UNIVERSIDAD DE ANTIOQUIA

PREDICCION DE LA DEMANDA USANDO MODELOS DE MACHINE LEARNING

Autor

Edwar Andrés Hincapié Herrera

Universidad de Antioquia

Facultad Ingeniería, Ingeniería de Sistemas

Medellín, Colombia

2021

Predicción de la Demanda Usando Modelos de Machine Learning

Edwar Andrés Hincapié Herrera

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Especialista en Analítica y Ciencia de Datos

Asesores (a):
Javier Fernando Botia Valderrama

Línea de Investigación:
Ciencia de Datos

Universidad de Antioquia
Facultad Ingeniería, Ingeniería de Sistemas
Medellín, Colombia
2021

Resumen

Para aquellas empresas dedicadas a la venta en *retail* o venta directa donde su portafolio de productos es muy amplio, la planeación de la demanda se convierte en un área determinante para la correcta administración del flujo de caja, rentabilidad y efectividad en ventas por varias razones: la primera de ellas es la gestión de compra de insumos por medio de negociación de precio por volumen con sus proveedores; el control de inventario donde se cuide un equilibrio entre uso efectivo del espacio de almacenamiento y reducción de obsolescencia contra la disponibilidad para distribución y por último en la venta efectiva respetando las estacionalidades, tendencias del mercado y satisfacción del cliente.

Las necesidades descritas sumadas a la existencia de múltiples factores que pueden afectar la demanda adicionan interés y complejidad sobre el desarrollo de la tarea, razón por el cual se propone explorar varios modelos con el objetivo de encontrar aquel que mejor se ajuste en la predicción de las unidades de cada producto que se venderá en un horizonte dado; donde se parte de una base de datos que representa la historia del comportamiento de varios productos. Es por tanto que después del correcto tratamiento de la base de datos mediante la depuración de su información y selección de características con mayor importancia, se evalúan modelos de regresión como el *Random Forest Regressor* y veinte modelos adicionales inmersos en *H2O AutoML*, luego de ello, se realiza de nuevo el proceso con el mejor modelo encontrado pero esta vez realizando un análisis de tipicidad para excluir las categorías de los productos que menos valor aportan al modelo y determinar su impacto sobre el *MAPE*, el cual es la métrica para encontrar el mejor modelo. Los resultados basados en la metodología anterior arrojan que el análisis de tipicidad y el uso de *H2O AutoML* son determinantes sobre la mejora en la predicción; ya que optimizan el control sobre complejidad de las categorías al cual pertenecen cada uno de los productos y su amplio rango de unidades demandadas.

Palabras clave: Tipicidad, *H2O AutoML*, *Random Forest*, planeación de la demanda

Tabla de contenido

1. Introducción

La estimación en la planeación de la demanda es un proceso que interviene en la cadena de abastecimiento y que busca predecir en un futuro de corto, mediano o largo plazo la cantidad de producto que será requerida para satisfacer las necesidades de un cliente interno o externo y a la vez sin incurrir en excesos que signifiquen costos de inventario para la compañía.

Generalmente, en el proceso de la estimación de la demanda se debe evaluar las características que pueden ser controladas a través de la oferta comercial y las variables exógenas al proceso. Para el primer caso, podemos encontrar variables relativas al precio, calidad, presentación y funcionalidad y para el segundo caso, puede encontrarse variables relacionadas a temas de orden público, cambios del mercado, situación país o la misma competencia. Normalmente, en el ejercicio de esta labor, se recurre a la generación de analítica descriptiva evaluada en el comportamiento de los datos históricos que presentan valores similares en las características de la oferta que se desea predecir, de manera que se pueda obtener el resultado más confiable. Existen otros ajustes que también son involucrados de acuerdo a la experticia o conocimiento del negocio y estos tienen que ver con la estacionalidad natural en una época del año, necesidades generales del mercado dado la escasez de un producto, conocimiento del ciclo de vida del producto o una fuerte campaña de mercadeo que impulse una compra emocional; todo esto es entonces considerado y luego se determina según el juicio del estimador si puede agregar o restar valor al resultado final.

Dado lo anterior, no es difícil imaginar que las variables que intervienen en el proceso tienen diferentes impactos, origen y subjetividad que suman incertidumbre sobre el dato de interés, sin tener en cuenta que solo por el conocimiento implícito del negocio la estimación dada entre diferentes personas pueden tener entre sí una amplia diferencia; lo que trae como primer requerimiento la necesidad de analizar de una manera estructurada y fundamentada matemáticamente la variable de entrada de tal forma que permita reducir la brecha y le mejore al negocio la sostenibilidad de sus procesos (Anaplan, s.f.). Por tanto, se decide explorar en metodologías basadas en *Machine Learning* con el fin de generar las eficiencias mencionadas mediante el uso de una base de datos de origen confidencial con una dimensionalidad de 265,852 observaciones y 52 características; el cual proporcionan las variables necesarias para describir la oferta comercial de una amplia variedad de productos en diferentes condiciones tales como el precio, temporalidad, tipo de exposición en la tienda ya sea al ocupar un lugar de alto o bajo flujo de clientes u organoléptico el cual permite que el cliente perciba por olor o sabor las propiedades del producto.

Una vez definida la base de datos y la descripción del problema, el objetivo de predicción serán las unidades que se venderán usando modelos definidos para regresión; de manera que una vez se ha limpiado la base de datos original, se inicia con la evaluación del modelo de *Random Forest* (Sklearn.Ensemble.RandomForestRegressor, s.f.) y luego se continúa con la librería de *H2O AutoML* (Ledell & Poirier, 2020) el cual permite la evaluación de varios modelos adicionales que se describirán más adelante.

Los *notebook* se encuentran privados en https://github.com/EdwarHinca/UdeAProject.git, y se debe solicitar acceso a edwar.hincapie@udea.edu.co.

2. Metodología

La base de datos proporcionada cuenta con un tamaño de 265,852 observaciones y 52 características que definen diferentes aspectos de la oferta comercial; básicamente enmarcados en el tiempo de la venta, precio, país de venta, venta generada, unidades vendidas, profundidad de descuento, categoría del producto, línea de negocio, canal de venta, tipo de pago, cantidad de clientes, tipo de oferta, exposición en los canales de venta, contenido de producto, público objetivo, tiempo que lleva el producto en el mercado, definición de fecha de salida del producto y costo del producto. Sin embargo, se observa otras variables que no aportan al dato de interés o pueden ser redundantes, un ejemplo de ello es la definición de las unidades vendidas y su venta correspondiente en dinero ya que también se cuenta con el precio del producto y genera por tanto una dependencia implícita, adicionalmente, también se cuenta con otras variables como la fecha de actualización de la base de datos en el sistema el cual no es una información que afecta al consumidor ni el comportamiento de la oferta comercial.

Finalmente, todo el desarrollo del trabajo que se describirá a continuación referente al tratamiento de datos, entrenamiento de los modelos, evaluación de las métricas y su representación gráfica se ejecutará usando *Python* con el fin de aprovechar su uso por medio de librerías como *Numpy* (Harris et al., 2020), *Matplotlib* (Hunter, 2007), *Pandas* (McKinney, 2010), *Scikit-Learn* (Pedregosa et al., 2011) y *H2O AutoML* el cual facilitará la solución de los requerimientos necesarios en cada punto del proceso.

Preparación de datos

En consecuencia, antes de evaluar algún modelo, se debe llevar a cabo una limpieza de la base de datos y acondicionamiento de las características finales en variables numéricas o categóricas:

1. Para comenzar, se toma un mercado de estudio con el propósito de garantizar homogeneidad en las preferencias culturales, unidades demandadas, penetración en el mercado y tipo de moneda local.
2. Se elimina características redundantes y aquellas que no aportan información respecto la oferta comercial o del mercado.
3. Se identifica los datos nulos, encontrando solo una característica relacionada con la profundidad de descuento y se reemplaza por cero dado que corresponde aquellos productos que son planeados para tener una venta promedio y no se pretende que sean impulsados en venta mediante ofertas por quiebre de precio.
4. Se inspeccionan las demás características y se hacen pequeños ajustes en la manera como se identifican los valores Booleanos y se elimina el 0.02% de las observaciones cuyas características numéricas contenían caracteres erróneos.
5. Se verifica el tipo de cada característica de manera que *Python* la lea como numérica o categórica.
6. Se filtran los canales de venta.

Hasta este punto, la base de datos ya se encuentra limpia y ha quedado con una dimensionalidad de 164,710 observaciones y 29 características, a continuación se llevará a

cabo evaluación de características para reducir dimensionalidad y adecuación de las variables para que puedan ser usadas como entrada en los modelos.

7. En la transformación de los campos categóricos se evalúa el uso del método *Dummy Coding* el cual incrementa la dimensionalidad a 1660 características *mientras Label Encoder* mantiene la dimensionalidad; con el propósito de optimizar el recurso computacional se toma en cuenta la metodología del *Label Enconder* (Sklearn.Preprocessing.LabelEncoder, s.f.)
8. De las características que hasta ahora tenemos en la base de datos, todas ellas describen un aspecto de la oferta comercial, sin embargo, algunas pueden tener mayor relevancia sobre las demás en los modelos de regresión. Por tal motivo, se emplea el algoritmo *RFECV* de *Scikit-Learn* (Sklearn.Feature_selection.RFECV, s.f.) para llevar a cabo la selección de aquellas que son más relevantes. Brevemente, lo que busca el algoritmo es aprovecharse de la importancia o coeficientes asignados para cada característica que arroja un estimador, luego se generan bucles comparando las características entre sí y a la vez computando su importancia, de manera que al final se obtiene como salida las variables con mayor importancia. Como se describe anteriormente, la escogencia del estimador es importante para la salida del algoritmo, de tal forma que se emplea el *RandomForestRegressor* (Sklearn.Ensemble.RandomForestRegressor, s.f.) y el *LinearRegression* (Sklearn.Linear_model.LinearRegression, s.f.), tomando de ellos los resultados con el uso del *RandomForestRegressor* el cual arroja una reducción de dimensionalidad a 22 características.
9. Por último, se toma un 80% de los datos para entrenamiento y el restante para su posterior validación en cada uno de los modelos; donde a su vez son normalizados mediante *StandardScaler* (Sklearn.Preprocessing.StandardScaler, s.f.).

A continuación en la Figura 1, se resume el proceso descrito:

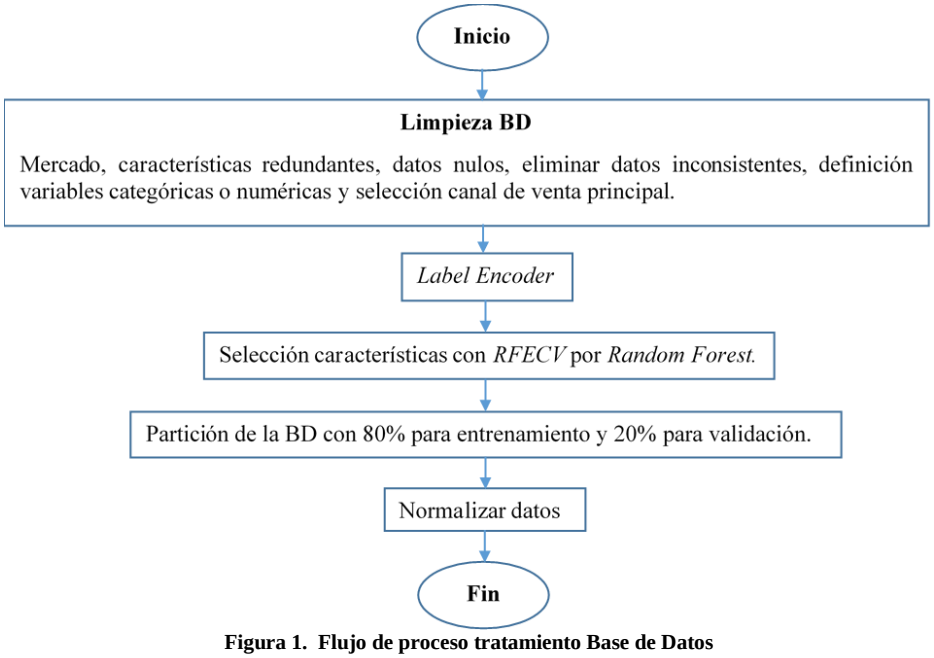

Figura 1. Flujo de proceso tratamiento Base de Datos

3. Análisis

3.1. Métrica

La métrica establecida para elegir el mejor modelo será el *MAP*E (Swamidass, 2004) a razón de que prevalece la importancia de medirla para cada uno de los productos del portafolio y no como un conjunto de datos; dado que cada producto representa oportunidad de venta o exceso de inventario para el negocio:

$$MAPE = \frac{ABS(Y\_True - Y\_Pred)}{Y\_True} * 100\% \qquad \textbf{(3.1)}$$

Donde *Y_true* representa al valor verdadero y *Y_pred* a la predicción del modelo.

3.2. Modelos de predicción

3.2.1. Random Forest Regressor

Una vez se ha establecido una base de datos con 22 características, se puede suponer que algunas de ellas presentan más importancia que otras sobre la variable objetivo; es por esto que aprovechando esta naturaleza, se comienza por la implementación de un *Random Forest Regressor* variando los siguientes hiperparámetros definidos en la librería de *Scikit-Learn* (Sklearn.Ensemble.RandomForestRegressor, s.f.) para encontrar el mejor resultado:

- *n_estimators*:[20,60,200, 400,500, 800]
- *max_depth*:[8,10,15,20]
- *max_features*:[10,17,26]

Una vez se ha entrenado el modelo, se encuentra que la mejor combinación de hiperparámetros que el modelo arroja es la siguiente:

RandomForestRegressor(max_depth=20, max_features=10, n_estimators=800)

Ahora, calculando el *MAPE* para cada uno de los productos, se obtiene el histograma de la Figura 2 el cual representa a una frecuencia máxima de 548 datos con un *MAPE* igual a 35,17%; sin embargo, se puede observar una gran proporción de datos que se desbordan por encima de este resultado; lo cual no es aceptable para los propósitos de negocio requeridos:

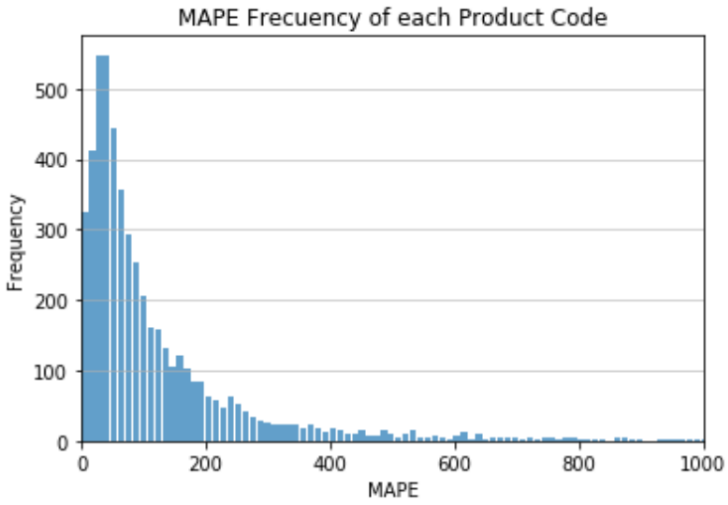

Figura 2. Histograma *MAPE Random Forest Regressor*

3.2.2. H2OAutoML

H2O es una plataforma muy útil escrita en Java para generar modelos predictivos y que se basa en el entrenamiento y evaluación de una base de datos dentro de un conjunto de algoritmos que buscan ajustarse de acuerdo a la combinación de diferentes hiperparámetros y luego de ello se toma aquel modelo que por defecto genera menor desviación en la media residual (Ledell & Poirier, 2020). Los algoritmos incluidos en la ejecución del modelo son:

- *DRF: Incluye Random Forest y Extremely Randomized Trees (XRT)*
- *GLM*
- *XGBoost (XGBoost GBM)*
- *GBM (H2O GBM)*
- *DeepLearning*
- *StackedEnsemble*

Una vez definido lo anterior, se condiciona para este caso el entrenamiento con 20 modelos usando el 80% de los datos y luego se validan los resultados con el 20% de los datos restantes. Así se obtiene de manera ordenada los modelos con menor desviación hasta el modelo de mayor desviación en cuánto a su media residual.

De esta manera, podemos ver en la Tabla 1 no solo la media residual si no también las diferentes métricas que son comunes para los modelos de regresión:

model_id	mean_residual_deviance	rmse	mse	mae	rmsle	training_time_ms	predict_time_per_row_ms
StackedEnsemble_AllModels_AutoML_20210317_153100	9.49289e+06	3081.05	9.49289e+06	1102.73	nan	2129	0.13504
StackedEnsemble_BestOfFamily_AutoML_20210317_153100	9.75903e+06	3123.94	9.75903e+06	1120.36	nan	2141	0.177619
XGBoost_grid__1_AutoML_20210317_153100_model_3	1.00086e+07	3163.64	1.00086e+07	1160.93	nan	516986	0.023827
XGBoost_grid__1_AutoML_20210317_153100_model_4	1.02502e+07	3201.59	1.02502e+07	1222.1	nan	77951	0.010793
GBM_grid__1_AutoML_20210317_153100_model_2	1.09462e+07	3308.5	1.09462e+07	1162.75	nan	25149	0.051012
XGBoost_2_AutoML_20210317_153100	1.14508e+07	3383.9	1.14508e+07	1275.23	nan	40868	0.006541
GBM_4_AutoML_20210317_153100	1.16796e+07	3417.55	1.16796e+07	1216.1	nan	25878	0.040213
XGBoost_1_AutoML_20210317_153100	1.17132e+07	3422.46	1.17132e+07	1335.15	nan	37709	0.004904
GBM_3_AutoML_20210317_153100	1.17825e+07	3432.56	1.17825e+07	1235.9	nan	25395	0.045796
GBM_2_AutoML_20210317_153100	1.18582e+07	3443.57	1.18582e+07	1256.28	nan	21301	0.040777
XGBoost_grid__1_AutoML_20210317_153100_model_2	1.18953e+07	3448.95	1.18953e+07	1233.34	nan	27711	0.003611
GBM_1_AutoML_20210317_153100	1.21966e+07	3492.37	1.21966e+07	1263.98	nan	20193	0.040706
GBM_grid__1_AutoML_20210317_153100_model_1	1.23052e+07	3507.88	1.23052e+07	1297.01	nan	22342	0.064547
XGBoost_3_AutoML_20210317_153100	1.25725e+07	3545.77	1.25725e+07	1432.04	nan	31116	0.003464
GBM_5_AutoML_20210317_153100	1.30409e+07	3611.22	1.30409e+07	1292.38	nan	40632	0.064009
DRF_1_AutoML_20210317_153100	1.33989e+07	3660.45	1.33989e+07	1206.15	0.862706	57392	0.01539
XGBoost_grid__1_AutoML_20210317_153100_model_1	1.35215e+07	3677.16	1.35215e+07	1511.19	nan	21678	0.002917
DeepLearning_grid__1_AutoML_20210317_153100_model_1	1.55351e+07	3941.45	1.55351e+07	1527.12	nan	598387	0.099835
DeepLearning_1_AutoML_20210317_153100	1.71133e+07	4136.82	1.71133e+07	1619.52	nan	18476	0.00851
DeepLearning_grid__2_AutoML_20210317_153100_model_1	1.82324e+07	4269.94	1.82324e+07	1612.09	nan	1.15498e+06	0.111026

Tabla 1. Resultados entrenamiento *H2O AutoML*

De acuerdo a lo anterior, se emplea el mejor modelo para predecir los datos del conjunto de validación y posteriormente calcular su *MAPE*. Es así que se encuentra una mejora significativa versus el *Random Forest* al tener en cuenta que se abarca mayor cantidad de productos con una métrica más baja. Esto se puede observar en la Figura 3 que representa el paso de una frecuencia máxima de 548 a 837 productos con un *MAPE* de 35,17% a 32,63%.

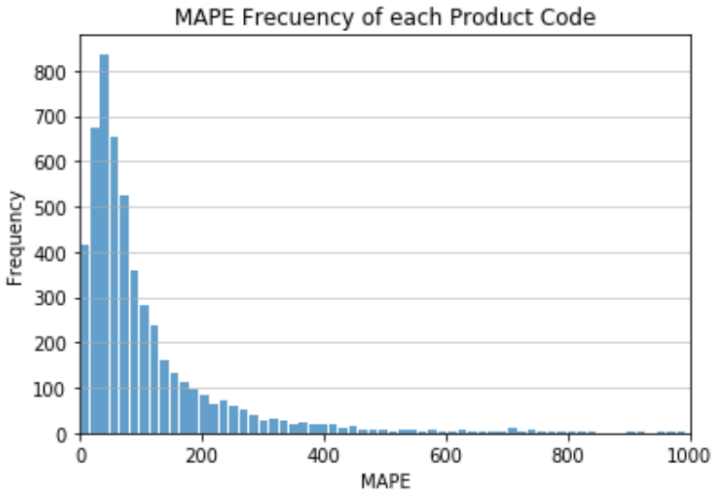

Figura 3. Histograma *MAPE* mejor modelo *H2O AutoML*

Si bien la mejora en el *MAPE* es aproximadamente del 3%, se destaca la capacidad de acortar la dispersión en los resultados teniendo en cuenta la heterogeneidad de los productos

considerados en la base de datos y la sensibilidad del modelo para detectarlo; es así que con la necesidad de comprender un poco más cómo se comportan las diferentes variables, se desplegará a continuación varias perspectivas sobre el comportamiento mencionado.

En primer lugar, se verá entonces el análisis residual resultante entre la predicción y el conjunto de validación; donde en promedio la recta trazada entre los puntos alrededor del cero para valores menores a 30,000 indican un buen equilibrio en predicción por encima y por debajo del valor real, agregando que justo en este rango es donde se ubica la mayor parte de los datos. Congruentemente con lo anterior, entonces para valores mayores a 30,000 la dispersión de los datos es mucho mayor y esto se debe posiblemente a que hay pocos datos después de este umbral que le permita al modelo ajustarse apropiadamente en su entrenamiento y por tanto arroja valores erróneos, ver Figura 4.

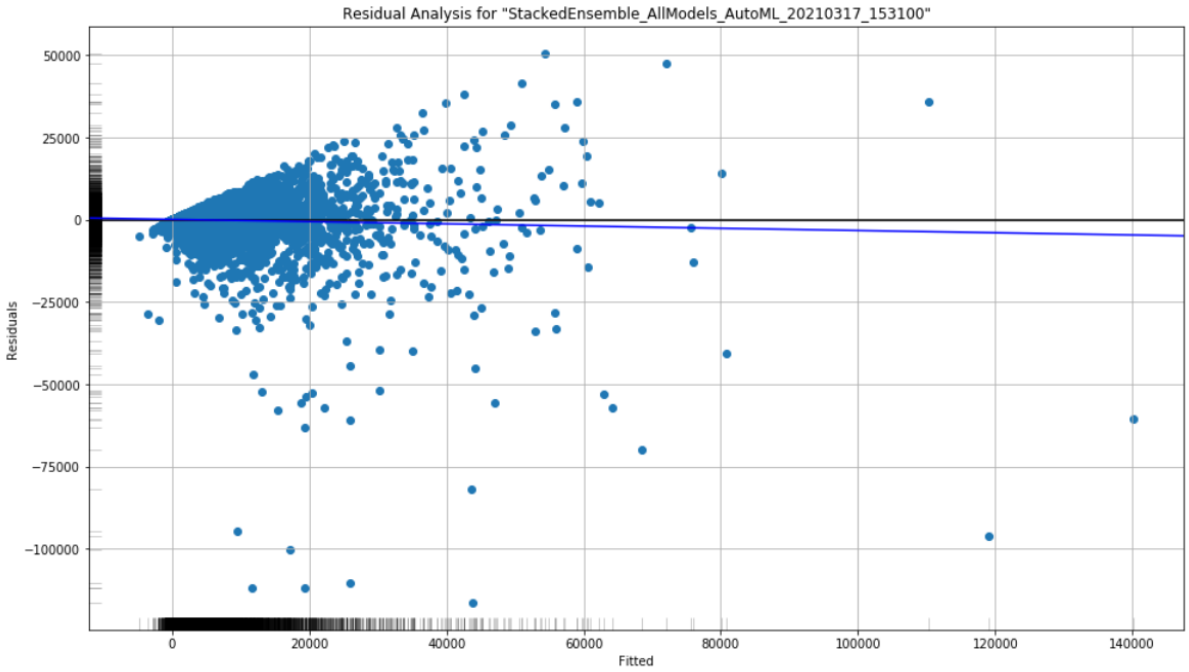

Figura 4. Análisis de residuales del mejor modelo encontrado en *H2O AutoML*

Si deseamos conocer que variables son las que más nos afectan en nuestro modelo, en la Figura 5 se destacan que las variables relacionadas con el precio o nivel de descuento en los productos son las más relevantes, resultando lógico que para un mercado cualquiera al detectar oportunidades de compra con menor inversión promuevan las unidades de ventas en el negocio, sin embargo, resulta también importante la presencia de otras variables como la exposición del producto en la tienda, la estacionalidad de la venta, la posibilidad que el cliente pueda percibir la calidad del producto mediante estimulaciones organolépticas y el hecho de que algunos productos tienen más relevancia sobre otros dada su posición natural en el mercado:

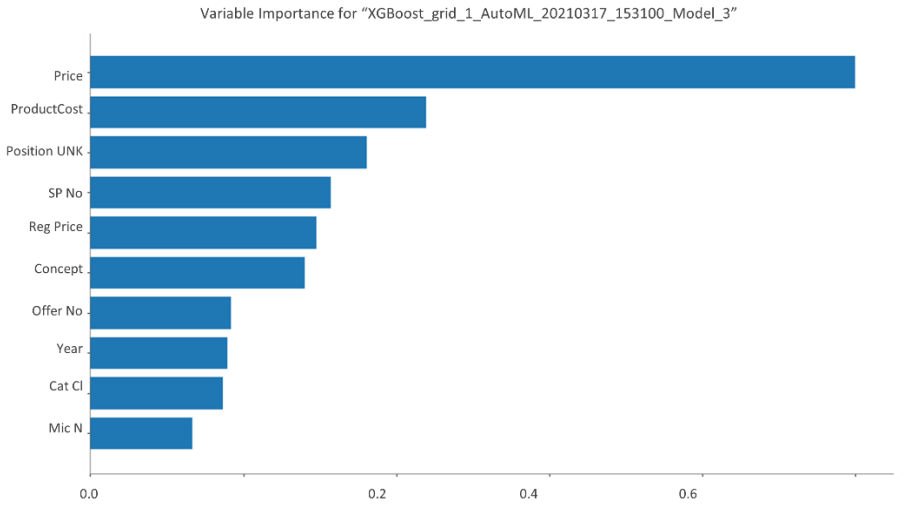

Figura 5. Importancia de las variables en el mejor modelo de *H2O AutoML*

En este orden de ideas, cuando queremos observar el comportamiento del precio del producto en cada uno de los modelos respecto a la respuesta media, tenemos que la mayoría de los modelos presentan una sensibilidad alta en los niveles de precio más bajos y luego se estabiliza a medida el precio aumenta, el cual dentro de la lógica de la demanda, se espera que los productos de menor valor puedan tener mayor demanda y alta variación dado que su adquisición requiere de menor esfuerzo y recíprocamente, luego de pasar cierto umbral en precio, la demanda parece ser más estable ya que el segmento de clientes se reduce a razón de que el esfuerzo económico es más alto para obtener el producto (Ver Figura 6). Por el contrario, hay dos modelos que no se comportan igual; uno de ellos es el modelo identificado como *GLM_1* el cual muestra una sensibilidad constante en cada nivel de precio y el modelo *DeepLearning_grid__1_model_1* cuyo comportamiento en los bajos niveles de precio parece ser similar a los demás modelos pero cuando está por encima de valores de 120,000 su comportamiento es monótonamente creciente (Kannan & Krueger, 1996).

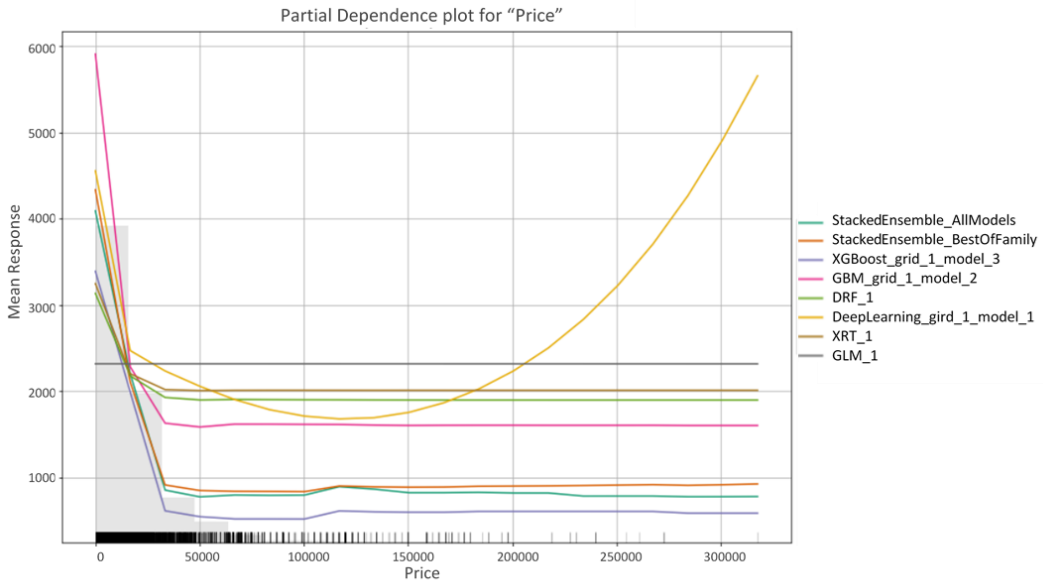

Figura 6. Dependencia característica de mayor relevancia sobre modelos de *H2O AutoML*

Dejando de lado el precio del producto, exploraremos ahora como influye su exposición sobre la variable objetivo, por lo que tomaremos la variable más influyente en este aspecto como se observa en la Figura 7. De acuerdo al conocimiento de cada una de las exposiciones, se resalta la respuesta del modelo *StackedEnsemble_AllModels* sobre la variable *"CONTR"* el cual es muy coherente con las condiciones reales del negocio, pues esta exposición es diseñada para ubicar los productos de mayor impacto en unidades. Similar a lo anterior, se observa una respuesta acorde en las demás variables aunque se destaca de nuevo la sensibilidad constante del modelo *GLM_1*, lo que puede reflejar que éste sería el modelo que menos se adapta a nuestros intereses.

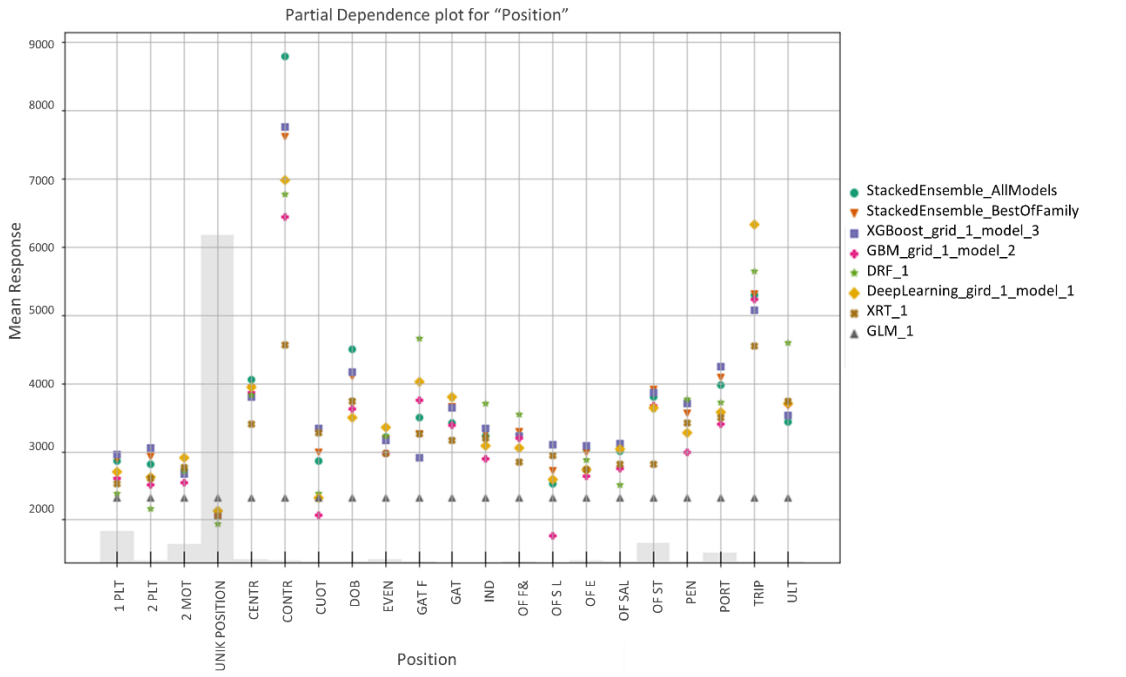

Figura 7. Dependencia característica de mayor relevancia en la exposición sobre modelos de *H2O AutoML*

Referente a la estacionalidad, podemos ver en la Figura 8 que la sensibilidad es ligeramente estable presentando picos y algunos valores bajos en los puntos que históricamente se espera esta tendencia y para la exposición respecto a cualidades organolépticas en la Figura 9 la sensibilidad es mayor para aquellos productos que tienen esas propiedades.

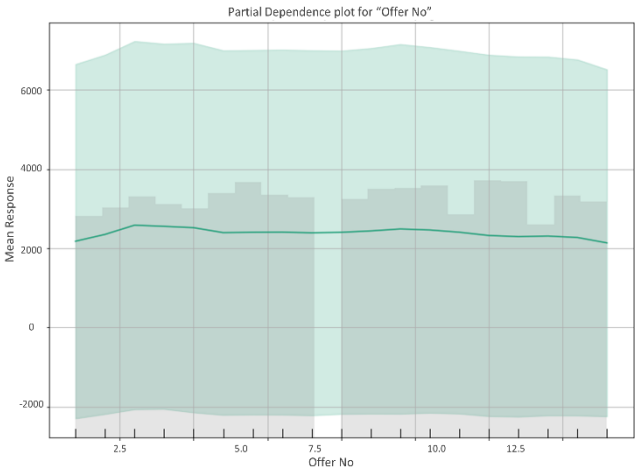

Figura 8. Respuesta en la estacionalidad

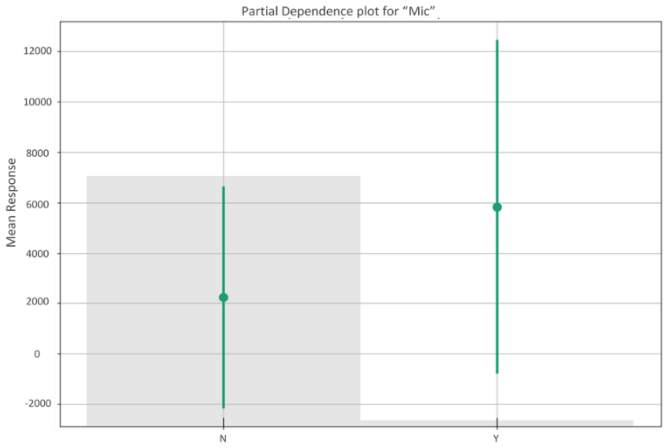

Figura 9. Respuesta propiedades organolépticas

3.2.3. H2OAutoML con análisis de tipicidad

Como se ha descrito en el análisis de la Figura 4, los residuales nos indican que el mejor modelo no tiene una buena aproximación para algunos productos y adicionalmente según la Figura 5, dentro de la importancia de las variables se destaca que el tipo de producto tiene una buena influencia sobre la predicción; por tal motivo, se decide llevar a cabo un análisis del grado de tipicidad con las categorías generales de los productos y discriminar de acuerdo a dicho análisis las categorías que no aportan valor al modelo; luego de esto, se entrena de nuevo el modelo y se analizan los resultados.

El grado de tipicidad clasifica al promedio de las muestras que son consideradas las más representativas de una categoría y es usado precisamente para la búsqueda de las características más comunes y aquellas que pueden ser discriminadas. El grado de tipicidad está definido por las siguientes funciones (Botía Valderrama & Botía Valderrama, 2018):

$$\mu_c(x) = \frac{1}{1+dis(x,x_0)} \qquad \textbf{(3.2)}$$

Donde $\mu_c(x)$ representa a la función de Cauchy y $dis(x, x_0)$ la distancia entre dos puntos que para este caso tomaremos la distancia euclidiana de un punto x respecto a la media de la categoría al cual pertenece dicho punto.

Luego de definir la función de Cauchy, se determina a continuación el grado de pertenencia G donde δ representa a un parámetro de ajuste en el cual normalmente es igual a 0,1.

$$G = \mu_c(x) * \delta \qquad \textbf{(3.3)}$$

De esta manera, mientras mayor sea el grado de pertenencia indica que aporta más información y análogamente, mientras más bajo sea, la característica aporta menos información.

Teniendo claro el concepto, se aplica para las diferentes categorías como se muestra en la Figura 10 para determinar si alguna de ellas aporta o no información, de no ser así, se excluye de la base de datos y se entrena de nuevo el modelo.

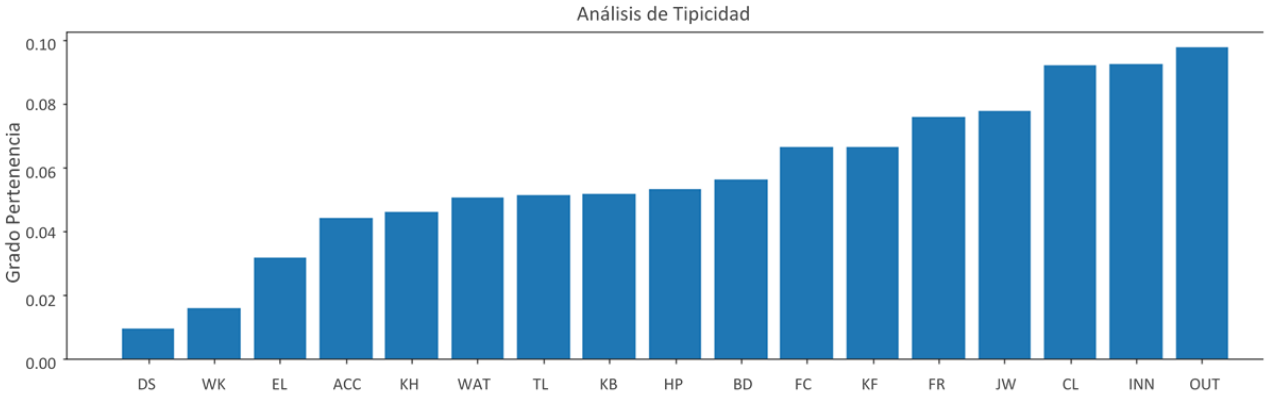

Figura 10. Análisis de tipicidad en las categorías

Los resultados del modelo entrenado son los siguientes una vez se eliminan las tres categorías con menor aporte de información o menor grado de pertenencia:

model_id	mean_residual_deviance	rmse	mse	mae	rmsle	training_time_ms	predict_time_per_row_ms
StackedEnsemble_AllModels_AutoML_20210414_041800	9.19577e+06	3032.45	9.19577e+06	1040.47	nan	2514	0.172219
StackedEnsemble_BestOfFamily_AutoML_20210414_041800	9.47675e+06	3078.43	9.47675e+06	1055.44	nan	831	0.033393
XGBoost_grid__1_AutoML_20210414_041800_model_2	9.69551e+06	3113.76	9.69551e+06	1058.55	nan	70270	0.009849
XGBoost_grid__1_AutoML_20210414_041800_model_4	9.75732e+06	3123.67	9.75732e+06	1067.19	nan	71466	0.009888
XGBoost_2_AutoML_20210414_041800	1.09542e+07	3309.72	1.09542e+07	1256.94	nan	47035	0.006859
XGBoost_1_AutoML_20210414_041800	1.11934e+07	3345.66	1.11934e+07	1297.09	nan	47017	0.00578
XGBoost_grid__1_AutoML_20210414_041800_model_1	1.12102e+07	3348.16	1.12102e+07	1107.2	nan	74863	0.005681
GBM_grid__1_AutoML_20210414_041800_model_2	1.12774e+07	3358.19	1.12774e+07	1184.37	nan	9891	0.018666
GBM_4_AutoML_20210414_041800	1.13997e+07	3376.34	1.13997e+07	1206.33	nan	18467	0.03373
GBM_3_AutoML_20210414_041800	1.1403e+07	3376.83	1.1403e+07	1224.75	nan	19514	0.042979
GBM_2_AutoML_20210414_041800	1.14611e+07	3385.42	1.14611e+07	1240.08	nan	19045	0.04448
GBM_1_AutoML_20210414_041800	1.16322e+07	3410.61	1.16322e+07	1258.74	nan	17238	0.04068
XGBoost_grid__1_AutoML_20210414_041800_model_3	1.18889e+07	3448.03	1.18889e+07	1378.96	nan	39876	0.004316
XGBoost_3_AutoML_20210414_041800	1.23994e+07	3521.28	1.23994e+07	1420.37	nan	31224	0.003419
GBM_5_AutoML_20210414_041800	1.2712e+07	3565.39	1.2712e+07	1284.45	nan	39242	0.075058
DRF_1_AutoML_20210414_041800	1.27592e+07	3572.01	1.27592e+07	1184.65	0.849912	51226	0.016481
GBM_grid__1_AutoML_20210414_041800_model_1	1.59566e+07	3994.57	1.59566e+07	1572.5	nan	6440	0.027018
DeepLearning_1_AutoML_20210414_041800	1.79892e+07	4241.36	1.79892e+07	1655.69	nan	16422	0.0079
XRT_1_AutoML_20210414_041800	1.85399e+07	4305.8	1.85399e+07	1742.43	1.4965	22281	0.0145
DeepLearning_grid__1_AutoML_20210414_041800_model_1	1.88738e+07	4344.4	1.88738e+07	1548.62	nan	1.25946e+06	0.095585

Tabla 2. Resultados entrenamiento *H2O AutoML* con análisis de tipicidad

Si se compara las métricas de error arrojadas por el *H2O AutoML* con y sin análisis de tipicidad Tabla 1 y Tabla 2, no varían mucho entre sí, no obstante, al generar el *MAPE* por producto los resultados son mucho mejores ya que se eliminan los productos que no aportan información en la generación del modelo; obteniendo por tanto una frecuencia máxima de 2,525 productos con un *MAPE* en dicha frecuencia igual a 34,7%:

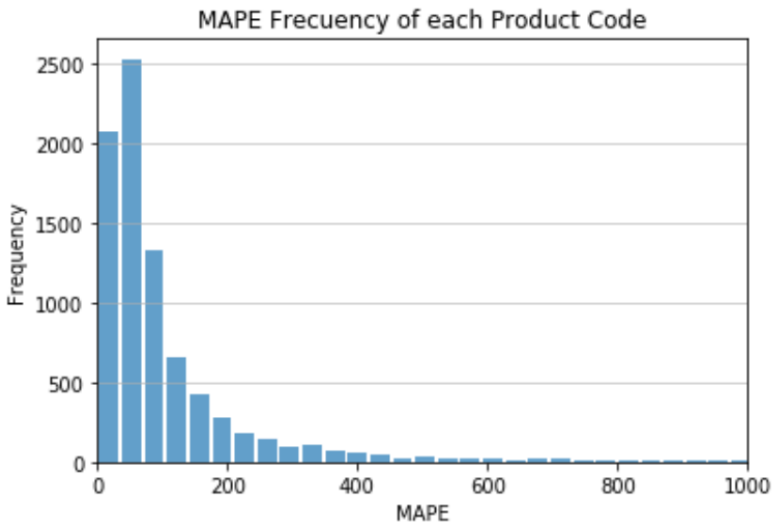

Figura 11. Histograma *MAPE* mejor modelo *H2O AutoML* y análisis de tipicidad

De igual manera, podemos ver mejoras en las gráficas expuestas en el item anterior, un ejemplo de ello se observa que los *outliers* en valores menores a 20,000 disminuye (Figura 12):

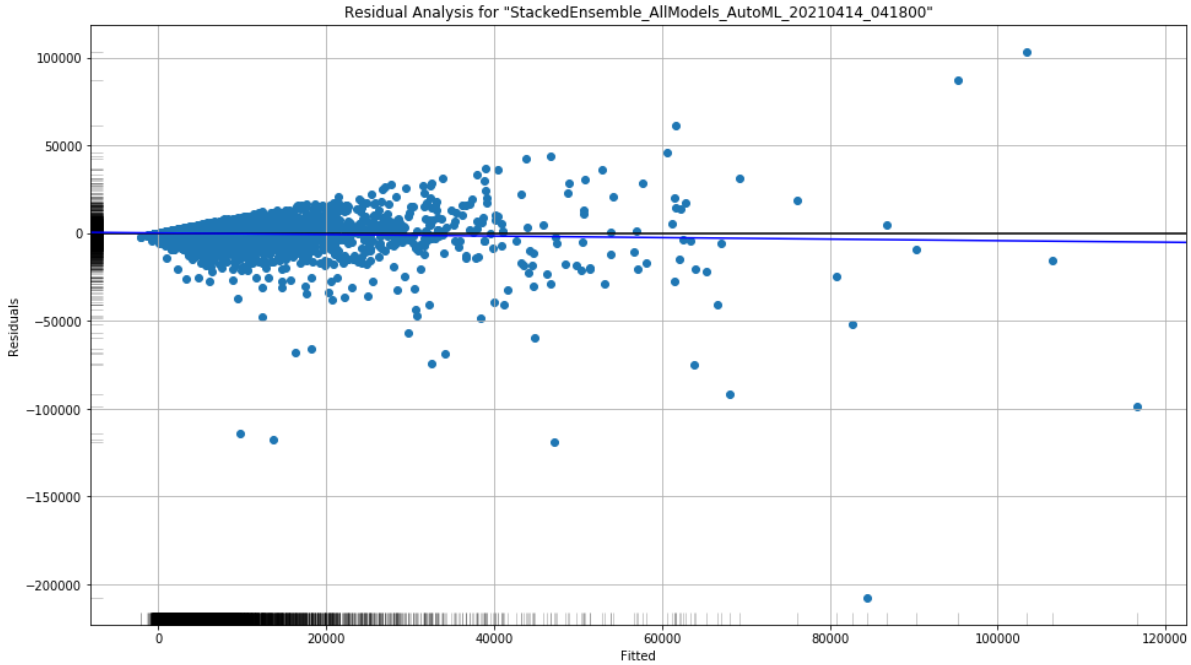

Figura 12. Análisis de residuales del mejor modelo encontrado en *H2O AutoML* y análisis de tipicidad

En relación a la importancia de variables, el precio la exposición, estacionalidad y tipo de producto siguen siendo las más relevantes Figura 13:

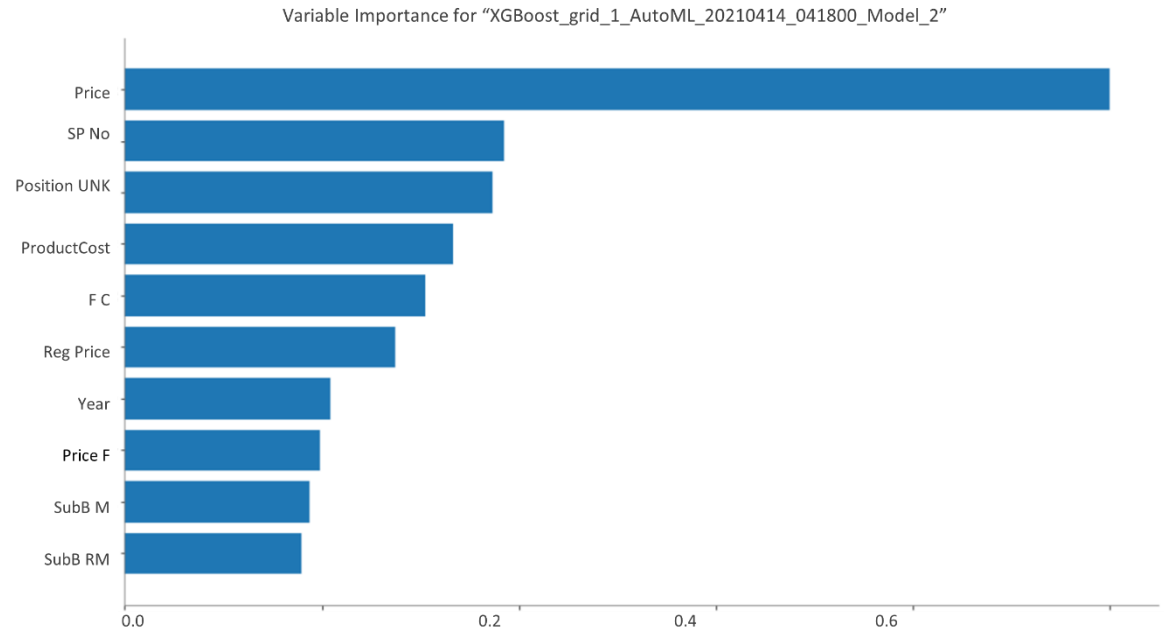

Figura 13. Importancia de las variables en el mejor modelo de *H2O AutoML* con análisis de tipicidad

La respuesta media en el precio sigue siendo abrupto para el modelo de *DeepLearning_1* el cual esta vez pasa a comportarse monótonamente decreciente en precios altos y en el modelo *GLM_1* lo que observamos es una sensibilidad constante para cada nivel de precio como se observó en el apartado anterior, ver Figura 14:

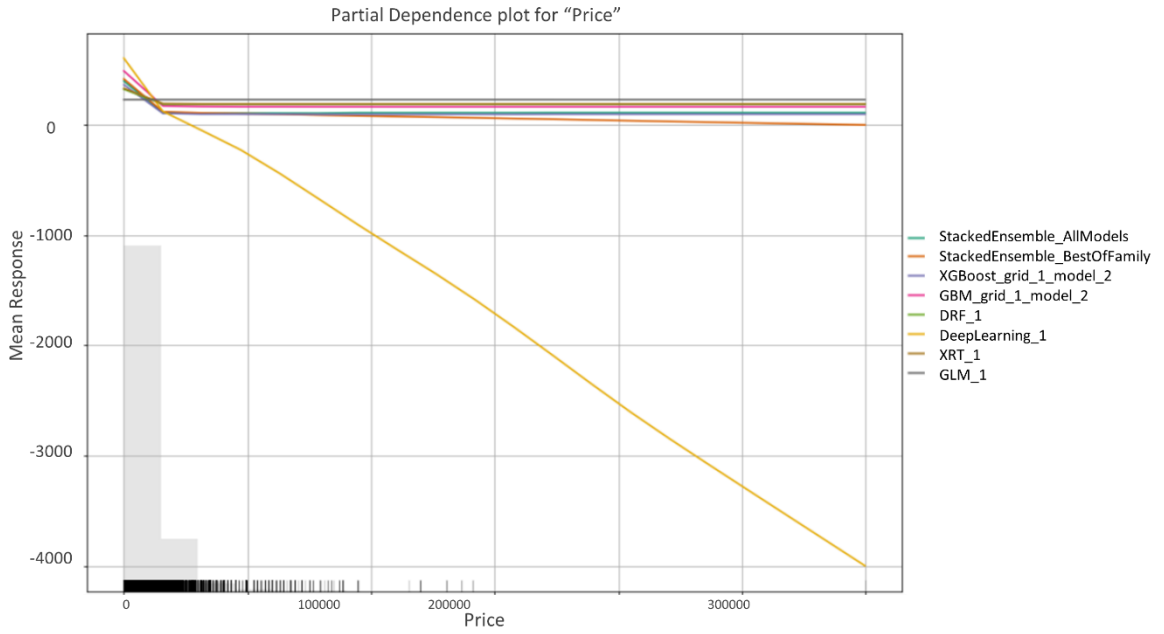

Figura 14. Dependencia característica de mayor relevancia sobre modelos de H2O AutoML y análisis de tipicidad

Finalmente, la sensibilidad en la exposición (Figura 15), estacionalidad (Figura 16) y propiedades organolépticas (Figura 17) son similares al item anterior:

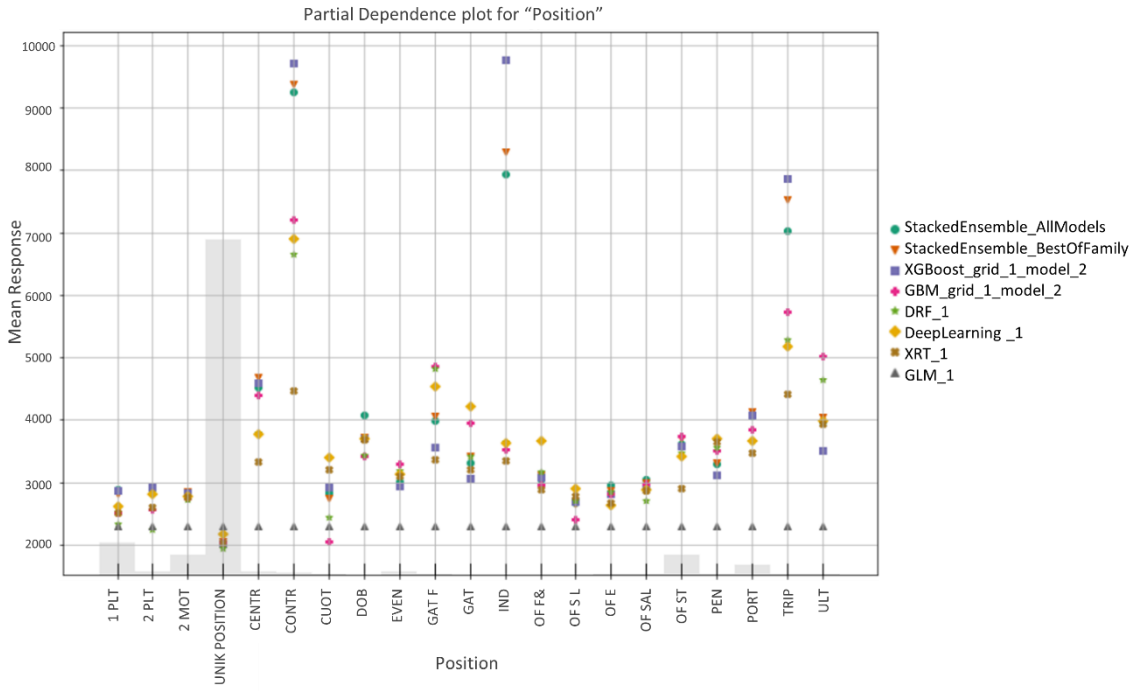

Figura 15. Característica de mayor relevancia en la exposición sobre modelos de *H2O AutoML* y análisis de tipicidad

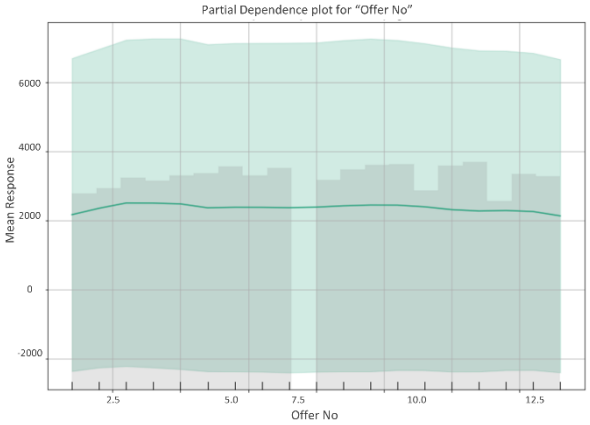

Figura 16. Respuesta en la estacionalidad con tipicidad

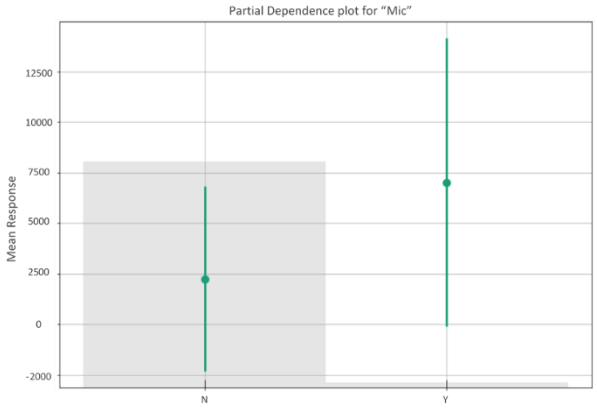

Figura 17. Respuesta propiedades organolépticas con tipicidad

3.3. Elección del modelo

De acuerdo a la Tabla 3, se toma como mejor modelo el obtenido mediante el análisis de tipicidad usando *H2O AutoML*; del cual predomina *StackedEnsemble_AllModels* como la mejor opción. Es importante mencionar que si nuestra decisión es fundamentada solo por medio de la comparación del menor *MAPE*, podríamos incurrir en un error, pues no basta sólo con tener la mejor media en la métrica si no también que impacte la mayor cantidad de productos posibles; por consiguiente, tener una muy buena predicción solo para una cantidad

pequeña de productos puede generar grandes errores en los demás, originando una afectación negativa en la negociación de precios por volumen, rotación de inventario y finalmente sobre la rentabilidad real.

Modelo	***Mape***	***Frecuencia Máxima***
StackedEnsemble_AllModels análisis de tipicidad	34,7%	2,525
StackedEnsemble_AllModels	32,6%.	837
Random Forest Regressor	35,2%	548

Tabla 3. Comparación de modelos

4. Conclusiones y recomendaciones

La demanda de un producto depende de múltiples factores; donde para la base de datos estudiada, los que tienen mayor peso son los que describen su precio, profundidad de descuento, la exposición que tienen en la tienda y dado un nicho de mercado se encuentra preferencias sobre categorías específicas de productos. Dadas estas preferencias, unas aportan más valor sobre la predicción de la demanda que otras, por tal razón, el análisis de la tipicidad permite encontrar cuáles son esas categorías más importantes para un modelo y esto se refleja sobre la optimización en el *MAPE*. Sumado a lo anterior, las bondades que ofrece el *H2O AutoML* al evaluar varios modelos, tener la capacidad de describir las variables más relevantes y tomar de todos los modelos lo mejor de ellos para unirlos en uno solo, condujo a la mejor adaptación que se encontró para el análisis que se ha descrito.

Profundizando sobre el análisis de tipicidad y dada la variedad en las categorías de los productos, se debe tener en cuenta que hay algunas categorías muy diferentes a las demás y esa diferencia le introduce errores al modelo ya que éste busca ajustarse a todas las condiciones de entrenamiento. El propósito del análisis de tipicidad resuelve esta incógnita al detectar esas categorías que no aportan valor al modelo; luego al filtrarlas y entrenar de nuevo el modelo se observa una mejora importante sobre el alcance del mismo.

Para futuros análisis, sería muy interesante revisar la posibilidad de generar uno o varios modelos que se adapten en diferentes rangos de precio; pues dado el aumento en la dispersión de los datos respecto su precio Figura 12 y la gráfica de la dependencia parcial del precio en cada uno de los modelos Figura 14, sugiere profundizar más en el análisis de esta variable para generar un incremento en la mejora de la predicción.

5. Agradecimientos

Especiales agradecimientos para mi tutor Javier Fernando Botia quien fue clave en el desarrollo del proyecto ya que con su guía y tiempo se condensó el conocimiento para la obtención de los resultados descritos, la empresa que ha proporcionado la base de datos el cual no solo permitió tener la información para evaluar los modelos si no que dada la complejidad de su realidad se obtuvo conocimiento importante sobre la interacción de varias características y comportamientos del mercado y finalmente para todo el plantel de profesores de la Especialización pues impartieron sin mesura todo su conocimiento cumpliendo todas las expectativas esperadas al iniciar este trayecto en la ciencia de datos.

6. Referencias bibliográficas

Anaplan. (n.d.). *Demand Planning Beginners Guide*. Retrieved April 30, 2021, from https://www.anaplan.com/blog/demand-planning-fundamentals-and-futures/

Botía Valderrama, J. F., & Botía Valderrama, D. J. L. (2018). On LAMDA clustering method based on typicality degree and intuitionistic fuzzy sets. *Expert Systems with Applications*, *107*, 196–221. https://doi.org/10.1016/j.eswa.2018.04.022

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825). https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3). https://doi.org/10.1109/MCSE.2007.55

Kannan, R., & Krueger, C. K. (1996). Monotone Functions. In *Advanced Analysis*. https://doi.org/10.1007/978-1-4613-8474-8_2

Ledell, E., & Poirier, S. (2020). *H2O AutoML: Scalable Automatic Machine Learning*. https://scinet.usda.gov/user/geospatial/#tools-and-software

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. https://doi.org/10.25080/Majora-92bf1922-00a

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

sklearn.ensemble.RandomForestRegressor. (n.d.). Retrieved March 20, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

sklearn.feature_selection.RFECV. (n.d.). Retrieved March 20, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html

sklearn.linear_model.LinearRegression. (n.d.). Retrieved March 20, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

sklearn.preprocessing.LabelEncoder. (n.d.). Retrieved March 20, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

sklearn.preprocessing.StandardScaler. (n.d.). Retrieved March 20, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn-preprocessing-standardscaler

Swamidass, P. M. (2004). *Encyclopedia of production and manufacturing management*. Springer Science + Business Media.