H2O Flow

H2O Flow es una interfaz de usuario de código abierto para H2O. Es un entorno interactivo basado en web que le permite combinar ejecución de código, texto, matemáticas y gráficos en un solo documento. H2O Flow permite importar archivos, crear modelos, mejorarlos y hacer predicciones sin escribir ninguna línea de código.

La interfaz de usuario híbrida de Flow combina la informática de línea de comandos con una interfaz gráfica. H2O Flow Envía comandos a H2O como una secuencia de celdas ejecutables. Cuando se ejecuta la celda, la salida es un objeto gráfico.

Crearemos modelos en H2O Flow usando Auto ML (aprendizaje automático automatizado), que permite automatizar el proceso de modelado.

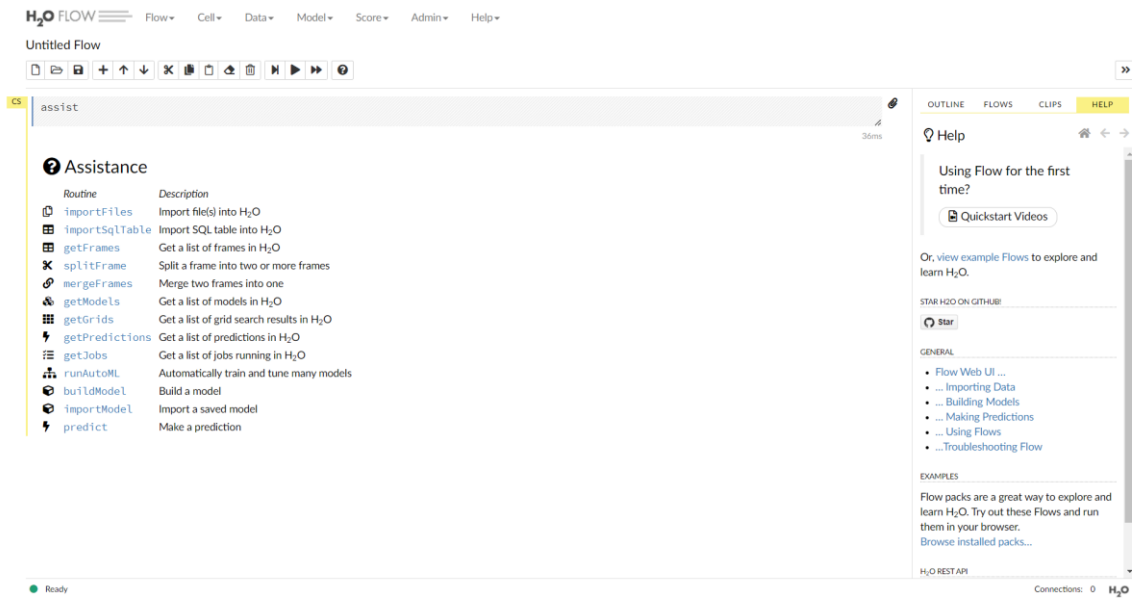
1. **Instalación de h2o.**

Descargar la ultima versión de H2O de la página oficial, lo que desacargará un zip en la carpeta de descargas. Tras descomprimierlo, hay que abrirlo en el terminal para escribir este comando, y así poder instalar H2O.

```
PS C:\Users\albah\Downloads\h2o-3.38.0.2\h2o-3.38.0.2> java -jar h2o.jar
```

Luego navegar a http://localhost:54321 para obtener acceso a la interfaz de usuario web de Flow.

La interfaz de Flow es muy parecida a la de Jupiter Notebook



2. **Cargar los datos.**

Para este proyecto se empleará un conjunto de datos disponibles gratuitamente, que tratan de campañas de marketing directo de un banco y se necesita predecir la inscripción de sus clientes. Es un problema de clasificación binaria ya que tenemos que predecir si los clientes se inscribirán en depósitos a plazo o no.

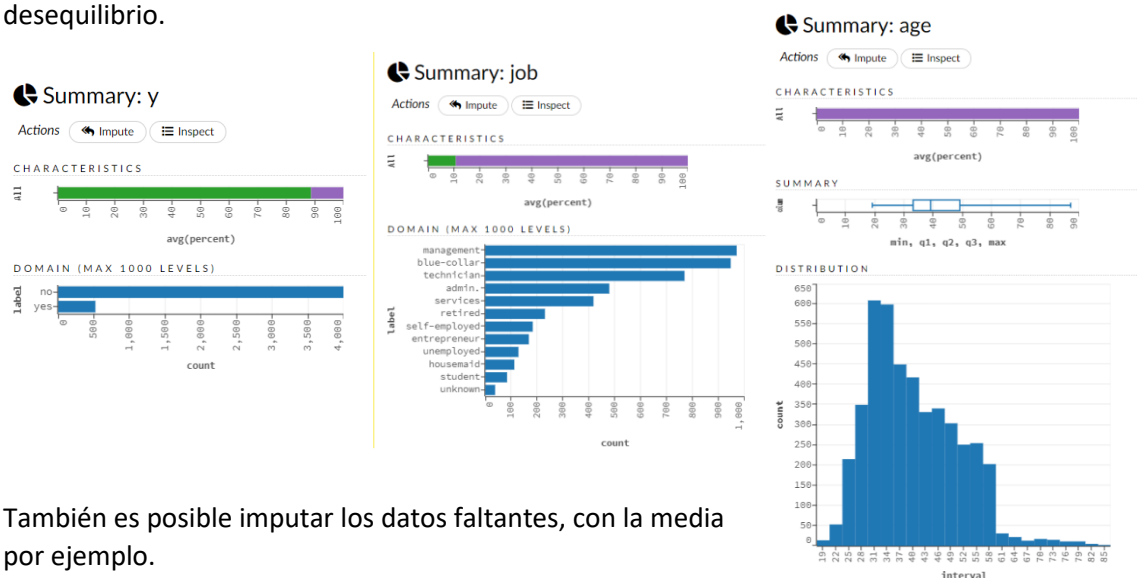
https://www.analyticsvidhya.com/blog/2021/05/a-step-by-step-guide-to-automl-with-h2o-flow/

En 'Upload File' indico la ruta del archivo csv y lo cargo. Luego, el esquema se detecta automáticamente mediante el parse guesser, aunque podemos modificar el tipo de dato si lo deseamos.

3. Análisis de los datos.

Se puede visualizar los datos para una mejor comprensión. Al seleccionar las columnas por separado, podemos obtener su distribución si es numérica o su frecuencia si es categórica.

Se puede apreciar que en la columna y, que es la variable para predecir, existe un alto nivel de desequilibrio.

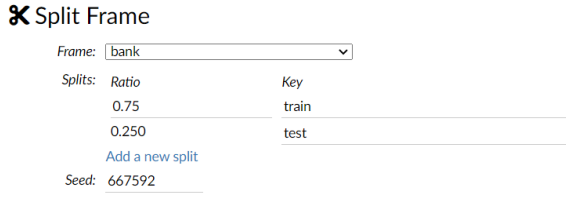


También es posible imputar los datos faltantes, con la media por ejemplo.

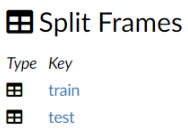
4. Separación en train y test.

Antes de entrenar al modelo, es necesario dividir los datos en el conjunto de entrenamiento y prueba. La división predeterminada es 75:25 respectivamente, aunque lo modificaremos a 80:20.

```
splitFrame
```




```
splitFrame "bank", [0.75], ["train","test"], 667592
```



5. Creación del modelo de Machine Learning con AutoML.

AutoML entrena varios tipos de modelos, incluidos GLM's random forests, distributed random forests (DRF), Gradient Boosted Machine (GBM), Deep Dearning, XGBoost, y StackedEnsembles. También presenta una tabla de clasificación con todos los modelos ordenados por algunas métricas.

Se selecciona Run AutoML, se elige el conjunto de entrenamiento, la variable para predecir, y el conjunto de validación, y se construyen los modelos

```
runAutoML {"input_spec":{"training_frame":"train","response_column":"y","validation_frame":"test","ignored_columns":
[],"sort_metric":"AUTO"},"build_control":{"distribution":"AUTO","nfolds":-1,"balance_classes":false,"stopping_criteria":
{"seed":-1,"max_models":0,"max_runtime_secs":300,"max_runtime_secs_per_model":0,"stopping_rounds":3,"stopping_metric":"AUTO","stop
ping_tolerance":-1},"keep_cross_validation_predictions":true,"keep_cross_validation_models":true,"keep_cross_validation_fold_assig
nment":false},"build_models":{"exclude_algos":[],"exploitation_ratio":-1,"monotone_constraints":[]}}, 'exec'
```


5.1m

Job

Run Time	00:05:02.559
Remaining Time	00:00:00.0
Type	AutoML
Key	AutoML_1_20221103_113410@@y
Description	AutoML build
Status	DONE
Progress	100%

Done.

El proceso de entrenamiento ha durado 5 minutos, como hemos especificado anteriormente 'mas_runtime_second = 300'. En esta tabla se muestran todoslos modelos entrenados por AutoML ordenados según el rendimiento.

▾ MODELS

models sorted in order of auc, best first

	model_id	auc	logloss	aucpr	mean_per_class_error	rmse
0	StackedEnsemble_AllModels_3_AutoML_1_20221103_113410	0.9477687592827355	0.1753711623255821	0.6562510892040977	0.16003051780162858	0.23707786804
1	GBM_grid_1_AutoML_1_20221103_113410_model_2	0.9471248465504283	0.17767785904566558	0.6536446197778906	0.15765278065512223	0.23818246974
2	GBM_grid_1_AutoML_1_20221103_113410_model_17	0.947070050297037	0.1781408641717221	0.6489071367441888	0.14749131464337084	0.23844007742
3	StackedEnsemble_BestOfFamily_4_AutoML_1_20221103_113410	0.9470226876234485	0.17619429552663787	0.6544111407740734	0.15772390649823695	0.23770276744
4	GBM_grid_1_AutoML_1_20221103_113410_model_22	0.946981834611719	0.17804316489764718	0.6509744552415474	0.15324269328732962	0.23783680265
5	StackedEnsemble_AllModels_2_AutoML_1_20221103_113410	0.9469106612528243	0.1763188419764525	0.6513436395152684	0.16719255020147747	0.23780222374
6	GBM_grid_1_AutoML_1_20221103_113410_model_10	0.9468471748912258	0.1783896325827972	0.6521912168754753	0.1520375083231808	0.23785850329
7	GBM_grid_1_AutoML_1_20221103_113410_model_27	0.94653614715447	0.17913353468790708	0.648552959480838	0.1650729715671908	0.23874333469
8	StackedEnsemble_BestOfFamily_3_AutoML_1_20221103_113410	0.9465043432590631	0.17694199769424793	0.6494588593566676	0.16006519904149058	0.23814481039
9	StackedEnsemble_AllModels_1_AutoML_1_20221103_113410	0.9464893546702415	0.177254695836689	0.6483756378182292	0.15436050729113807	0.23842534236
10	StackedEnsemble_BestOfFamily_2_AutoML_1_20221103_113410	0.9460743254501487	0.177473049443908	0.6473300574684195	0.15692118019060797	0.23864596054

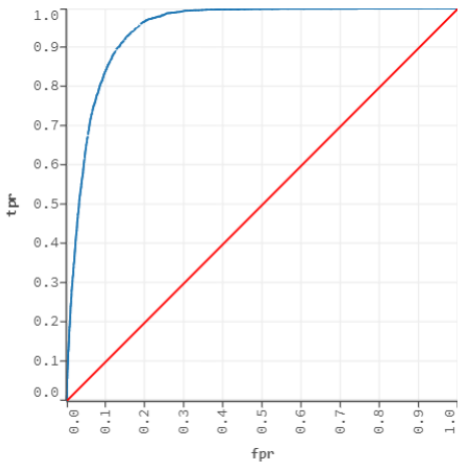
En este caso, el modelo StackedEnsemble es el mejor y GBM el segundo. El método Stacked Ensemble es una combinación óptima de una colección de algoritmos. El método Gradient Boosted Machine construye secuencialmente árboles de regresión.

6. Exploración del modelo.

AutoML permite ver gráficos de cada modelo y así analizar su precisión.

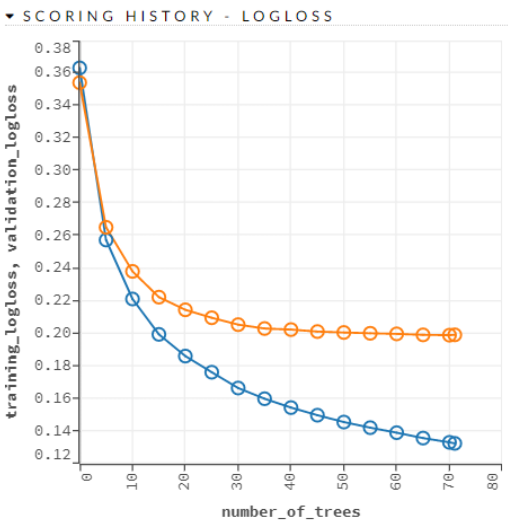
<u>Stacked Ensemble</u>

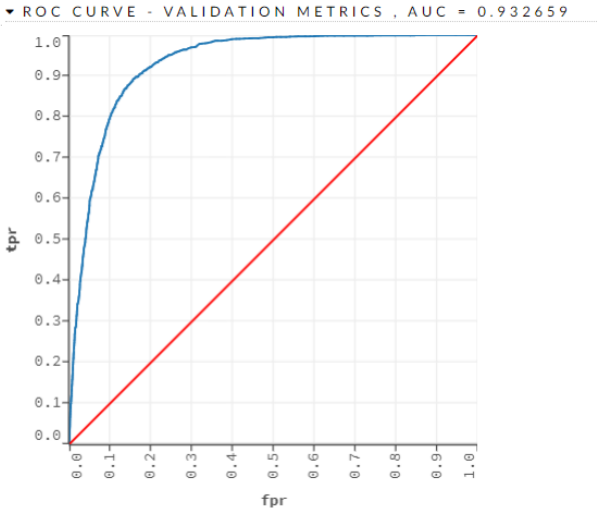
▾ ROC CURVE - CROSS VALIDATION METRICS , AUC = 0.947769

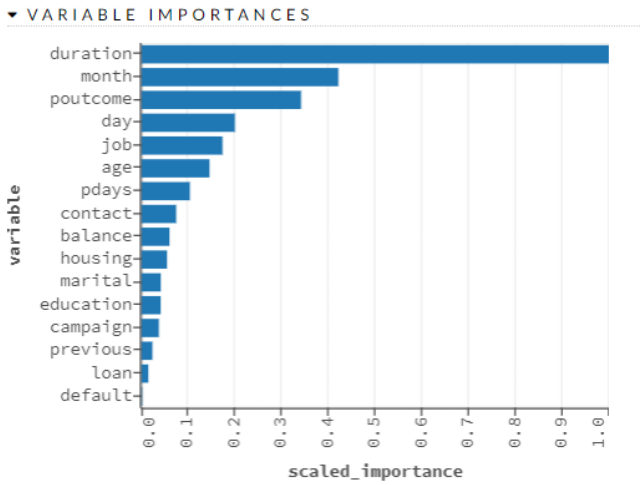


La curva ROC de la cross validation resume el rendimiento del modelo, evaluando los verdaderos positivos y los falsos positivos. Cuanto mayor sea el área bajo la curva, mejor será la predicción.

GBM







VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	no	yes	Error	Rate	Precision
no	9185	903	0.0895	903 / 10.088	0.97
yes	298	995	0.2305	298 / 1.293	0.52
Total	9483	1898	0.1055	1.201 / 11.381	
Recall	0.91	0.77			

Con el gráfico de pérdidas, podemos ver que el modelo tiene un rendimiento parecido en el conjunto de entrenamiento y el de validación.

La matriz de confusión compara los valores de la predicción con la realidad (TP, TN, FP, FN).

Las variables que más importancia tiene en este método es la duración y el mes.

7. Predicción.

Una vez seleccionado el modelo de ML, se realizarán las predicciones.

```
predict model: "StackedEnsemble_AllModels_3_AutoML_1_20221103_201900"
```

Predict

Name: prediction

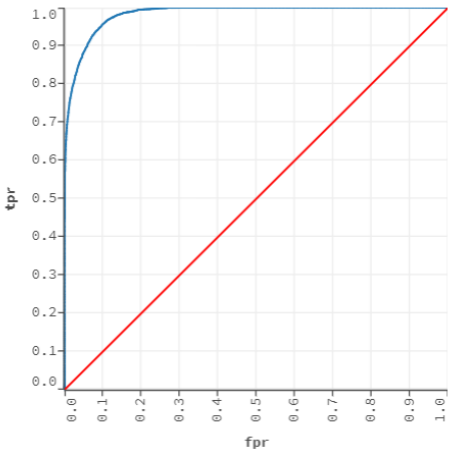
Model: StackedEnsemble_AllModels_3_AutoML_1_20221103_201900

Frame: train

Actions: Predict

La curva ROC y la matriz de confusión de la predicción.

ROC CURVE



	no	yes	Error	Rate	Precision
no	26890	412	0.0151	412 / 27.302	0.97
yes	821	2658	0.2360	821 / 3.479	0.87
Total	27711	3070	0.0401	1.233 / 30.781	
Recall	0.98	0.76			

Las características de la predicción.

PREDICTION

model	StackedEnsemble_AllModels_3_AutoML_1_20221103_201900
model_checksum	8059304726940656324
frame	train
frame_checksum	6717578143759451204
description	·
model_category	Binomial
scoring_time	1667503661782
predictions	prediction
MSE	0.057136
RMSE	0.239032
nobs	30781
custom_metric_name	·
custom_metric_value	0
r2	0.430060
logloss	0.192255
AUC	0.983505
pr_auc	0.909038
Gini	0.967011
mean_per_class_error	0.125539
residual_deviance	11835.583999
null_deviance	21718.578447
AIC	11853.583999
null_degrees_of_freedom	30780
residual_degrees_of_freedom	30772