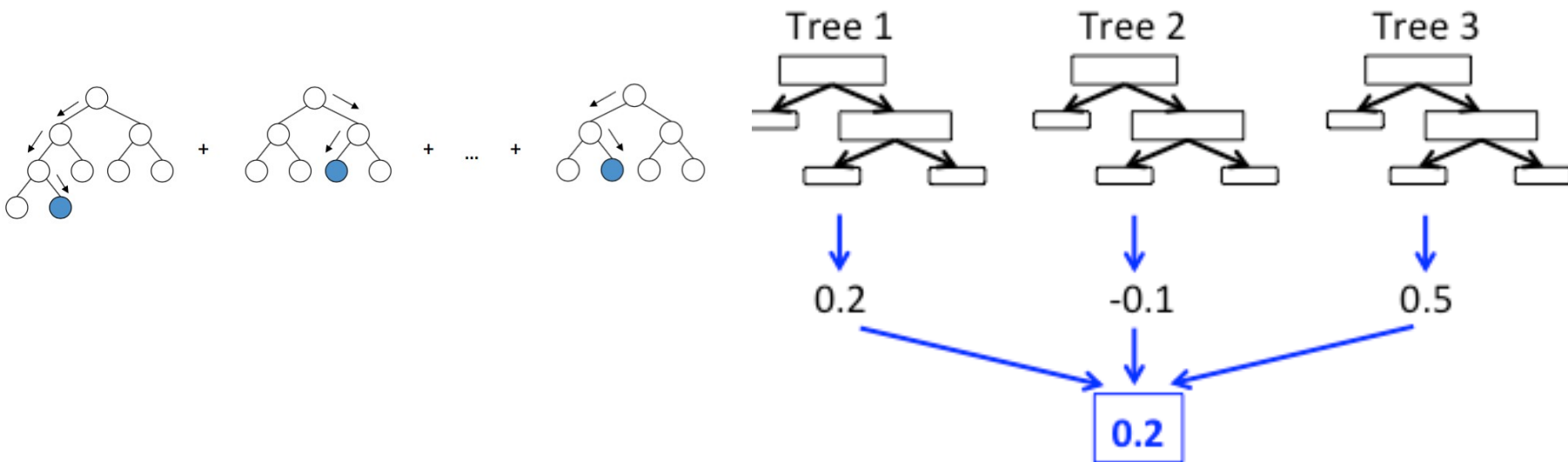
MIRAR PDF

Ensembles ó modelado conjunto

Consiste en combinar varios modelos para formar un modelo más robusto, generalmente se suelen usar árboles.

Hay dos técnicas fundamentales:

- **Bagging:** crear múltiples modelos con cierto grado de independencia y dar como predicción la media de sus salidas
- **Boosting:** crear una secuencia de modelos sencillos donde cada uno corrige los errores del anterior y dar como predicción la salida de la secuencia



K-vecinos cercanos

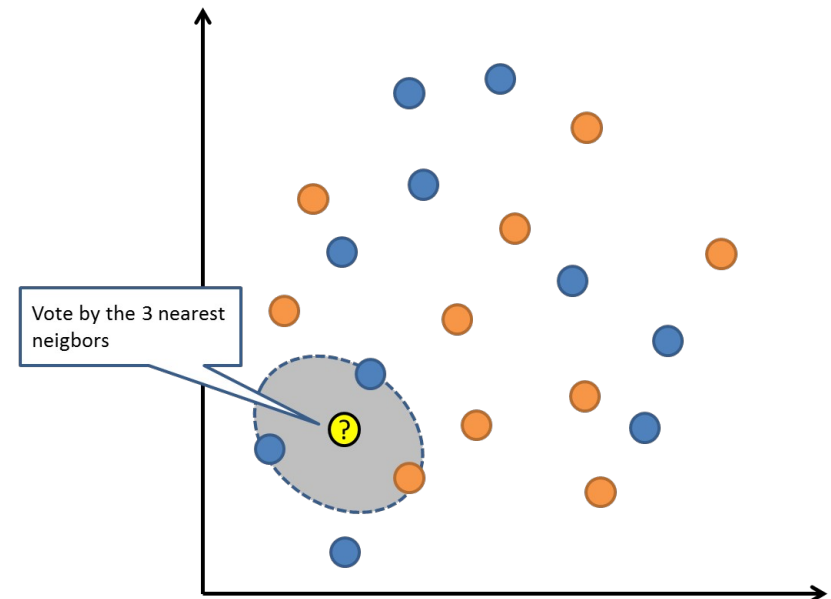
Es un familia de modelos **no paramétricos** que se basa en la proximidad para realizar la predicción.

El esquema sería:

1. Nueva entrada
2. Encontrar los k elementos más cercanos en el *dataset* de entrenamiento
3. Realizar la media, mediana ó voto por mayoría de estos elementos para dar un valor de predicción de esta entrada

Si bien el modelo es muy sencillo, incurre en dos problemas:

- **Maldición de la dimensionalidad**
- **Infraestructura para ejecutarse**



Métricas en regresión

- **MSE (*mean squared error*)**: el error cuadrático medio representa los errores de predicción elevados al cuadrado y toma su media
- **MAE (*mean absolute error*)**: el error absoluto medio representa la media de los valores absolutos de los errores de predicción
- **MAPE (*mean absolute percentage error*)**: el error medio absoluto porcentual representa la desviación absoluta proporcional respecto a la variable objetivo
- **R²**: representa la proporción de la variabilidad de la variable objetivo que queda explicada por el modelo

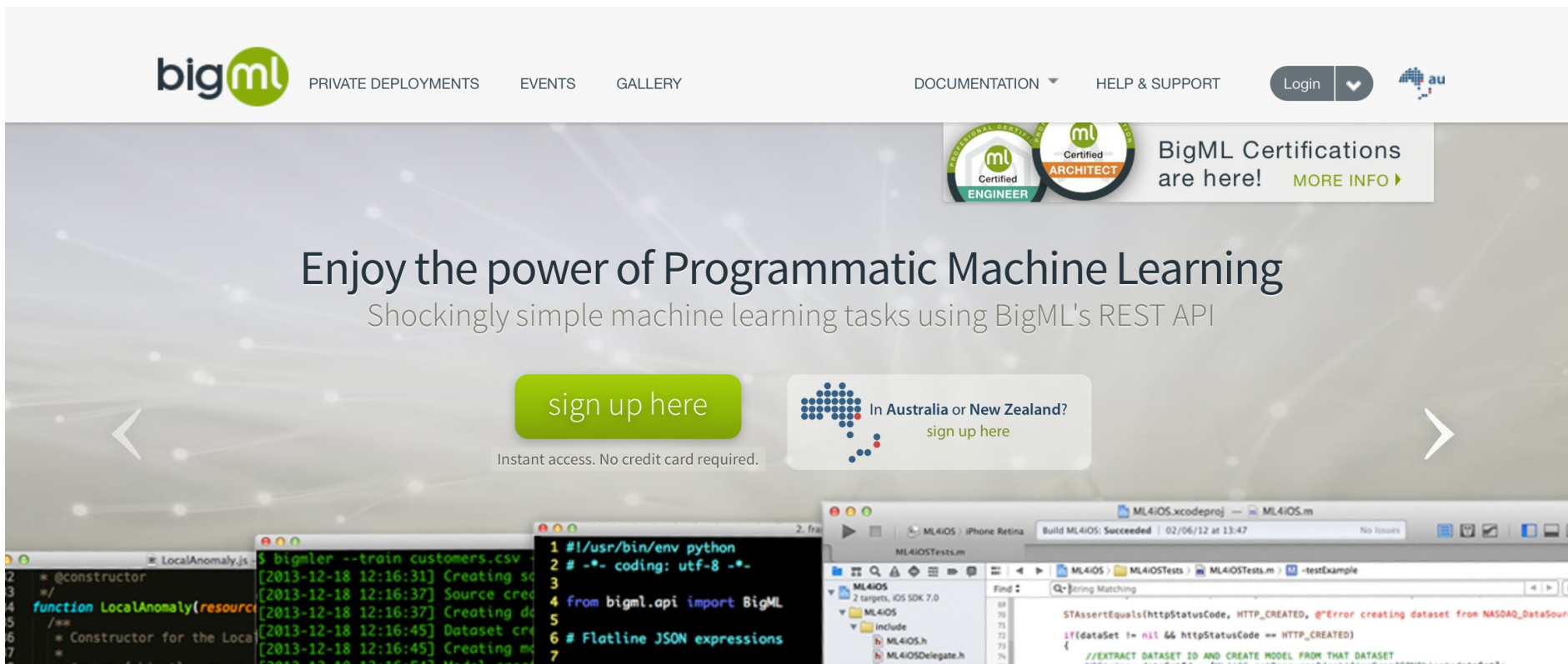
Funciones de error robustas estilo **Huber**, **Cauchy** ó **Soft-L1** sirven para minimizar el impacto de valores atípicos en *datasets* donde estos supongan un problema.

Aplicaciones de modelos de regresión

- Número de llamadas en centralita telefónica
- Precio de un producto
- Volumen de ventas
- Número de personas asistiendo a un centro comercial
- Valor de ocupación de un hotel
- RPC de una *keyword* en marketing online
- Tráfico en un tramo de carretera
- Volumen de personas requiriendo transporte público en un trayecto

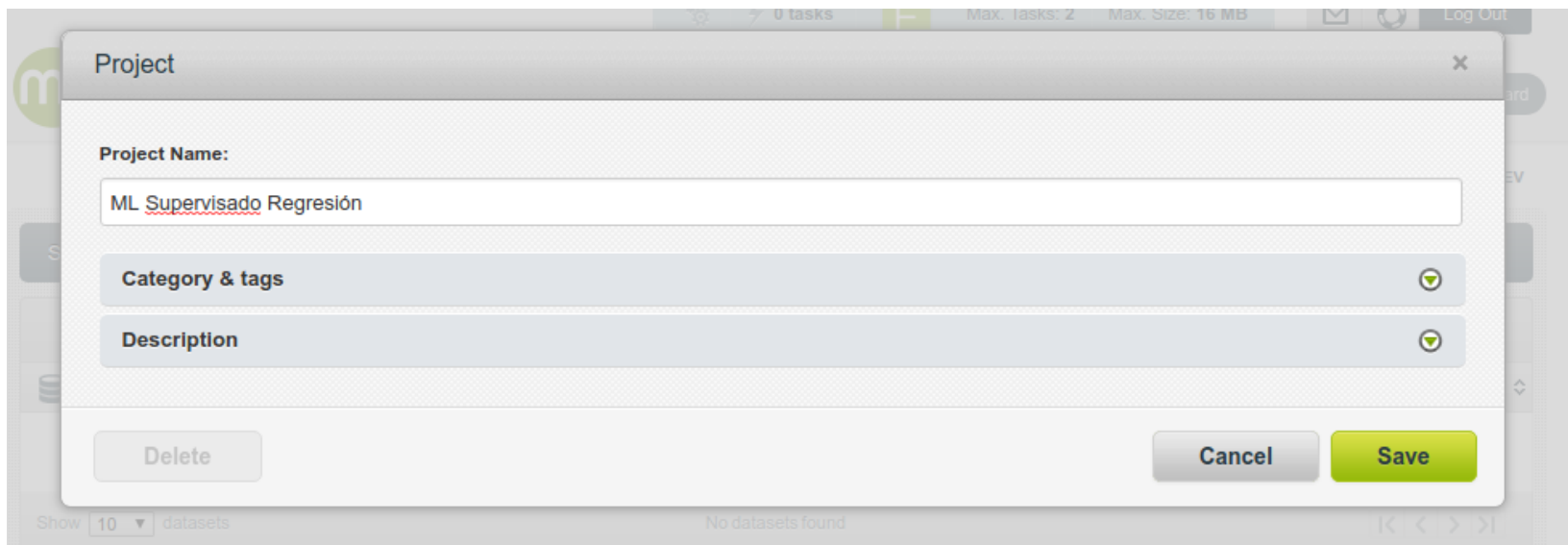


Introducción a Big ML



Introducción a Big ML: crear proyecto

Creamos proyectos para dividir los datos y los modelos que creamos y así organizar el contenido











The screenshot shows a 'Project' dialog box with the following elements:

- Project Name:** A text input field containing 'ML Supervisado Regresión'.
- Category & tags:** A dropdown menu with a downward arrow.
- Description:** A text area with a downward arrow.
- Buttons:** 'Delete' (disabled), 'Cancel', and 'Save' (highlighted in green).

The background interface shows a top bar with '0 tasks', 'Max. tasks: 2', 'Max. Size: 16 MB', and a 'Log Out' button. At the bottom, it says 'Show 10 datasets' and 'No datasets found'.

Big ML: fuentes de datos

Sources					WhizzML ▾				
Sources									
Type ▾	Name								
	Iris Flower Classification		2y 10m	4.7 KB	0				
	Country Stats Mashup		2y 10m	12.0 KB	0				
	Fictional Wine Sales		2y 10m	51.9 KB	0				
	Titanic Survival		2y 10m	78.0 KB	0				
	US Car Accidents in 2011		2y 10m	685.5 KB	0				
	Premier League 2011-2012 Season		2y 10m	24.7 KB	0				
	Churn in the Telecom Industry		2y 10m	270.4 KB	1				
	Arrhythmia Diagnosis		2y 10m	533.6 KB	0				

Big ML: crear una fuente de datos de url ó fuente local

Cargamos <https://archive.ics.uci.edu/ml/machine-learning-databases/00275/Bike-Sharing-Dataset.zip>

A continuación vamos al margen superior derecho y creamos un dataset a partir de housetrain.csv

Sources Datasets Supervised **NEW** Unsupervised Predictions Tasks WhizzML

Create a source from a URL

URL: (http://, https://, s3://, asv://, odata://, odatas://, dropbox://, gcs://, gdrive:// or hdfs://)

[ive.ics.uci.edu/ml/machine-learning-databases/communities/communities.data](https://archive.ics.uci.edu/ml/machine-learning-databases/communities/communities.data)

Examples

Name:

Unnamed remote source

Cancel Create

Type	Name	Size
CSV	US Crime	1.1 MB
CSV	Automobile ranking	25.3 KB
XLS	Iris Flower Classific	4.7 KB
CSV	Country Stats Mash	12.0 KB
TSV	Fictional Wine Sales	51.9 KB
CSV	Titanic Survival	78.0 KB
B22	US Car Accidents in 2011	685.5 KB
	Premier League 2011-2012 Season	

Big ML: crear un dataset

Sources
Datasets
Supervised ^{NEW}
Unsupervised
Predictions
Tasks
WhizzML

3

1

•

•

•

Bike Sharing

PDF

Dataset name

Size: 1.16 MB

Bike Sharing

100%

Create dataset

✓

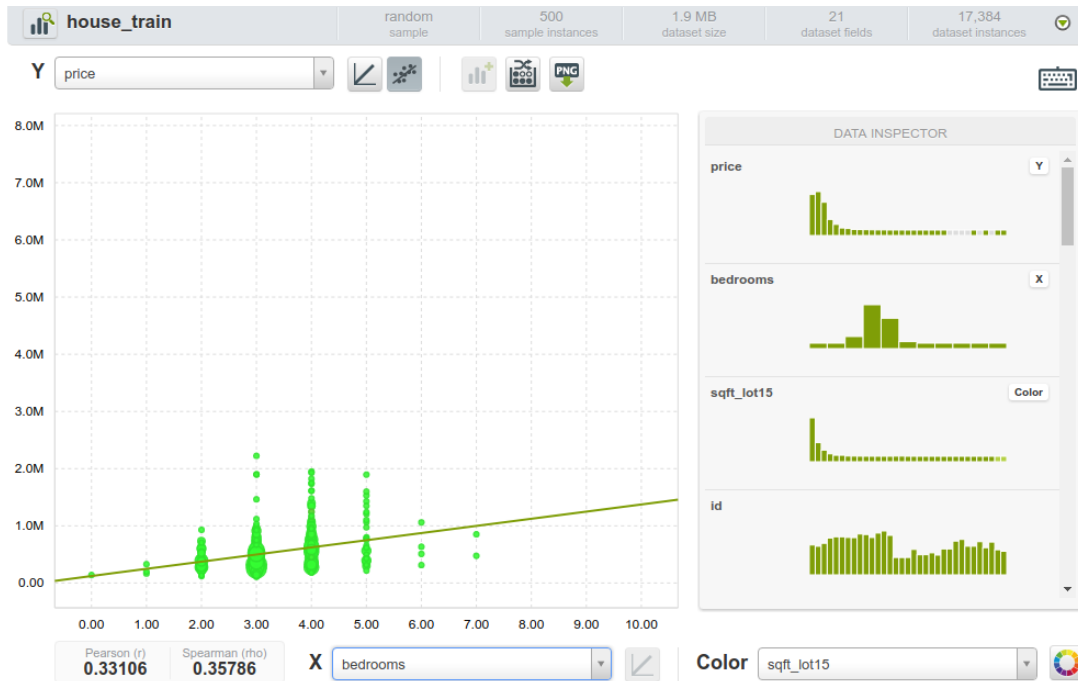
Q

×

Name	Type	Instance 1	Instance 2	Instance 3
instant	1 2 3	1	2	3
dteday	YYYY-MM-DD	2011-01-01	2011-01-01	2011-01-01
season	1 2 3	1	1	1
yr	A B C	0	0	0
mnth	1 2 3	1	1	1

Big ML: dynamic scatterplot del dataset

A través del gráfico de dispersión podemos ver las relaciones de los predictores con la variable objetivo y sus correlaciones. Además podemos observar las distribuciones de cada predictor y ayuda a obtener una primera intuición de los datos.



Encuentra variables respecto a las que tenga alto grado de explicabilidad.

Big ML: dynamic scatterplot del dataset

Realiza los gráficos de SalePrice respecto a:

- Centralair
- LotArea
- LotFrontage
- TotalBsmntSF

Las explicaciones de las variables están en el archivo .txt descriptor

- ¿Es importante el año de construcción?
- ¿Es importante el año de remodelación?
- ¿Qué tal es como predictor el número de coches que caben en el garaje?
- ¿Influye tener piscina?
- ¿Cuál es el mayor coeficiente de correlación que hayamos?

Big ML: partimos el dataset en entrenamiento y test

Es importante poner en **Seed** el valor 0 para que los resultados sean **reproducibles**

SPLIT DATASET CONFIGURATION

Training: 80% | Test: 20%

Seed:

Linear split: ?

Training dataset name: Houses train | Training (80%)

Test dataset name: Houses train | Test (20%)

Reset | Create Training | Test

Name	Type	Count	Missing	Errors	Histogram
Id	123	1,460	0	0	
MSSubClass	123	1,460	0	0	
MSZoning	ABC	1,460	0	0	
LotFrontage	123	1,201	259	0	
LotArea	123	1,460	0	0	
Street	ABC	1,460	0	0	

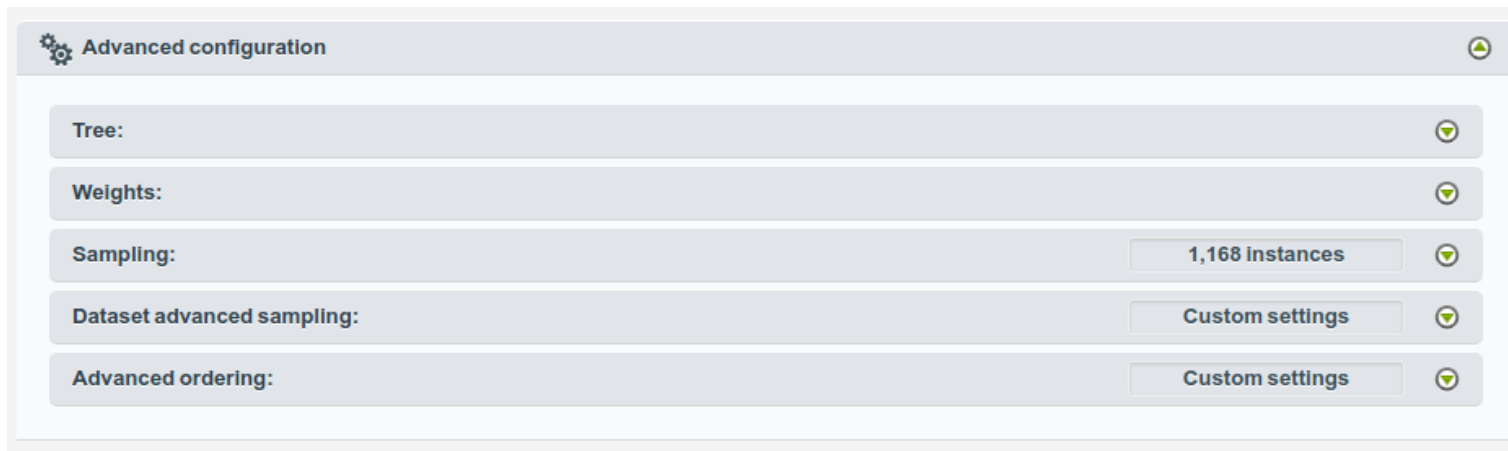
Big ML: aplicamos modelos en el conjunto de entrenamiento

BigML sólo soporta modelos de árboles en el ámbito supervisado, de modo que aplicaremos este tipo.

Cargamos el configurador de modelo sobre el conjunto de entrenamiento creado.

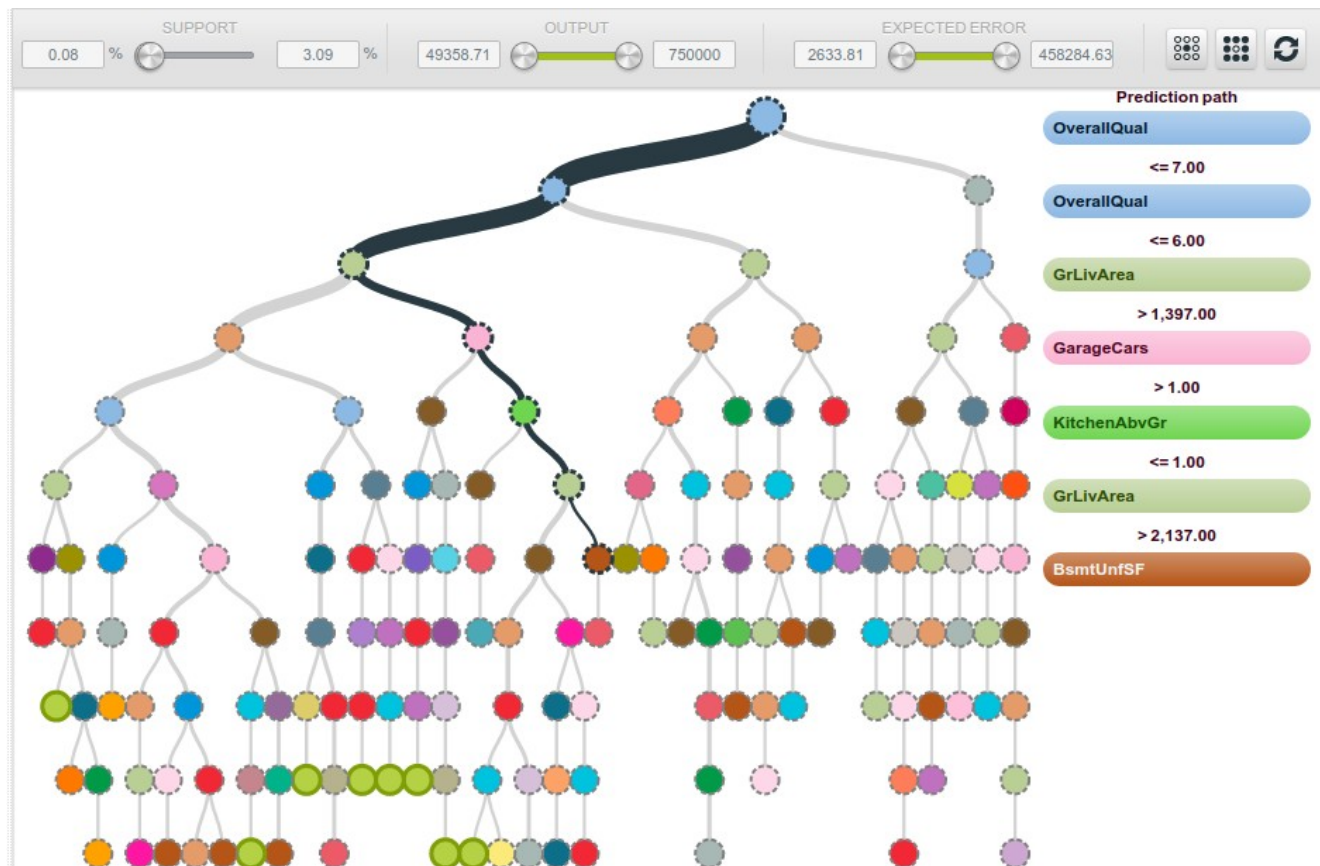


- La primera opción es
- Las opciones avanzadas son



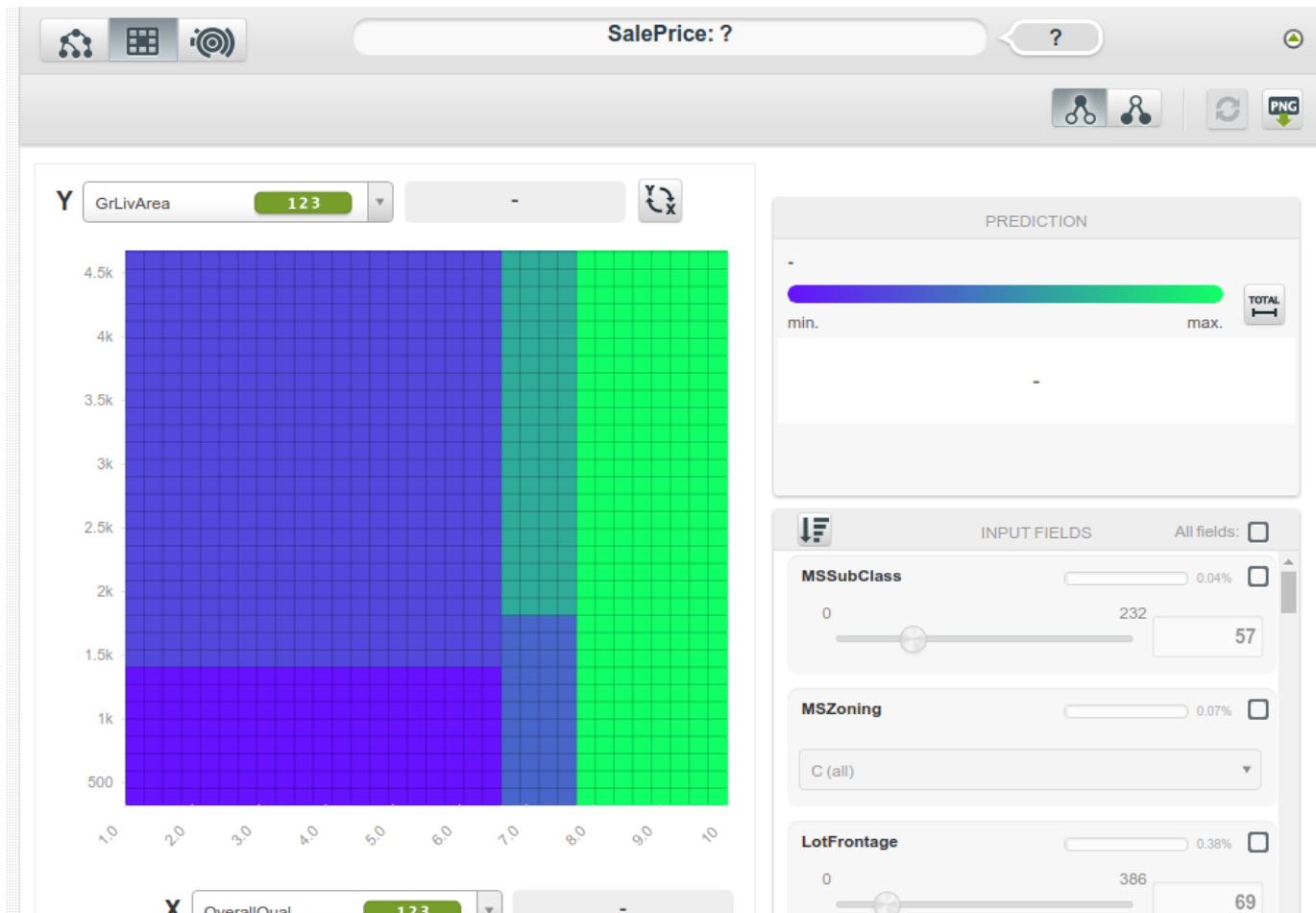
Big ML: la salida del modelo creado - árbol

Son muy interesantes los dos filtros de soporte de patrones generales y patrones extraños.

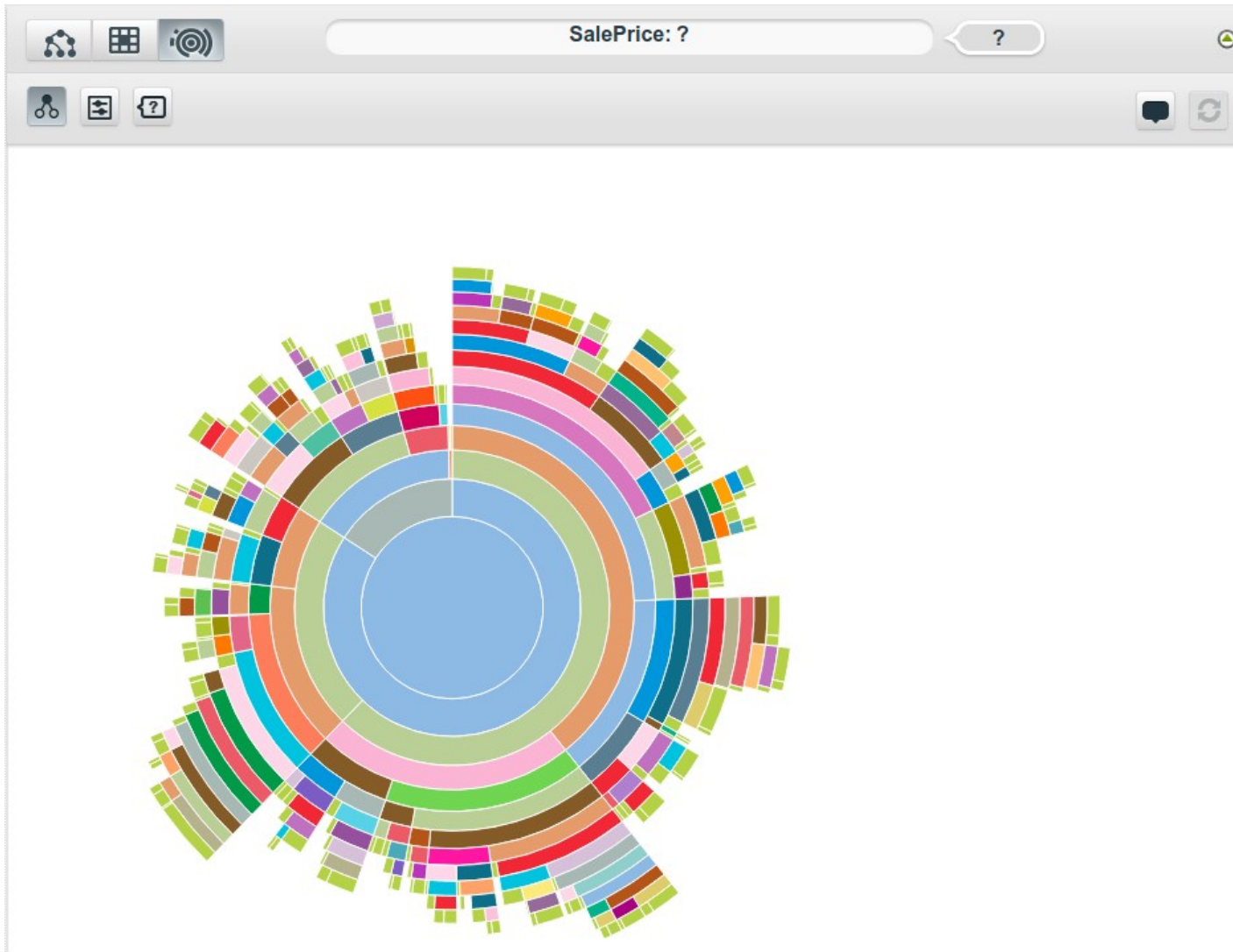


Big ML: la salida del modelo creado - PDP

La salida PDP permite observar los valores de predicción para cruces de dos variables, aportando interpretabilidad



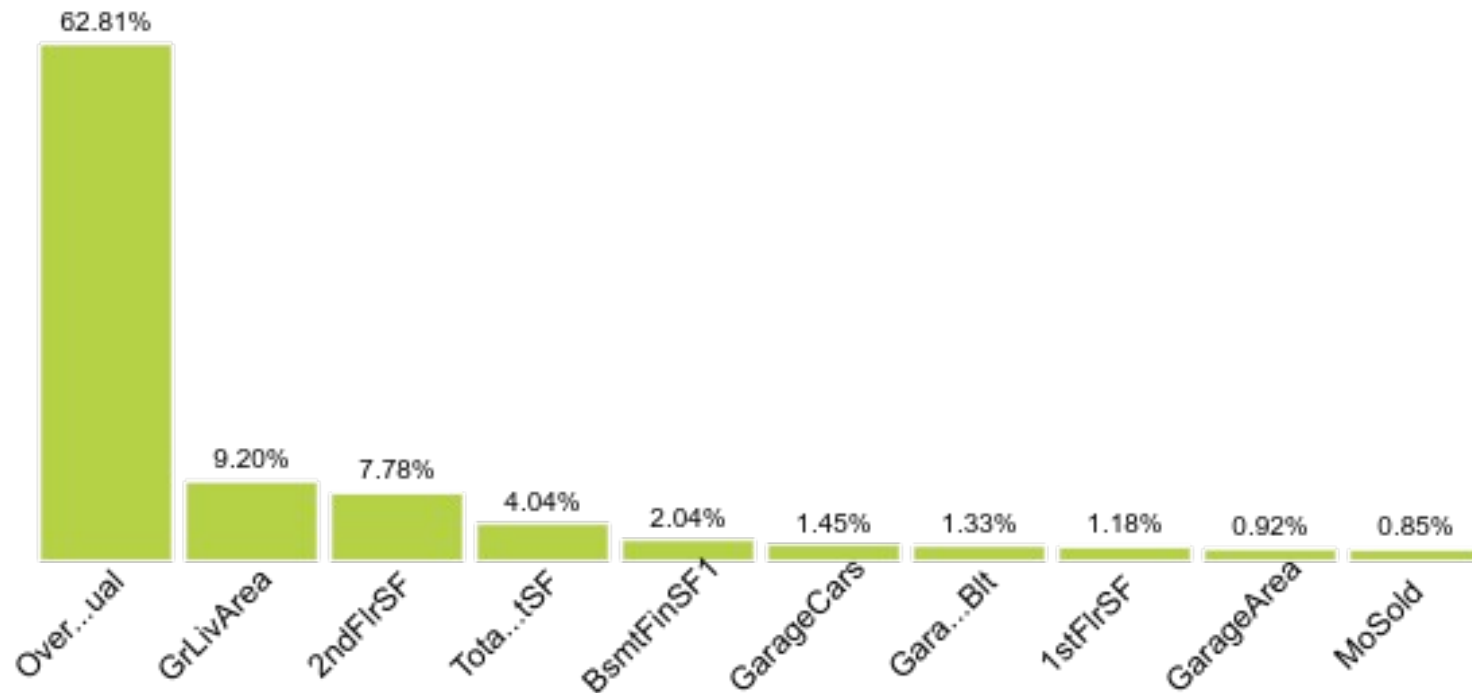
Big ML: la salida del modelo creado - Sunburst



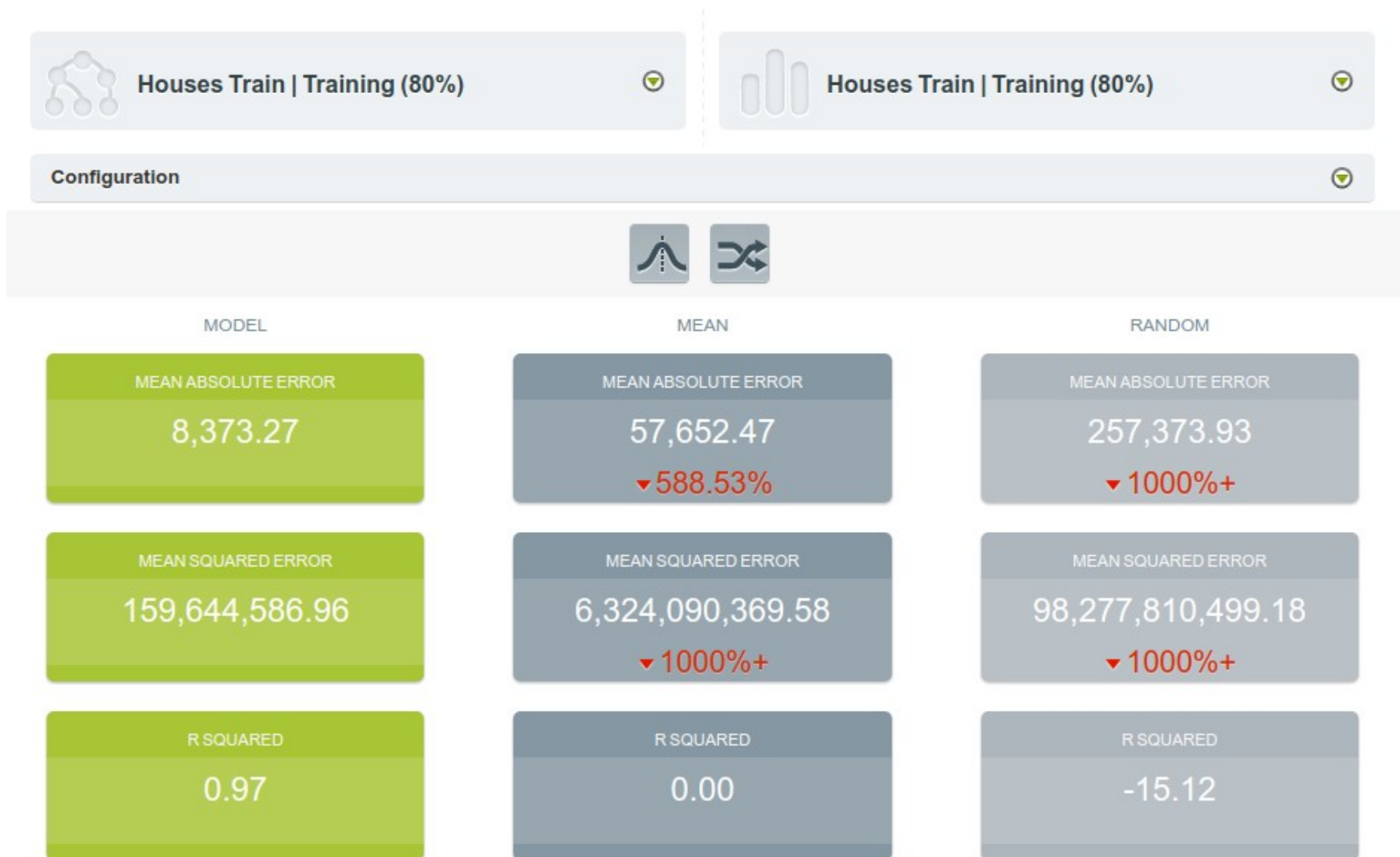
Big ML: la salida del modelo creado

En **model summary** podemos ver la importancia de atributos

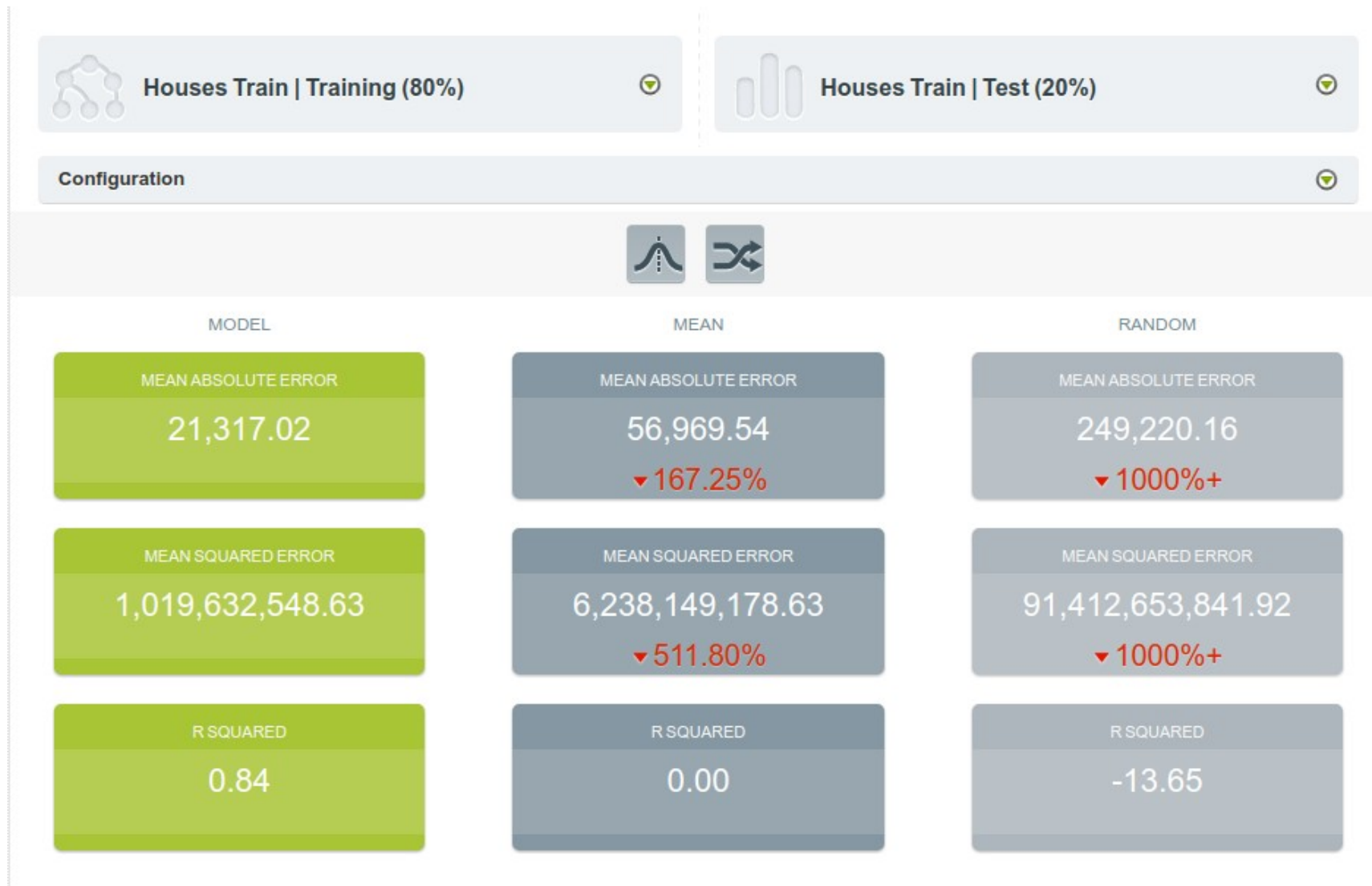
Houses train | Training (80%) Field Importances



Big ML: evaluamos el modelo en el conjunto de entrenamiento

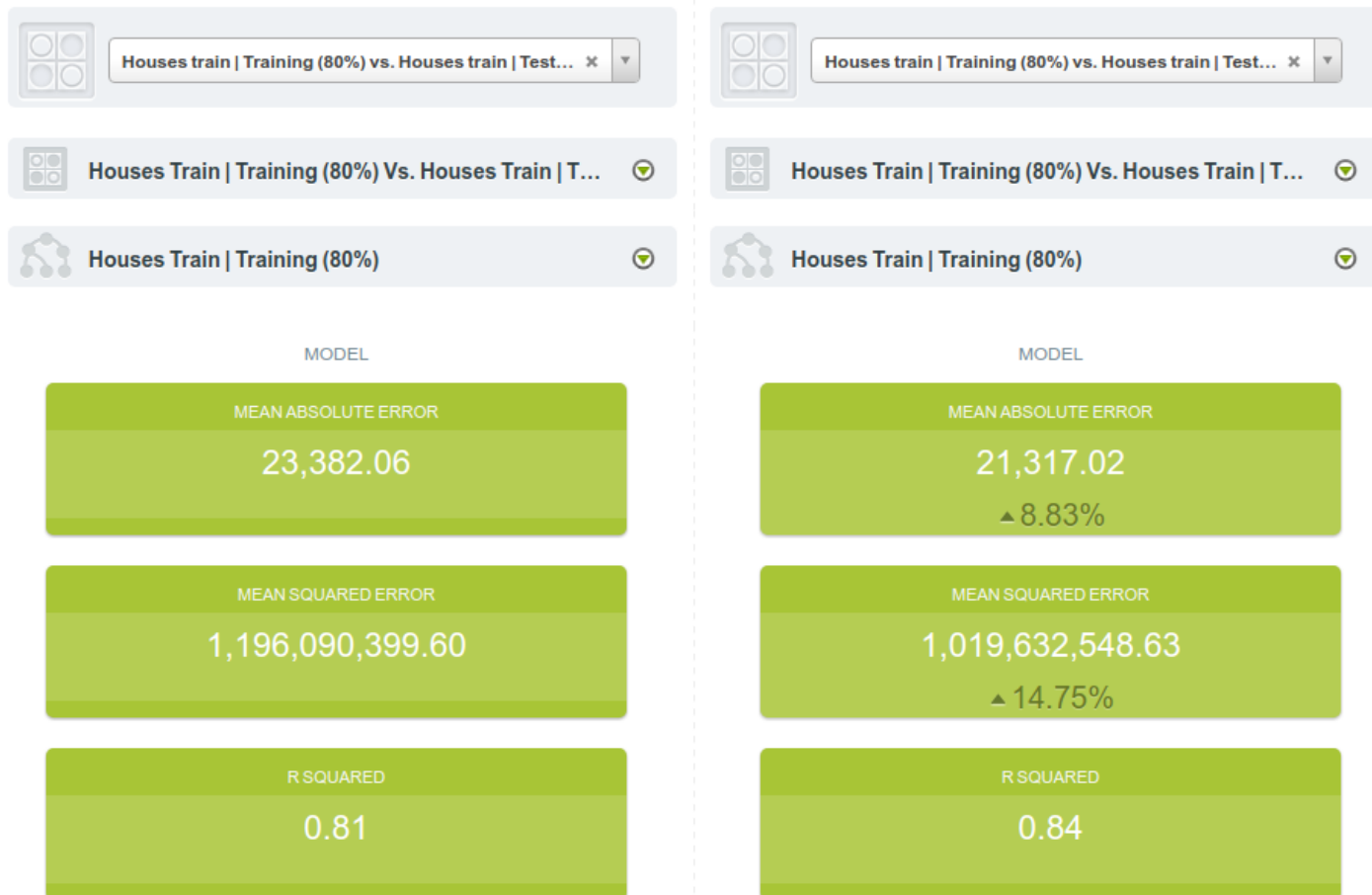


Big ML: evaluamos el modelo en el conjunto de test



Big ML: comparación de evaluaciones de modelos

Creamos un modelo sin *statistical pruning* pero incluyendo *missing splits* y con un valor de 1000 en el *node threshold* y evaluamos las capacidades en test en comparación con el automático



Big ML: realizar una predicción con el mejor

Usamos la opción **creat batch prediction** y realizamos una predicción sobre el conjunto test. Podremos descargar un .csv con los valores

Houses train | Training (80%)

SalePrice Wed, 20 Sep 2017 16:14:14

360.0 KB size	52 fields	1,168 instances
------------------	--------------	--------------------

Description:

Houses train | Test (20%)

SalePrice Wed, 20 Sep 2017 16:05:29

89.9 KB size	81 fields	292 instances
-----------------	--------------	------------------

Description:

Configure

Preview of the prediction file (using the type of each field)

```
Id,MSSubClass,MSZoning,LotFrontage,LotArea,Street,Alley,LotShape,LandContour,Utilities,LotConfig,LandSlope,Neighborhood,Condition1,Condition2,BldgType,HouseStyle,OverallQual,OverallCond,YearBuilt,YearRemodAdd,RoofStyle,RoofMatl,Exterior1st,Exterior2nd,MasVnrType,MasVnrArea,ExterQual,ExterCond,Foundation,BsmtQual,BsmtCond,BsmtExposure,BsmtFinTypel,BsmtFinSF1,BsmtFinType2,BsmtFinSF2,BsmtUnfSF,TotalBsmtSF,Heating,HeatingQC,CentralAir,Electrical,1stFlrSF,2ndFlrSF,LowQualFinSF,GrLivArea,BsmtFullBath,BsmtHalfBath,FullBath,HalfBath,BedroomAbvGr,KitchenAbvGr,KitchenQual,TotRmsAbvGrd,Functional,Fireplaces,FireplaceQu,GarageType,GarageYrBlit,GarageFinish,GarageCars,GarageArea,GarageQual,GarageCond,PavedDrive,WoodDeckSF,OpenPorchSF,EnclosedPorch,3SsnPorch,ScreenPorch,PoolArea,PoolQC,Fence,MiscFeature,MiscVal,MoSold,YrSold,SaleType,SaleCondition,SalePrice,SalePrice
123,123,ABC,123,123,ABC,ABC,ABC,ABC,ABC,ABC,ABC,ABC,ABC,ABC,123,123,123,123,ABC,ABC,ABC,ABC,ABC,123,ABC,ABC,ABC,ABC,ABC,ABC,ABC,123,ABC,123,123,123,ABC,ABC,ABC,ABC,123,123,123,123,123,123,123,123,123,123,ABC,123,ABC,123,ABC,AB
```

Prediction name:

Houses train | Test (20%) with Houses train | Training (80%) MOI

Reset **Predict**

Big ML: montamos un modelo ensemble tipo RandomForest

ENSEMBLE CONFIGURATION

Objective field:

SalePrice

1 2 3

Type:

Decision Forest

Number of models:

202

Number of iterations: ?

64



Advanced configuration

Tree:

Missing splits:



MISSING SPLITS ?

YES

Node threshold:

512

NODE THRESHOLD ?

512

Randomize:



Random candidates:

Ratio of fields

40%

RANDOMIZE ?

40%

Big ML: montamos un modelo ensemble tipo RandomForest

ENSEMBLE CONFIGURATION

Objective field:

SalePrice

1 2 3

Type:

Decision Forest

Number of models:

202

Number of iterations: ?

64



Advanced configuration

Tree:

Missing splits:



MISSING SPLITS ?

YES

Node threshold:

512

NODE THRESHOLD ?

512

Randomize:



Random candidates:

Ratio of fields

40%

RANDOMIZE ?

40%

Big ML: resultados de RandomForest

1



model/59c2a798400683200501860f

Data distribution:



Predicted distribution:



2



model/59c2a7984006832005018611

Data distribution:



Predicted distribution:



3



model/59c2a7994006832005018613

Data distribution:



Predicted distribution:



4



model/59c2a7994006832005018615

Data distribution:

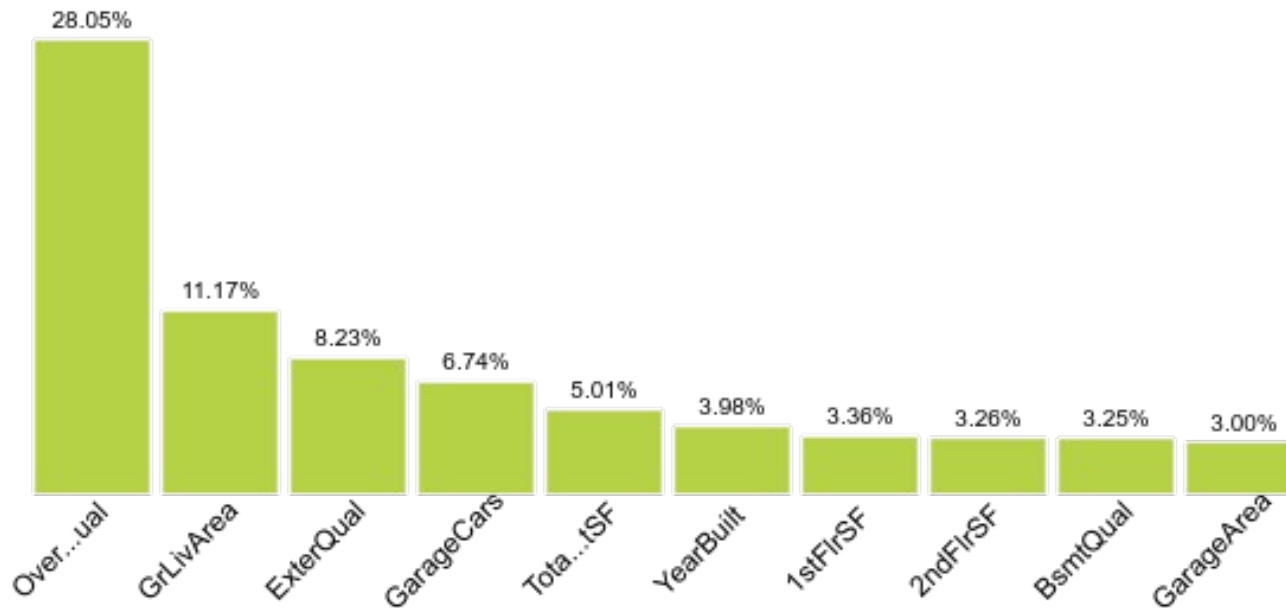


Predicted distribution:



Big ML: importancia de atributos establecida por RandomForest

Houses train | T...0%) RandomForest Field Importances





Big ML: evaluamos los resultados de RandomForest

Es especialmente interesante la opción de configuración que establece la ponderación de los modelos combinados



Podemos observar que además, los resultados son mejores en test que los otros modelos entrenados cuyo valor mínimo de MAE era 21

 		
ENSEMBLE	MEAN	RANDOM
MEAN ABSOLUTE ERROR 16,391.73	MEAN ABSOLUTE ERROR 56,969.54 ▼247.55%	MEAN ABSOLUTE ERROR 267,969.25 ▼1000%+
MEAN SQUARED ERROR 913,905,020.54	MEAN SQUARED ERROR 6,238,149,178.63 ▼582.58%	MEAN SQUARED ERROR 103,135,600,272.65 ▼1000%+
R SQUARED 0.85	R SQUARED 0.00	R SQUARED -15.56

Big ML: montamos un modelo Gradient Boosting

ENSEMBLE CONFIGURATION

Objective field:
SalePrice

123


Type:
Boosted Trees

Number of models:
10


Number of iterations:
312

Advanced configuration

Tree:

Missing splits:


Node threshold:
512

Randomize:


Random candidates:
Ratio of fields

49%

MISSING SPLITS
NO

NODE THRESHOLD
512

RANDOMIZE
49%

Boosting:

Early stopping:
Early out of bag

Learning Rate (LR):
5%

EARLY STOPPING
Early out of bag

LEARNING RATE
5.00%

Big ML: evaluamos el modelo Gradient Boosting

