Laboratorio de: ANALÍTICA DE DATOS – BIG DATA

Estudiante: Danny Sebastián Díaz Padilla

Práctica No.: 6

Tema: Auto modelamiento de datos usando RapidMiner

Objetivos:

- Utilizar el auto modelador de la herramienta Rapid Miner.
- Predecir el precio de una casa a partir de un conjunto de datos.
- Encontrar el patrón que influye en la compra de una casa.

Marco teórico:

Generalized Linear Model

Investigan la relación entre una variable de respuesta y uno o más predictores. Una diferencia práctica entre ellos es que las técnicas de modelo lineal general por lo general se utilizan con variables de respuesta categóricas. La regresión de mínimos cuadrados normalmente se utiliza con variables de respuesta continuas.

Tanto las técnicas de modelo lineal general como las técnicas de regresión de mínimos cuadrados estiman los parámetros del modelo de forma que se optimice el ajuste del modelo. La regresión de mínimos cuadrados minimiza la suma de los errores al cuadrado para obtener estimaciones de máxima verosimilitud de los parámetros. Los modelos lineales generales obtienen estimaciones de máxima verosimilitud de los parámetros utilizando un algoritmo iterativo de mínimos cuadrados reponderados. [1]

Deep Learning

El Deep Learning es una técnica de aprendizaje automático que enseña a los ordenadores a hacer lo que resulta natural para las personas: aprender mediante ejemplos. El Deep Learning es una tecnología clave presente en los vehículos sin conductor que les permite reconocer una señal de stop o distinguir entre un peatón y una farola. Resulta fundamental para el control mediante voz en dispositivos tales como teléfonos, tabletas, televisores y altavoces manos libres. [2]

Decision Tree

Un Árbol de Decisión (o Árboles de Decisiones) es un método analítico que a través de una representación esquemática de las alternativas disponible facilita la toma de mejores decisiones, especialmente cuando existen riesgos, costos, beneficios y múltiples opciones.

El nombre se deriva de la apariencia del modelo parecido a un árbol y su uso es amplio en el ámbito de la toma de decisiones bajo incertidumbre (Teoría de Decisiones) junto a otras herramientas como el Análisis del Punto de Equilibrio. [3]

Random Forest

Random Forest es una técnica basada en árboles de decisión, que soluciona el problema que tienen los árboles de decisión de no servir para reproducir escenarios predictivos what-if.

Con el árbol de decisión veíamos qué variables podían predecir una variable objetivo determinada, pero no podíamos saber la importancia de cada variable.

Con Random Forest sí vamos a poder conocer la importancia de cada variable.

Al igual que en los árboles de decisión, la variable objetivo en un Random Forest puede ser categórica o cuantitativa. Y el grupo de variables explicativas también. [4]

Gradient Boosted Trees

Un modelo impulsado por gradiente es un conjunto de modelos de árbol de regresión o clasificación. Ambos son métodos de conjunto de aprendizaje directo que obtienen resultados predictivos a través de estimaciones mejoradas gradualmente.

Boosting es un procedimiento de regresión no lineal flexible que ayuda a mejorar la precisión de los árboles. Al aplicar secuencialmente algoritmos de clasificación débiles a los datos modificados de forma incremental, se crean una serie de árboles de decisión que producen un conjunto de modelos de predicción débiles. Si bien impulsar árboles aumenta su precisión, también disminuye la velocidad y la capacidad de interpretación humana. El método de aumento de gradiente generaliza el aumento de árboles para minimizar estos problemas. [5]

Support Vector Machine

El Support Vector Machine (SVM) es un modelo supervisado de aprendizaje con algoritmos asociados que analizan los datos y reconocen patrones, que se utiliza para la clasificación y el análisis de regresión en la Inteligencia de Negocios.

El SVM básico toma un conjunto de datos de entrada y predice, para cada entrada dada, a cuál de las dos clases de salida pertenece, por lo que es un clasificador no-probabilístico lineal binario (solo escoge entre 2 opciones). Dado un conjunto de ejemplos de entrenamiento, cada uno marcado como perteneciente a una de dos categorías, un algoritmo de entrenamiento construye un modelo que asigna nuevos ejemplos en una categoría u otra. [6]

Desarrollo de la práctica:

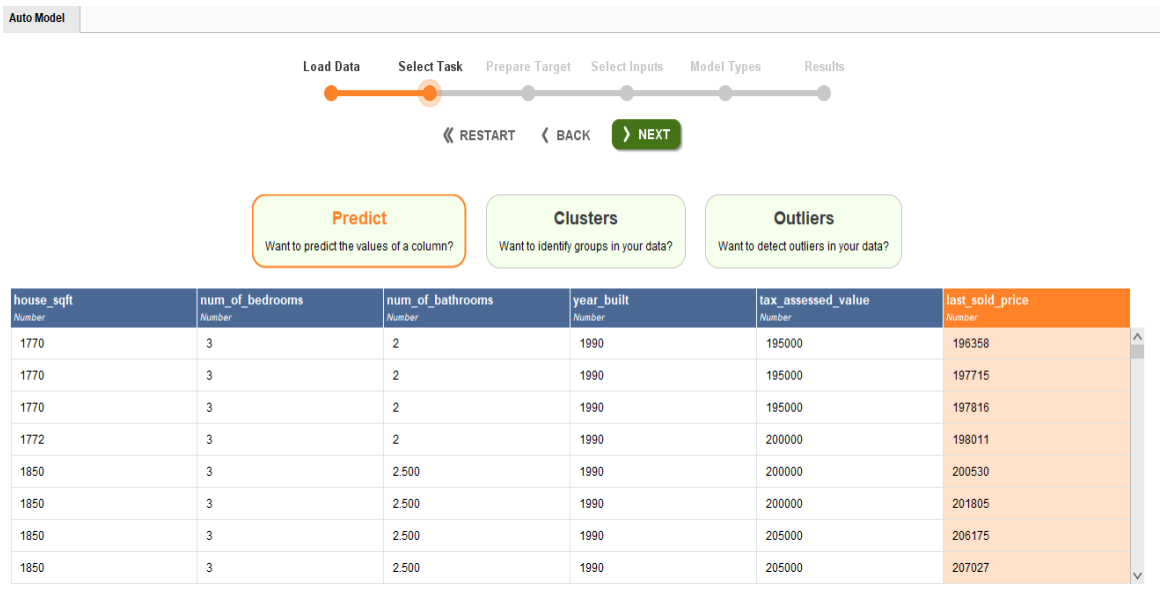
1. **Importar data de casas en venta de una Inmobiliaria.**

En el archivo CSV HousePrices-Dataset1.csv, se encuentran las siguientes columnas con varia información acerca de las casas de una inmobiliaria.

house_sqft – superficie de la casa
num_of_bedrooms - Número de habitaciones
num_of_bathrooms - Número de baños
year_built - El año en que se construyó la casa
tax_assessed_value - Valor de acceso fiscal de la casa
last_sold_price - Último precio de venta de la casa
rate_per_sqfoot – Tasa por pie cuadrado.
home_type – (value apartment, townhouse or single family home)
school_rating_1to10 – Calificación de la escuela que le corresponde al lugar

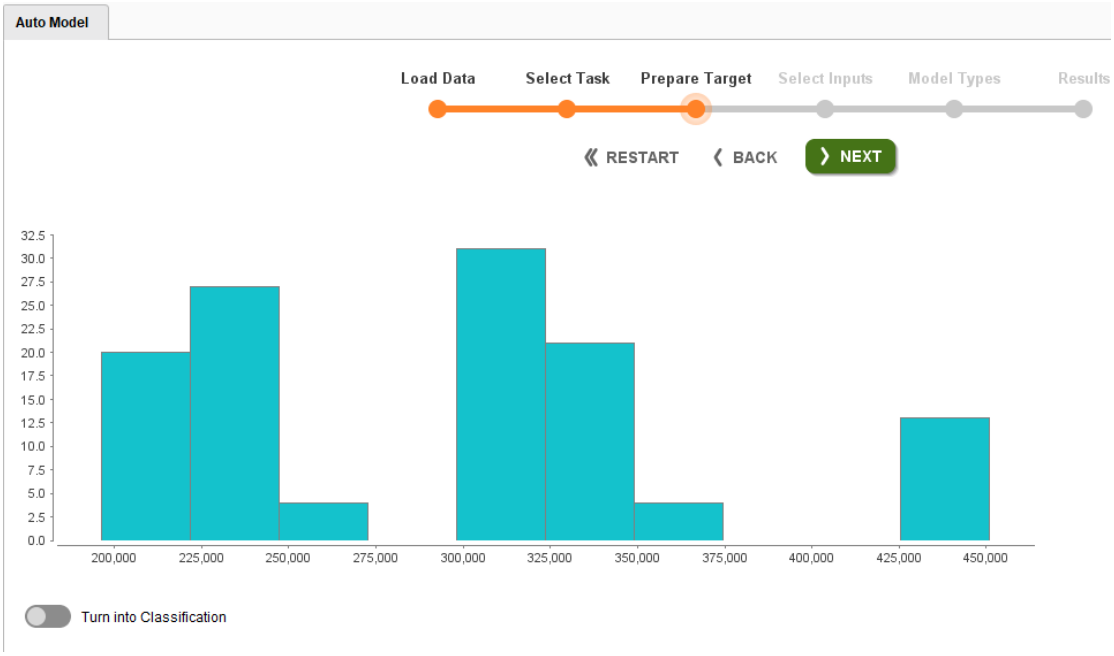
Al ingresar a RapidMiner pulsamos la pestaña que dice “Auto-model” y cargamos desde nuestra computadora el archivo CSV proporcionado en clase.

Una vez hecho aparecerá la siguiente pantalla, en ella damos click en la columna que se busca predecir, en este caso será el precio de cada casa. Luego se da clic en el botón “Next”.



house_sqft Number	num_of_bedrooms Number	num_of_bathrooms Number	year_built Number	tax_assessed_value Number	last_sold_price Number
1770	3	2	1990	195000	196358
1770	3	2	1990	195000	197715
1770	3	2	1990	195000	197816
1772	3	2	1990	200000	198011
1850	3	2.500	1990	200000	200530
1850	3	2.500	1990	200000	201805
1850	3	2.500	1990	205000	206175
1850	3	2.500	1990	205000	207027

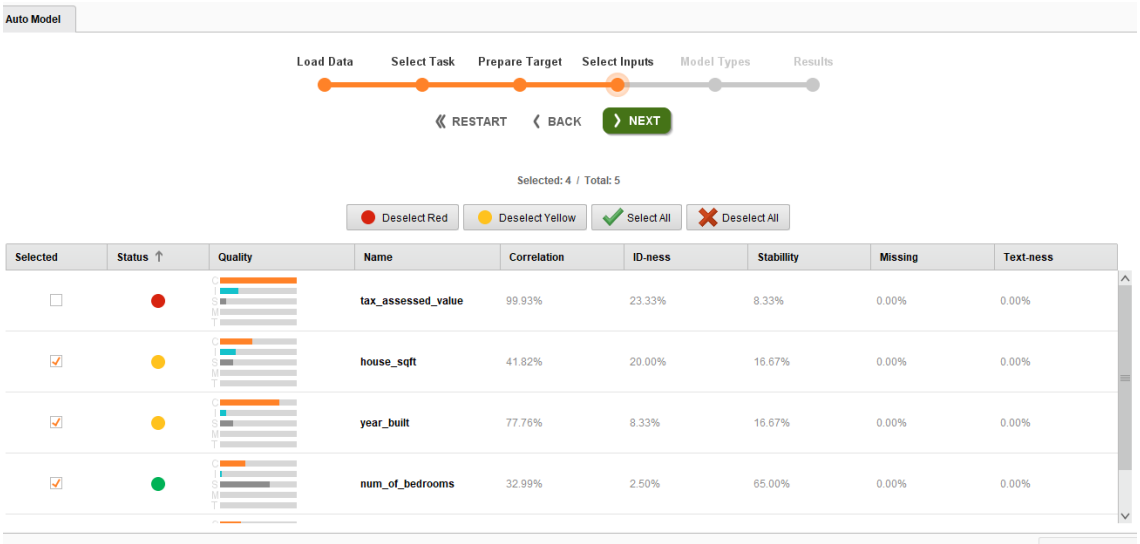
Luego se mostrará un histograma con las frecuencias de los precios de las casas:



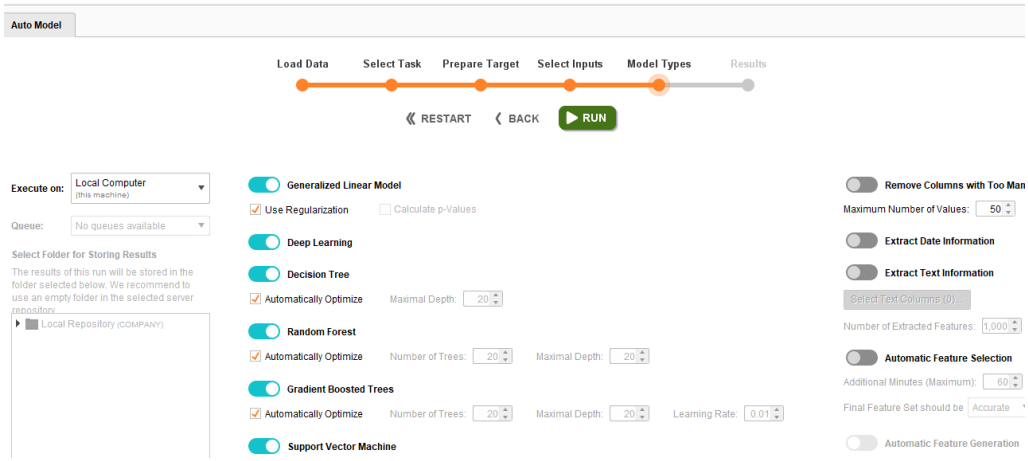
Pulsamos de nuevo en “Next” y se mostrará las variables (columnas) que se utilizarán para identificar el patrón:

- El color rojo, indica que no aportarán nada significativo en la búsqueda del patrón.
- El color verde, es probable la distinción de valores para formar un patrón.
- El color amarillo, indica que la herramienta no está segura si aportarán o no al patrón.

En este caso las variables marcadas con estado amarillo si son importantes: la superficie de la casa y el año en que fueron construidas, esto se demostrará más adelante en el informe.



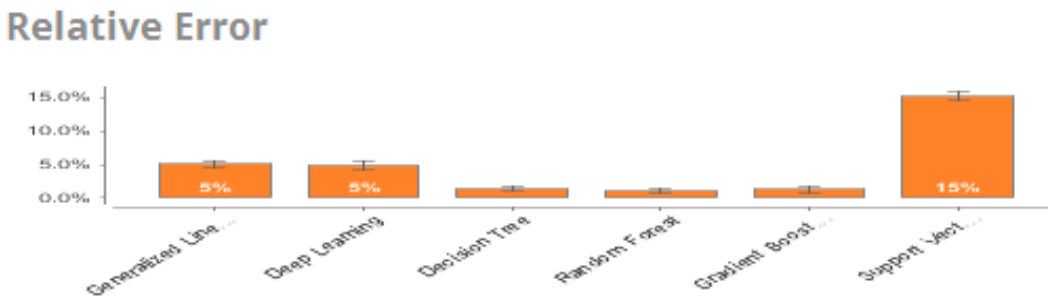
Pulsamos siguiente y luego aparecerá una pantalla con todos los modelos que podrían acoplarse a este conjunto de datos, se seleccionan todos para poder compararlos.



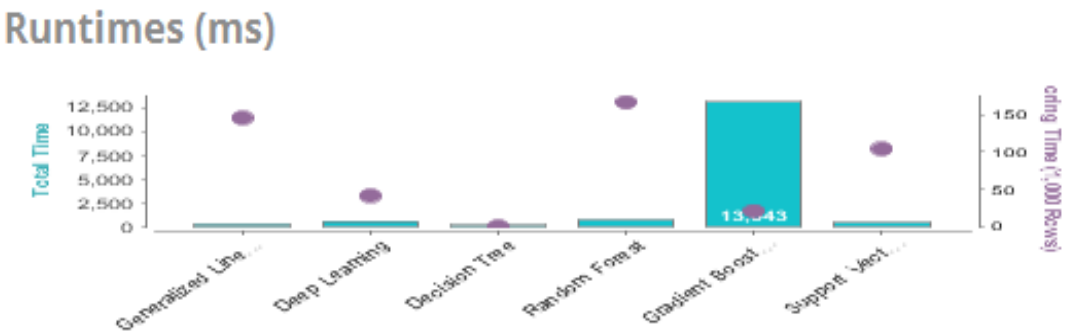
Al pulsar “Run” los modelos se empezarán a entrenar y luego se puntuarán con un score.

Después de varios minutos (dependiendo del tipo de procesador que se utilice) se obtendrá estos resultados de los modelos.

Se destaca que el error de “Support Vector Machine” es bastante alto por lo que es el peor algoritmo en el problema. Y aparentemente: Decision Tree, Random Forest y Gradient Boost son los mejores algoritmos.



En términos de tiempo: Gradient Boost fue el más demoroso y Decision Tree el más rápido.



Cuadro resumen de errores relativos, desviaciones estándar, tiempo de entrenamiento, tiempo de scoring y tiempo total.

Model	Relative Error	Standard Deviation	Gains	Total Time	Training Time (1,000 Rows)
Generalized Linear Model	5.1%	± 0.4%	?	254 ms	425 ms
Deep Learning	4.9%	± 0.7%	?	596 ms	2 s
Decision Tree	1.4%	± 0.3%	?	136 ms	8 ms

Model	Relative Error	Standard Deviation	Gains	Total Time	Training Time (1,000 Rows)
Random Forest	1.1%	± 0.2%	?	697 ms	100 ms
Gradient Boosted Trees	1.3%	± 0.5%	?	13 s	3 s
Support Vector Machine	15.1%	± 0.6%	?	406 ms	92 ms

2. Identificar los algoritmos con mejor resultados

El modelo con mayor velocidad al otorgar un score y en general es el Árbol de decisiones con solo un 1.4% de error.

Decision Tree	1.4%	± 0.3%	?	136 ms	8 ms

El mejor modelo en cuestión de predicción es “Random Forest” con una menor desviación que el árbol de decisiones pero se demora 5.125 (o mejor dicho 697/136) veces más al entrenar y 12.5 (o mejor dicho 100/8) veces más al otorgar scores.

Random Forest	1.1%	± 0.2%	?	697 ms	100 ms

Predecir 10 distintos escenarios de venta de casas con distintos algoritmos

A continuación, se tomará datos de los precios que dio cada algoritmo para los siguientes escenarios:

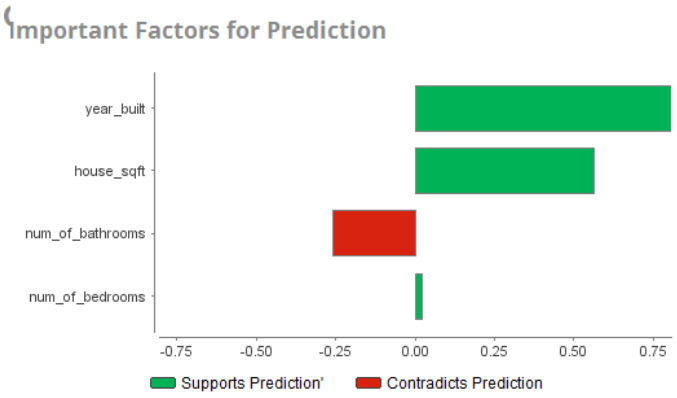
1. Con número promedio de datos.
2. Con bajo número de cuartos
3. Con alto número de cuartos
4. Con bajo número de baños
5. Con alto número de baños
6. Con una superficie pequeña
7. Con una superficie enorme
8. Antigua
9. Moderna
10. Con alta superficie, bajo número de cuartos, pero construida recientemente.

El cuadro que resume los datos proporcionados por los simuladores está en la siguiente tabla. En la sección de "Análisis de resultados" se profundizará con un mapa de calor sobre estos datos recolectados.

	Algoritmo					
Escenario	**G. Linear Model**	**Deep Learning**	**Decision Tree**	**Random Forest**	**Gr. B. Trees**	**S. Vector Machine**
1	338115,03	324001,49	307775,80	334173,73	303783,17	303643,06
2	338115,03	324328,47	307775,80	334173,73	303783,17	299283,64
3	338115,03	329832,86	333419,50	342996,86	312337,48	310760,37
4	362211,07	349464,36	307775,80	334156,44	303783,17	299340,70
5	310537,66	301744,64	307775,80	342698,07	310880,61	311715,98
6	273077,50	260952,60	301303,00	304227,01	306830,08	296460,74
7	390301,90	382543,38	307775,80	331807,49	327689,17	310759,01
8	247685,58	246634,64	230345,00	247992,38	213431,25	295893,75
9	412102,77	422240,04	435287,33	436980,42	437775,92	312007,47
10	470764,51	437065,87	448046,00	420910,12	441310,26	309162,92

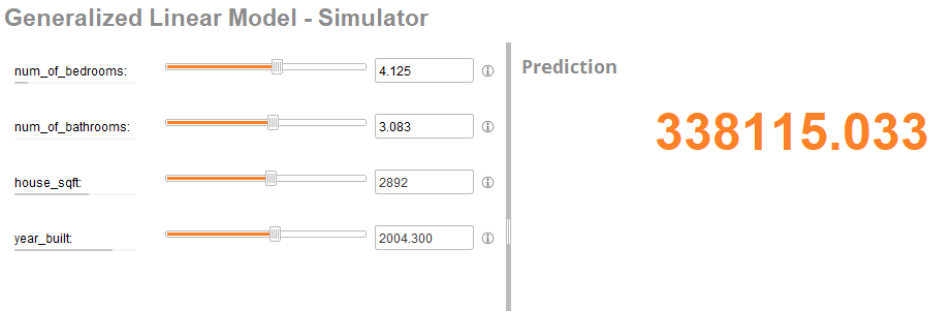
A. Algoritmo: Generalized Linear Model

Factores:

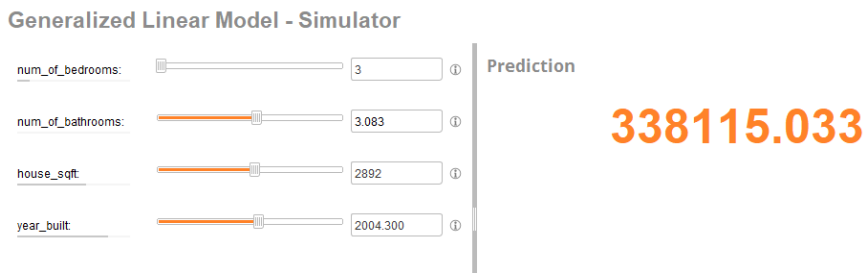


Escenarios:

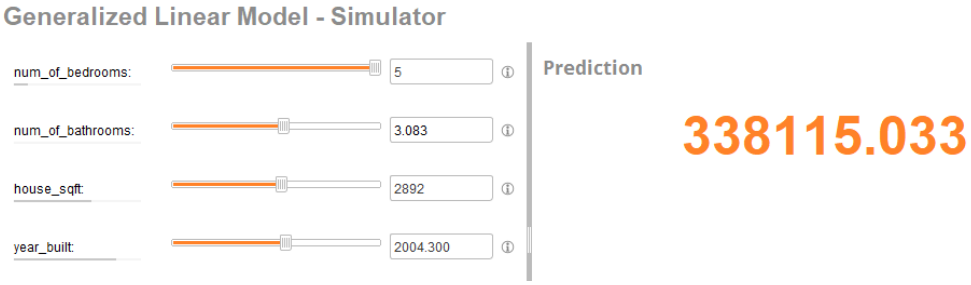
1. Con número promedio de datos.



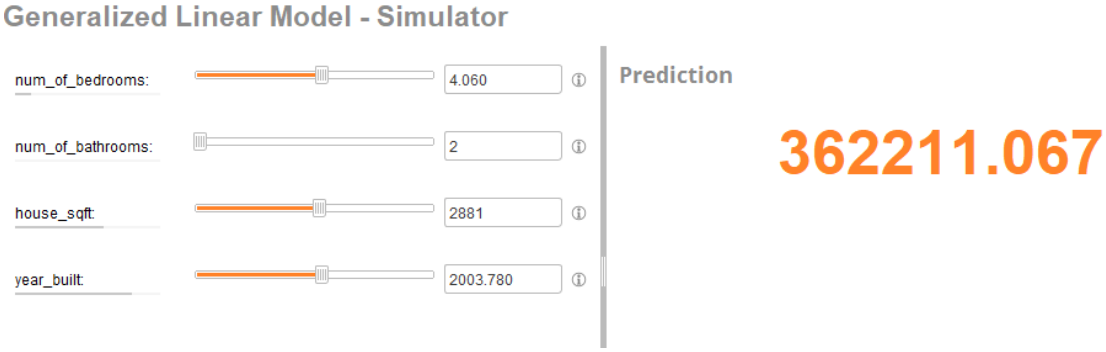
2. Con bajo número de cuartos



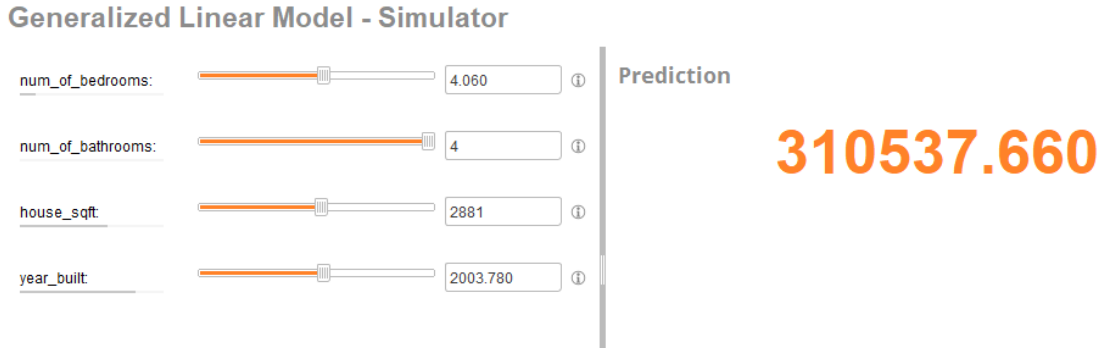
3. Con alto número de cuartos



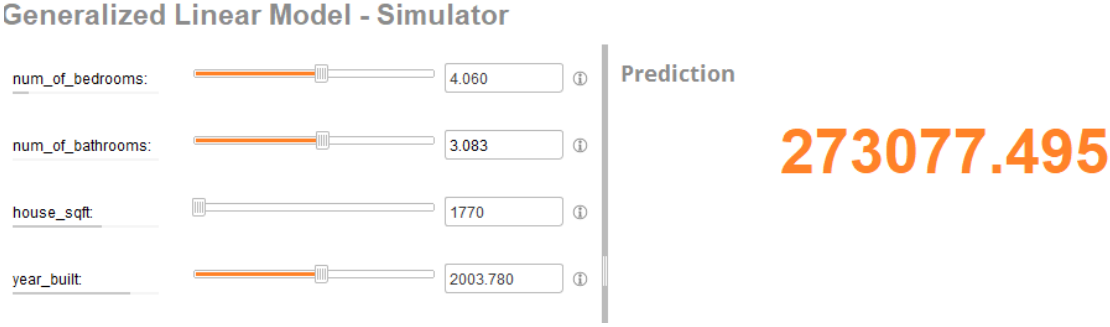
4. Con bajo número de baños



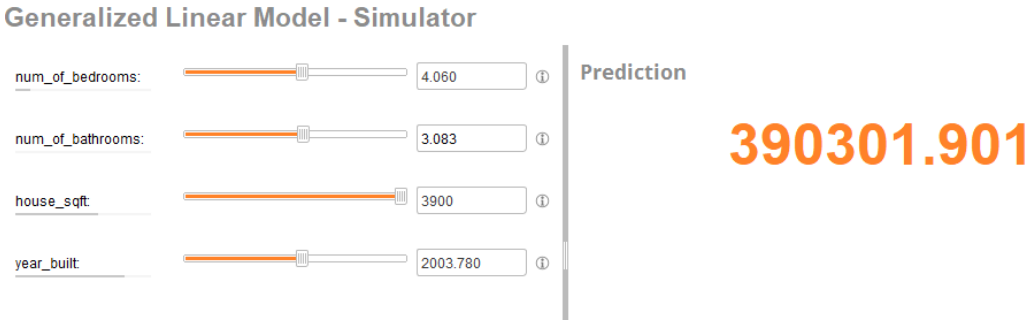
5. Con alto número de baños



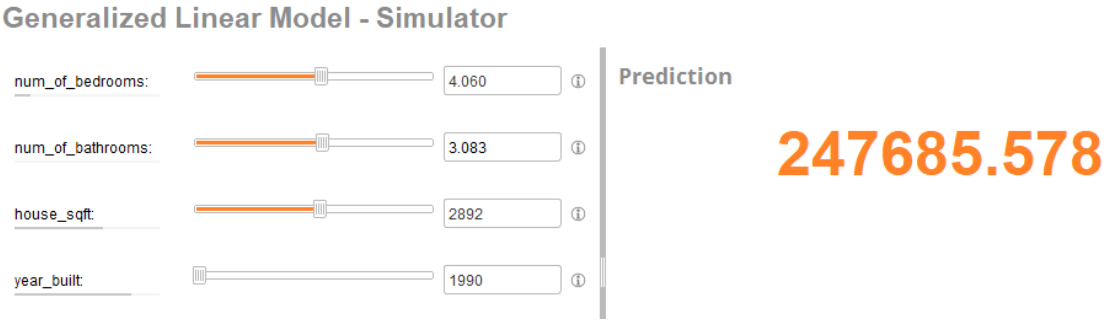
6. Con una superficie pequeña



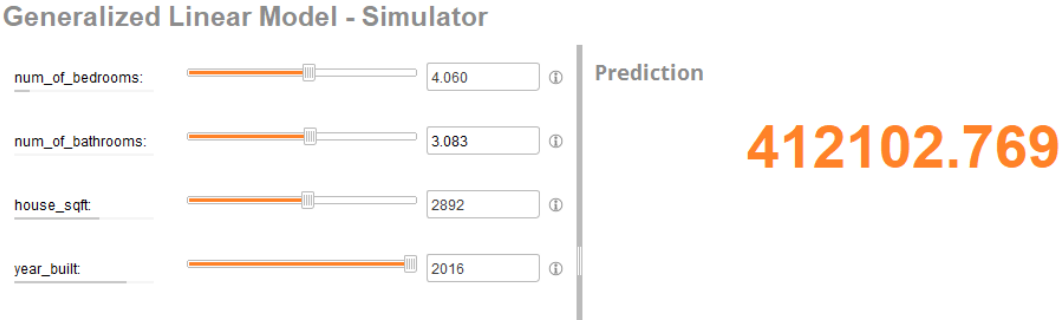
7. Con una superficie enorme



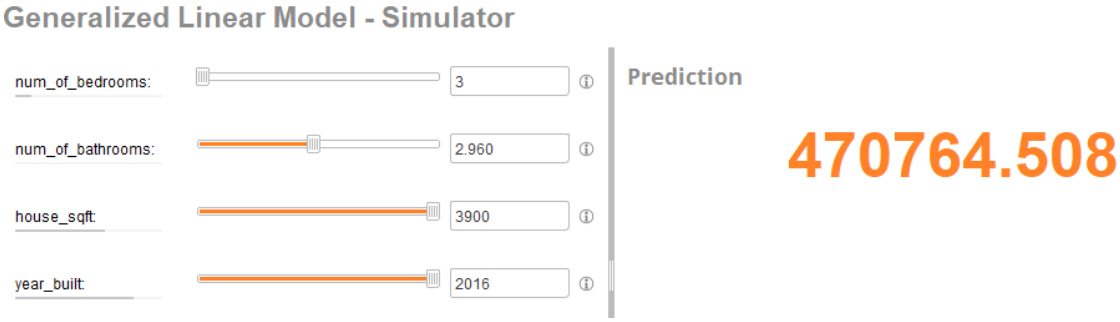
8. Casa antigua



9. Casa moderna

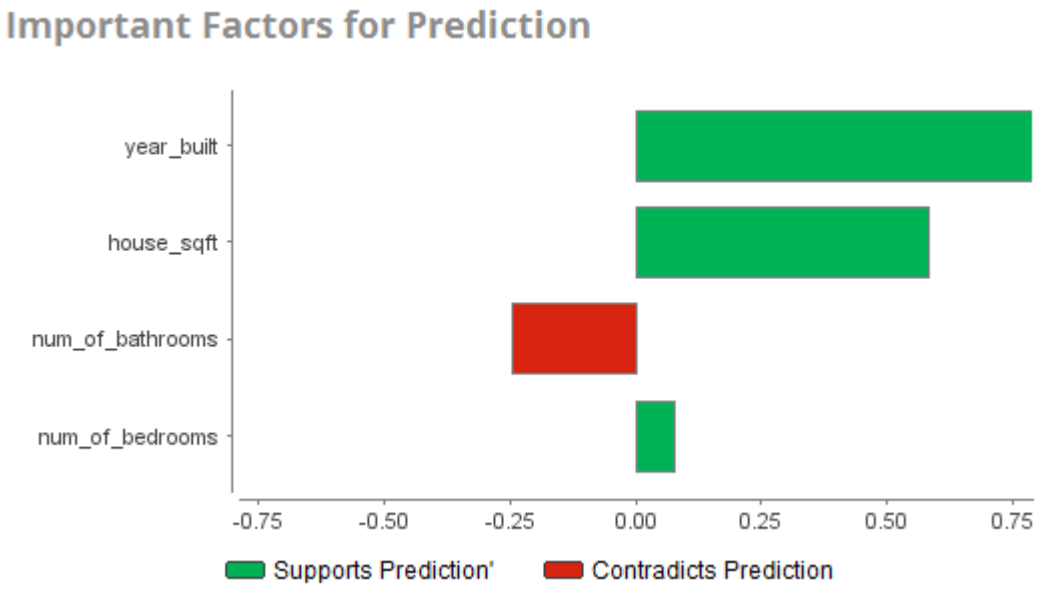


10. Con alta superficie, bajo número de cuartos pero construida recientemente



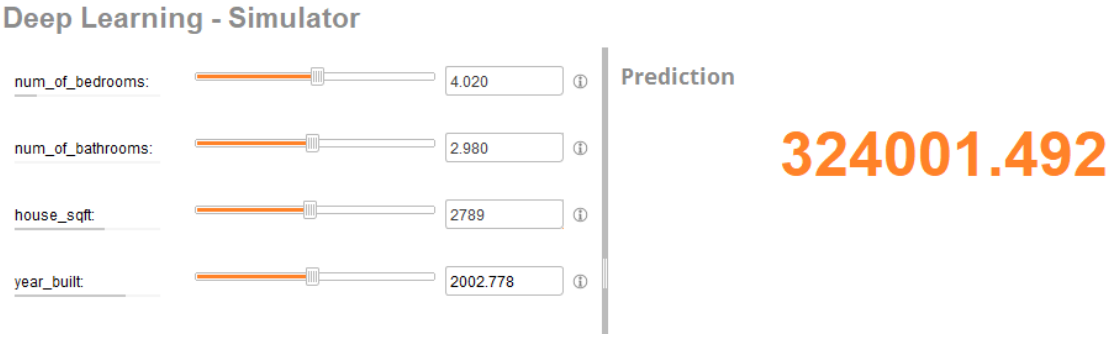
B. Algoritmo: Deep Learning

Factores:

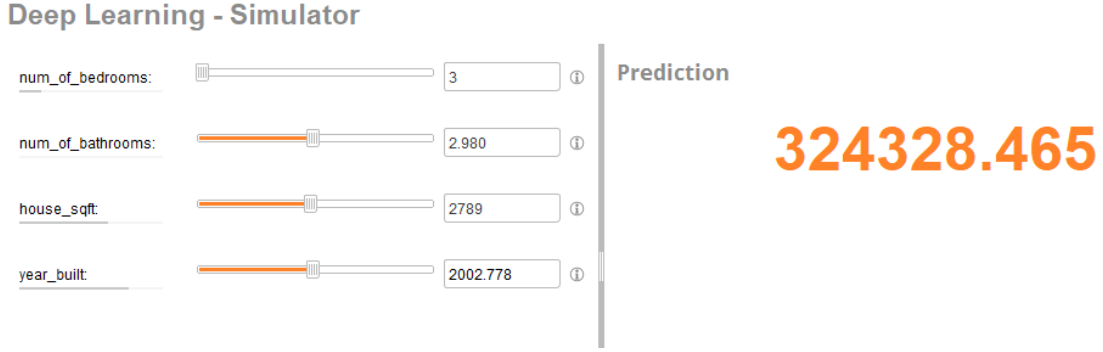


Escenarios:

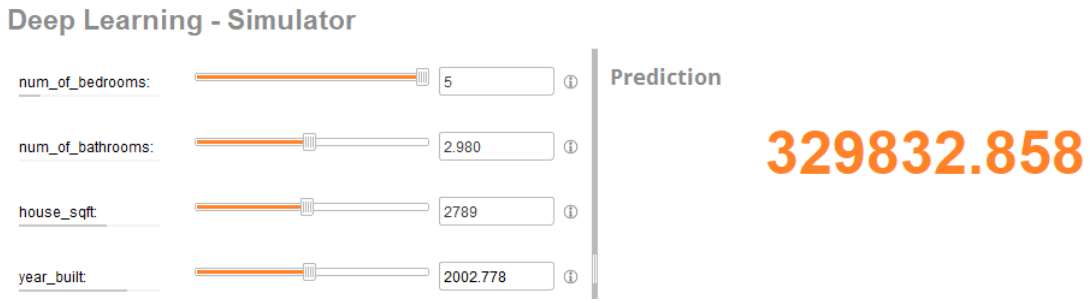
1. Con número promedio de datos



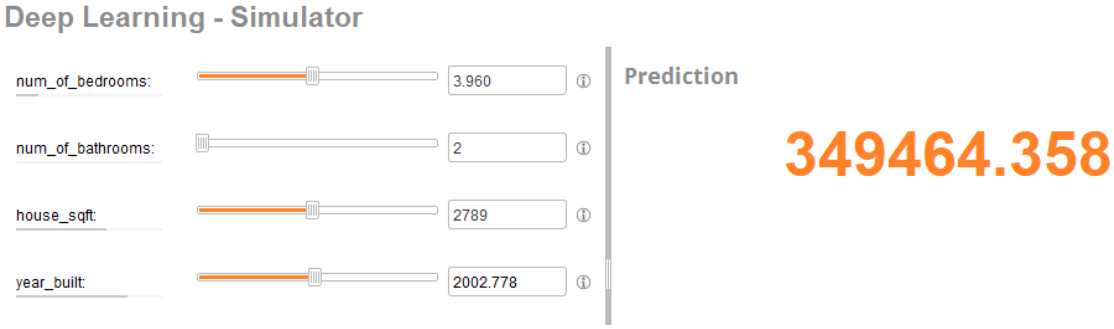
2. Con bajo número de cuartos



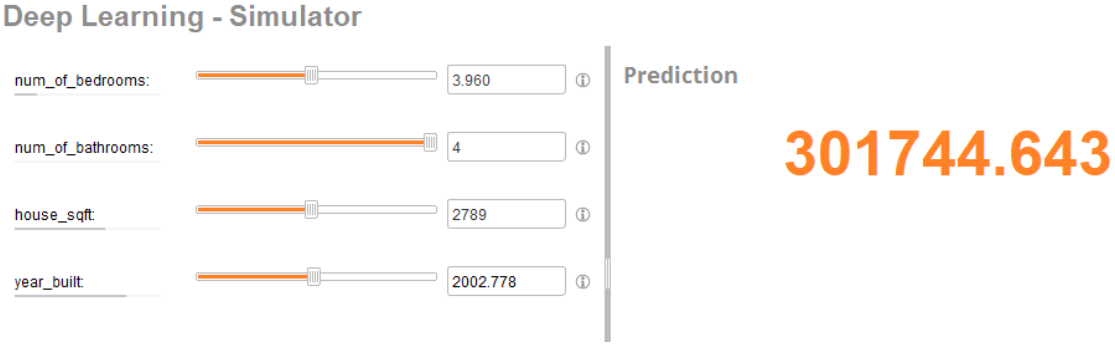
3. Con alto número de cuartos



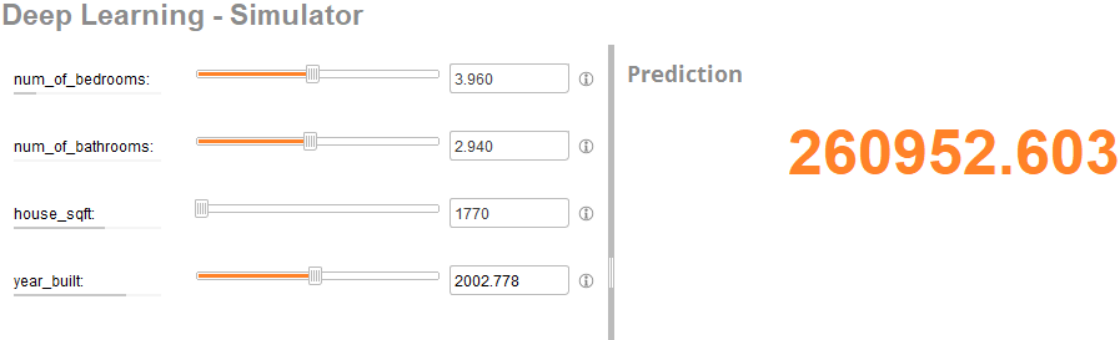
4. Con bajo número de baños



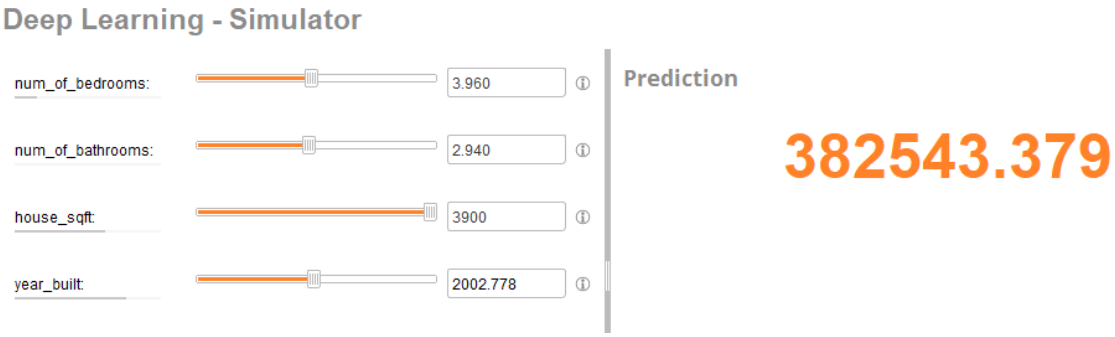
5. Con alto número de baños



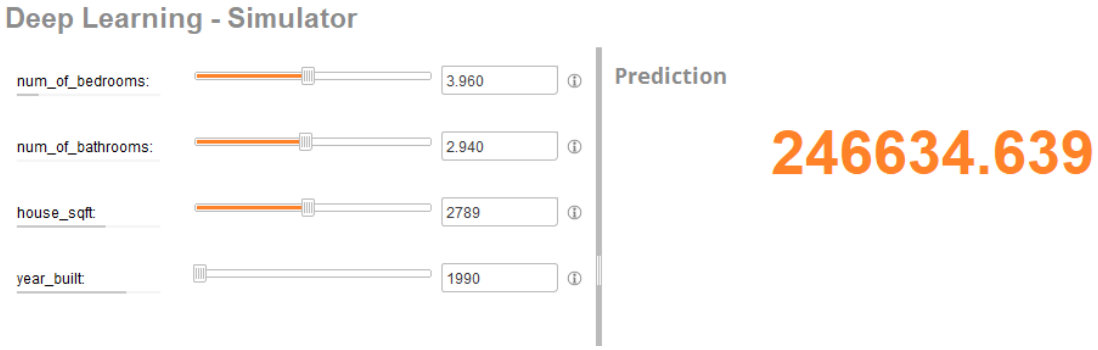
6. Con poca superficie



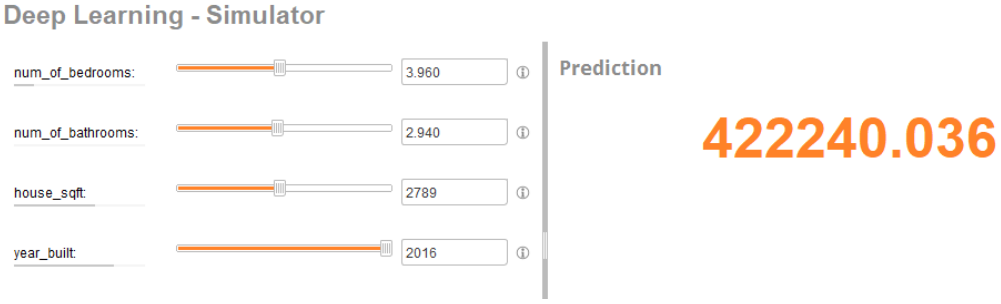
7. Con gran superficie



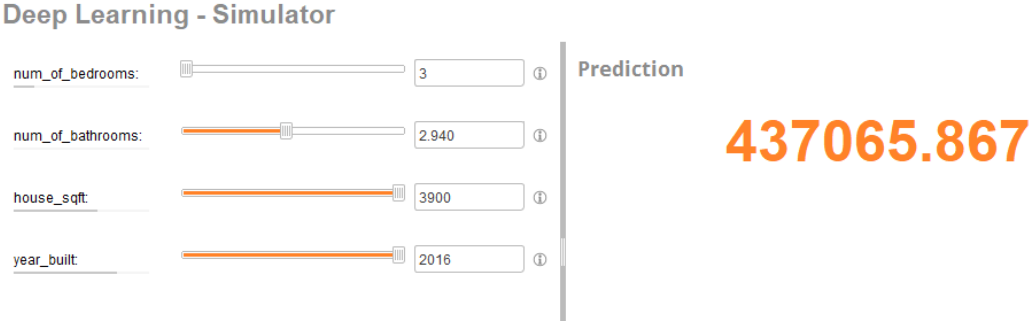
8. Casa antigua



9. Casa moderna

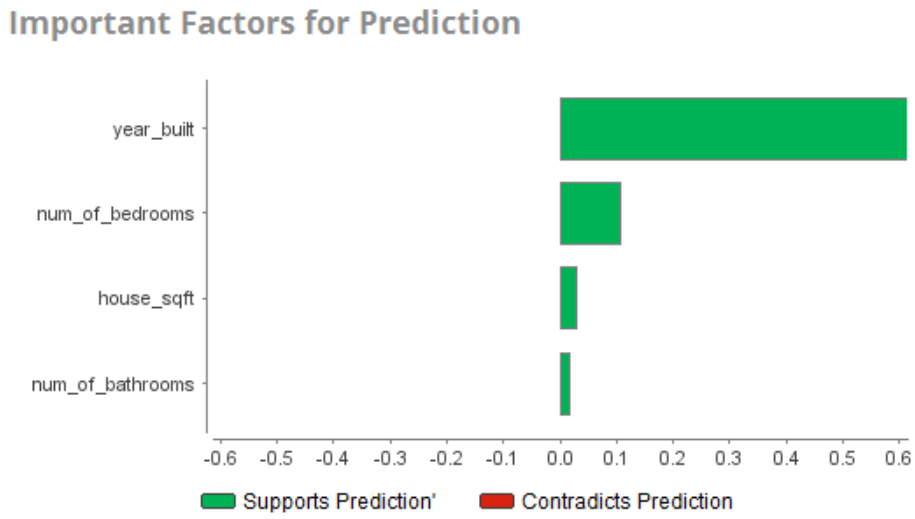


10. Casa con pocos cuartos, construida actualmente y con gran superficie



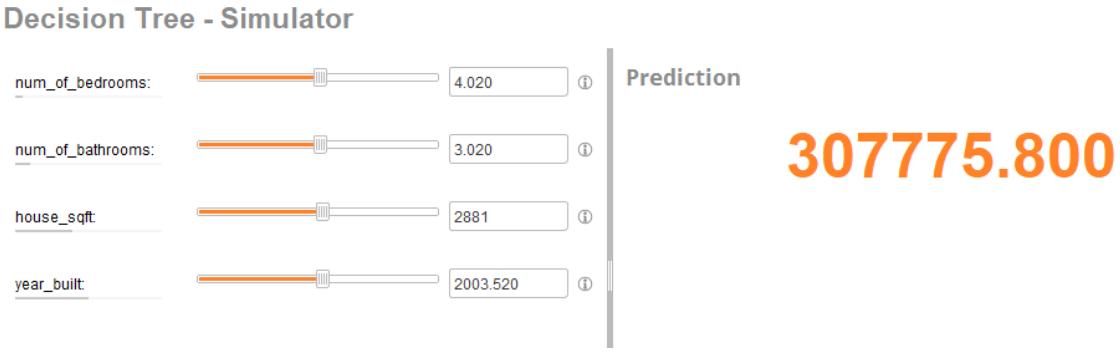
C. Algoritmo: Decision Tree

Factores:

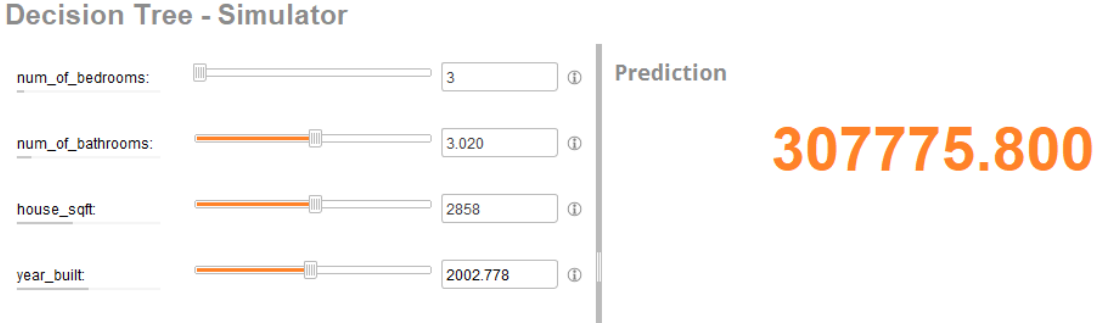


Escenarios:

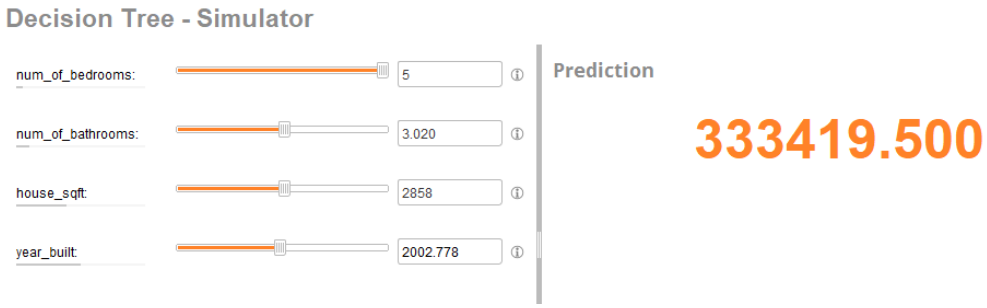
1. Con número promedio de datos



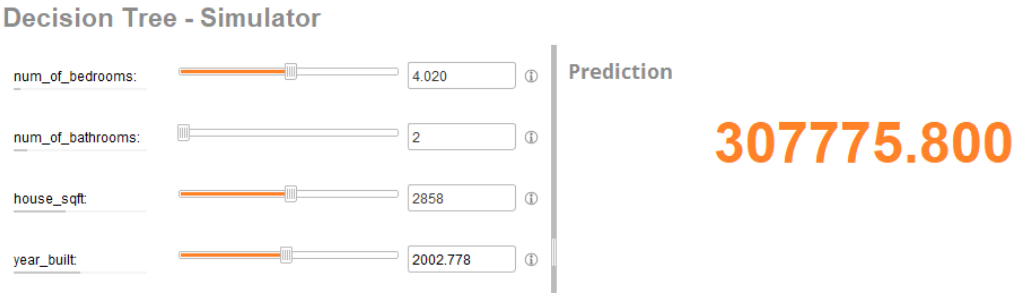
2. Con bajo número de cuartos



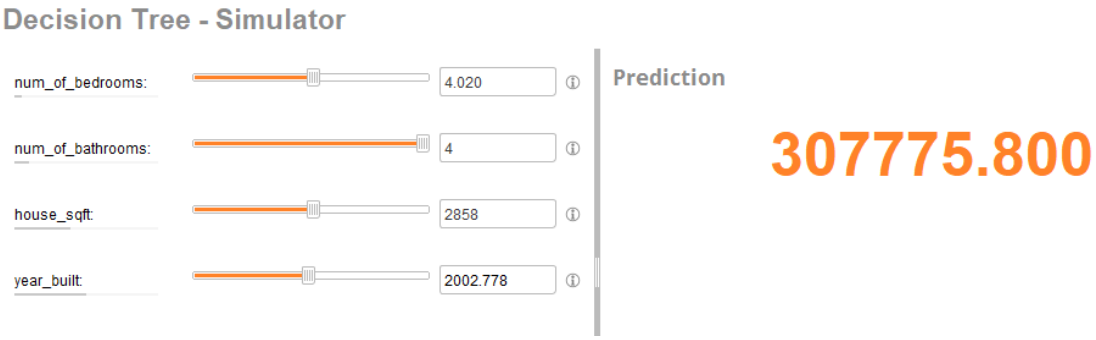
3. Con alto número de cuartos



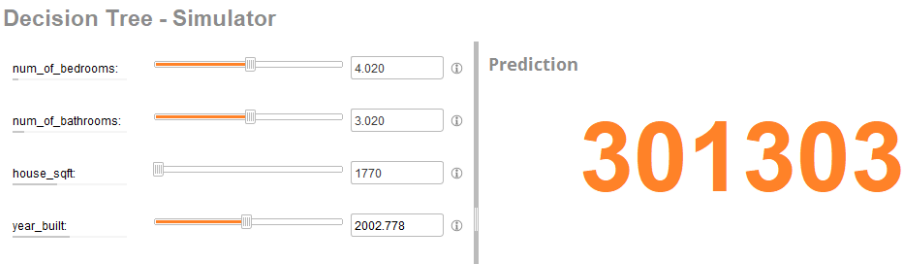
4. Con bajo número de baños



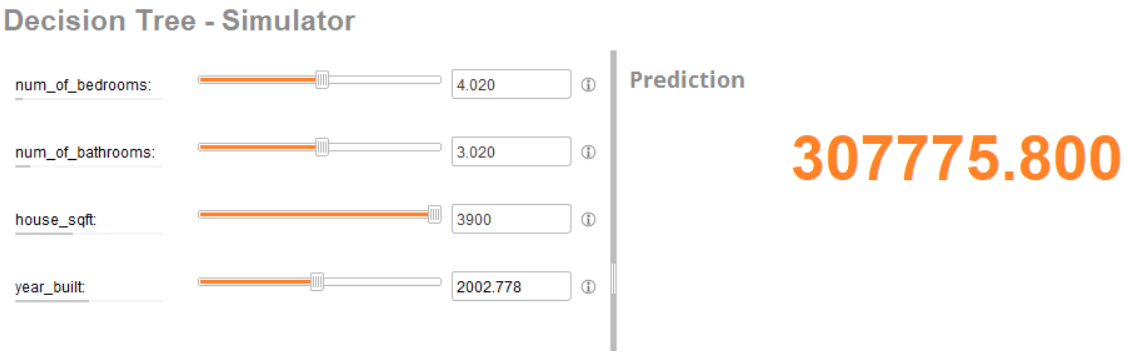
5. Con alto número de baños



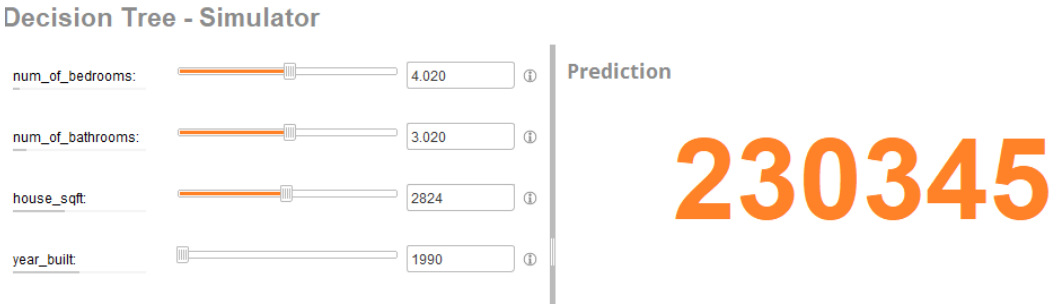
6. Con poca superficie



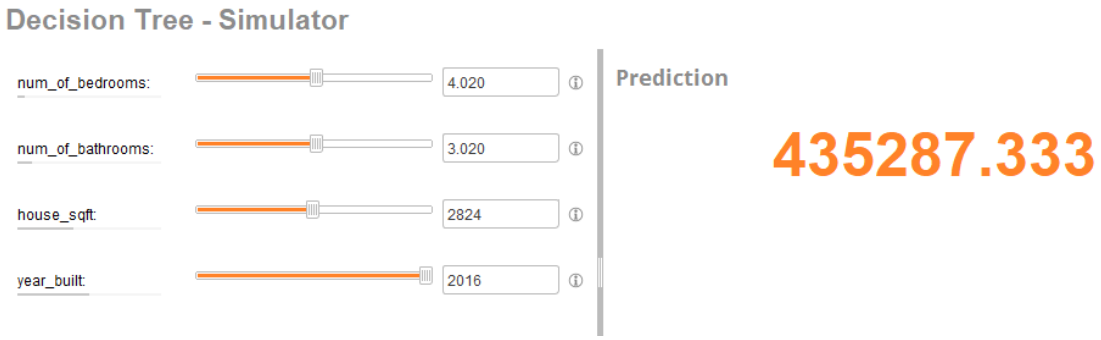
7. Con gran superficie



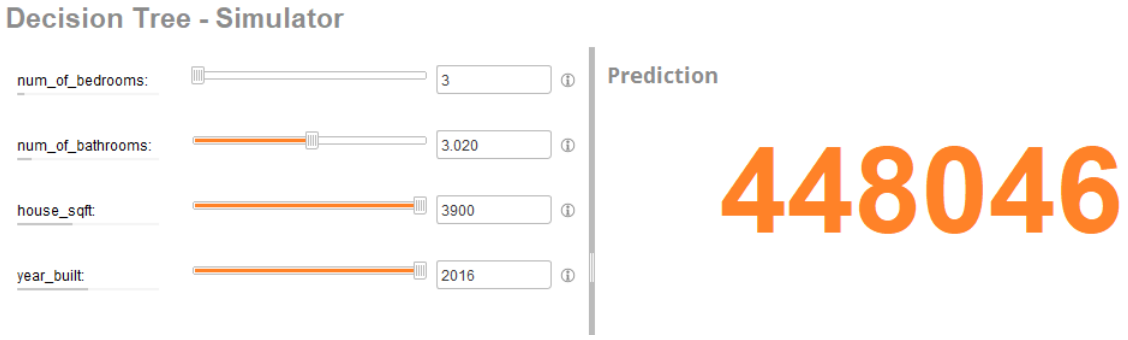
8. Casa antigua



9. Casa moderna

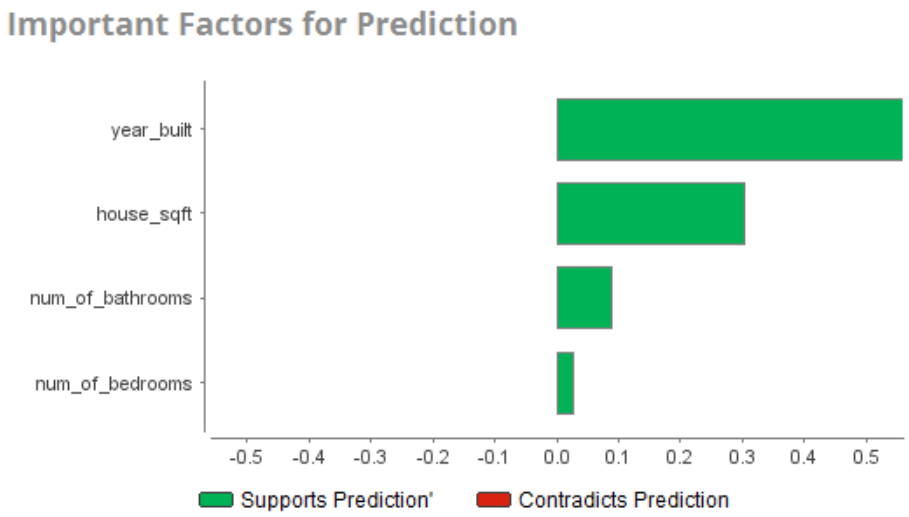


10. Casa con pocos cuartos, construida actualmente y con gran superficie



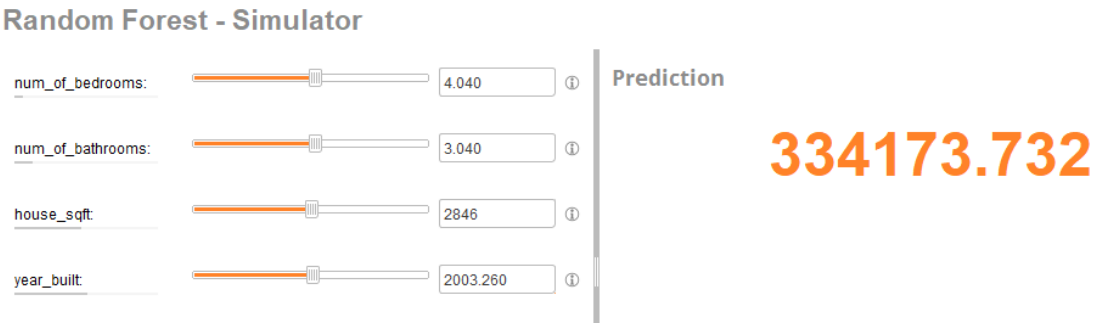
D. Algoritmo: Random Forest

Factores:

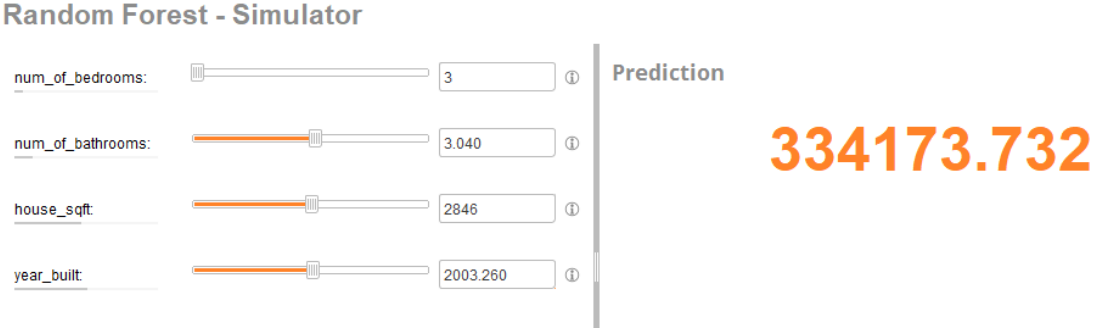


Escenarios:

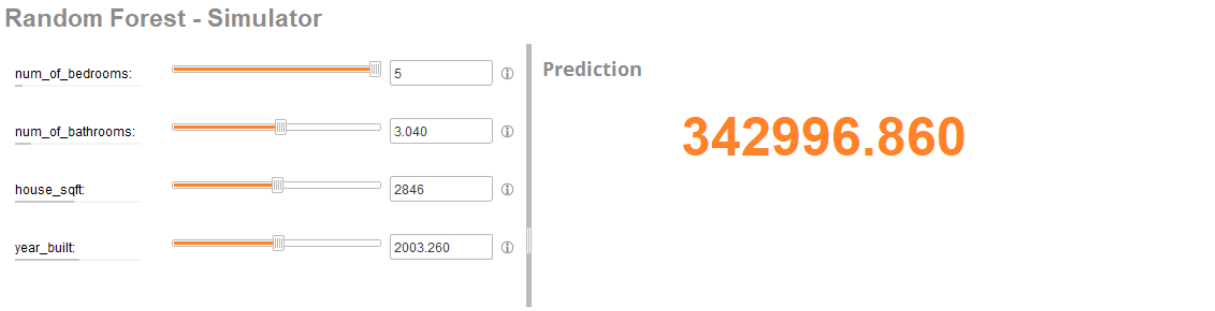
1. Con número promedio de datos



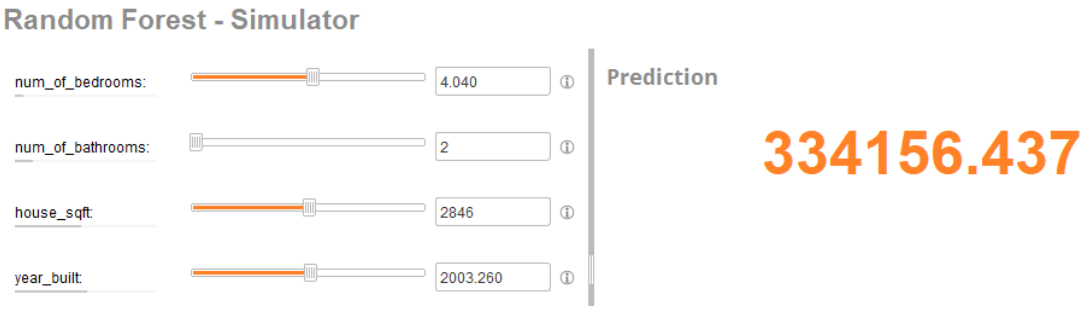
2. Con bajo número de cuartos



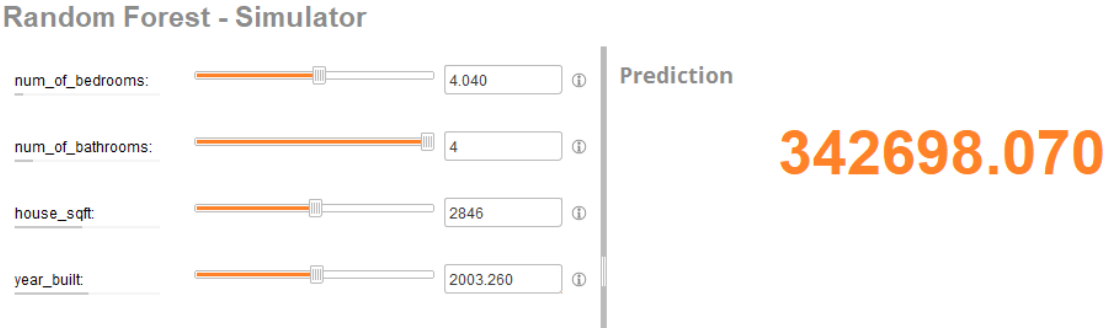
3. Con alto número de cuartos



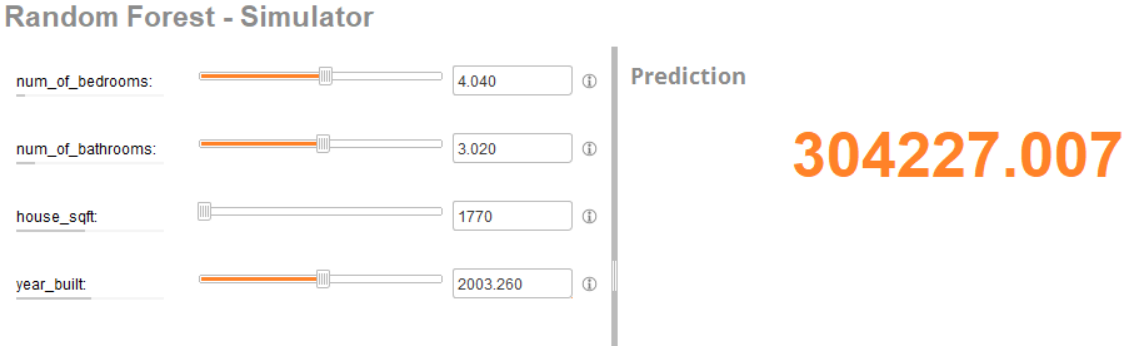
4. Con bajo número de baños



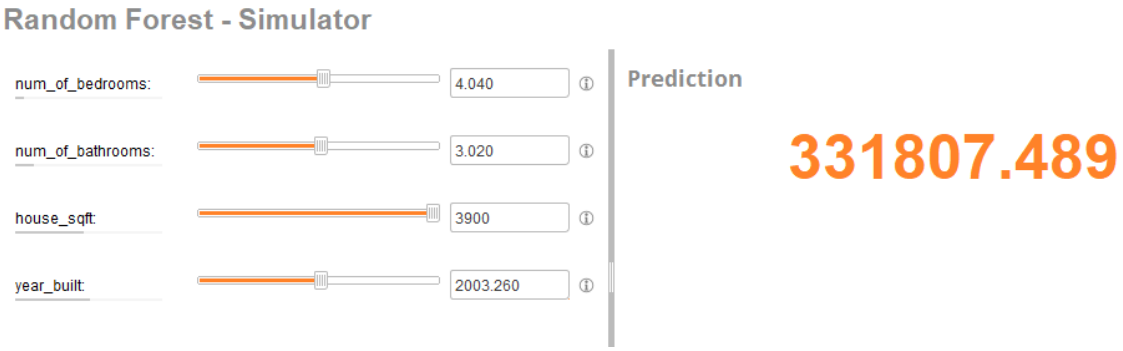
5. Con alto número de baños



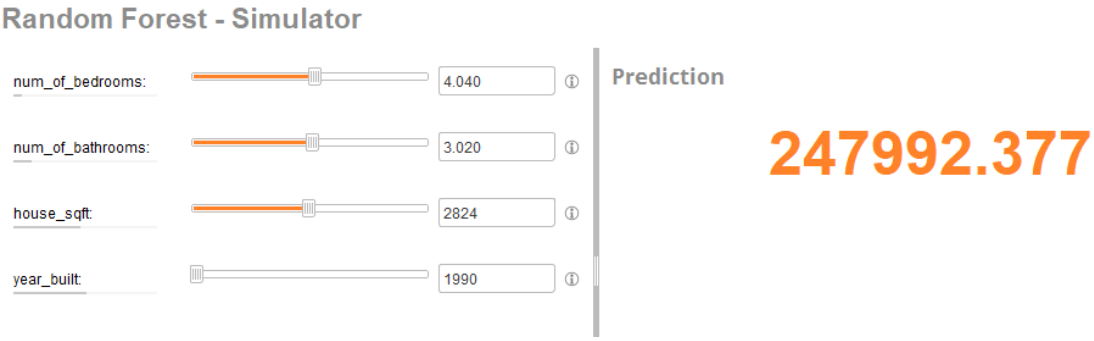
6. Con poca superficie



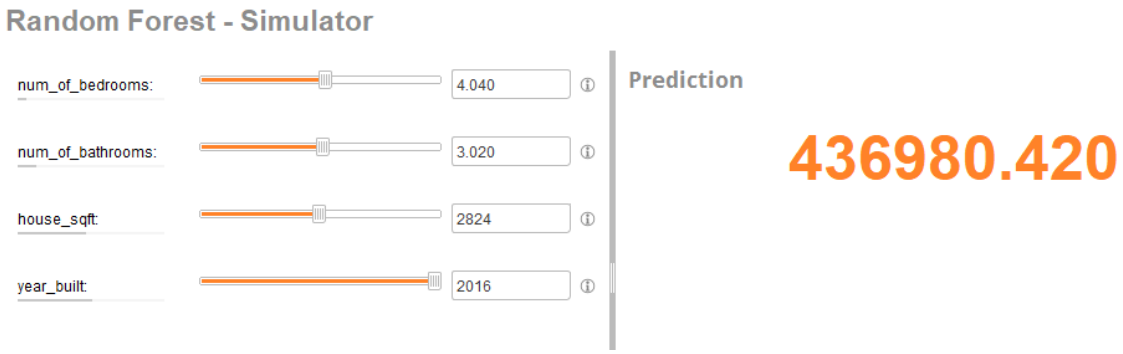
7. Con gran superficie



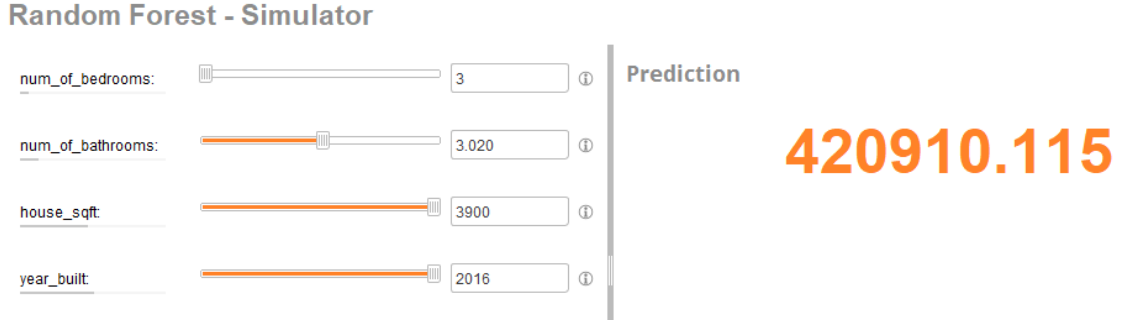
8. Casa antigua



9. Casa moderna

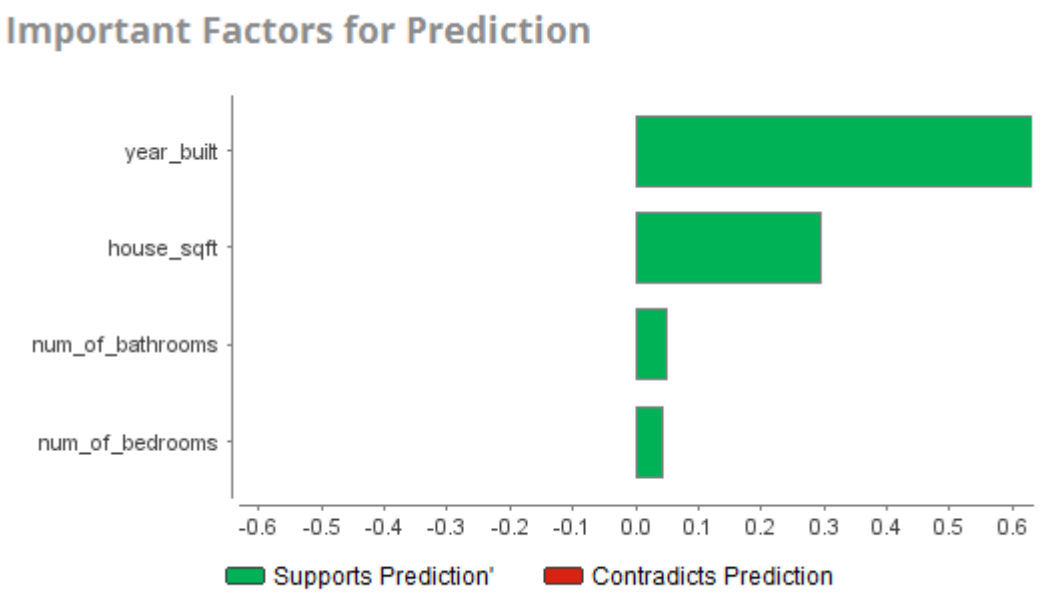


10. Casa con pocos cuartos, construida actualmente y con gran superficie



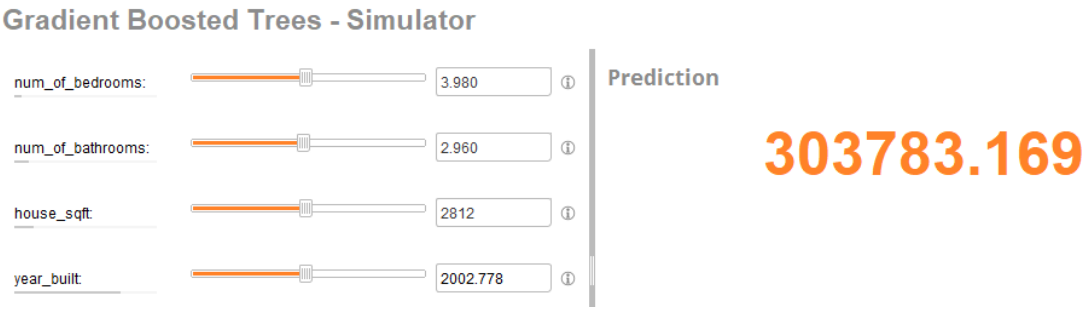
E. Algoritmo: Gradient Boosted Trees

Factores:

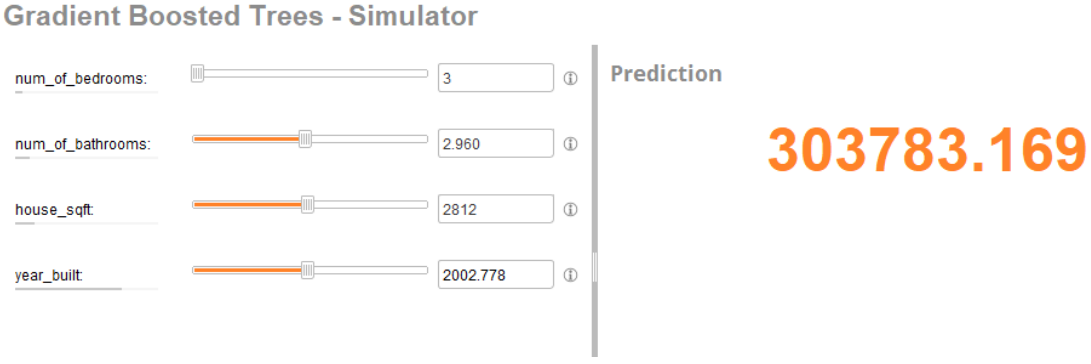


Escenarios:

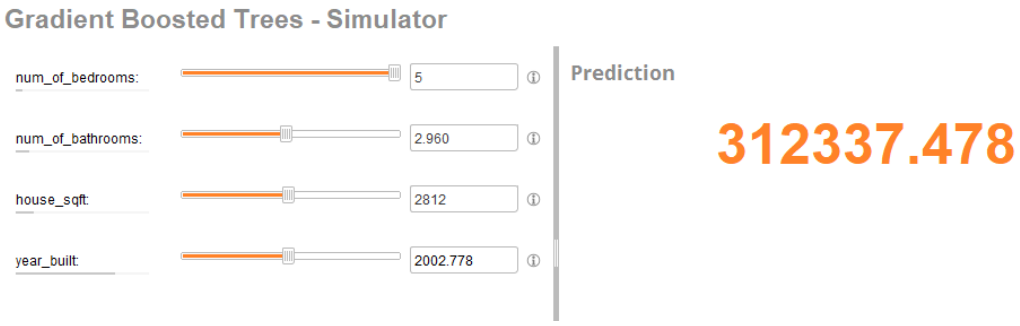
1. Con número promedio de datos



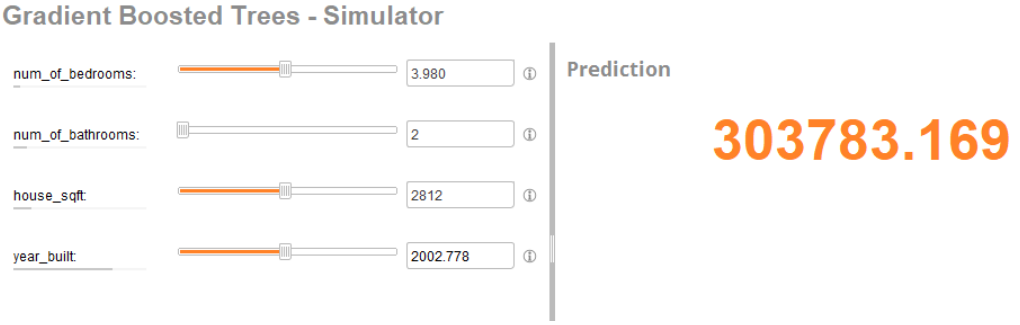
2. Con bajo número de cuartos



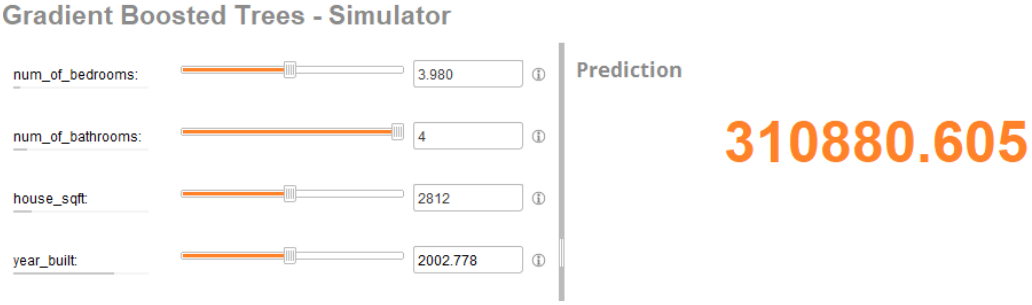
3. Con alto número de cuartos



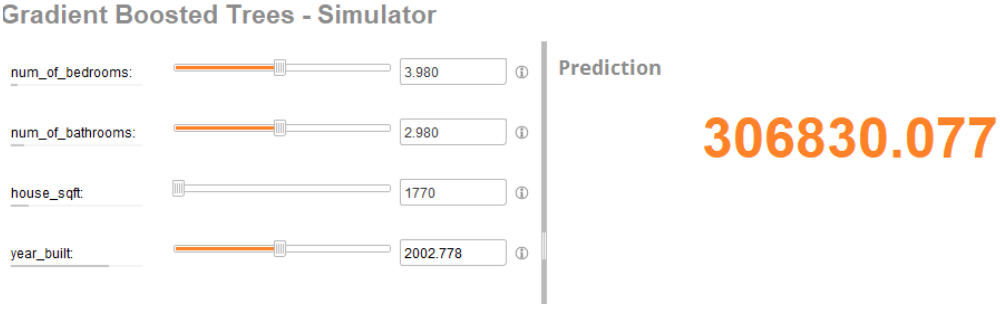
4. Con bajo número de baños



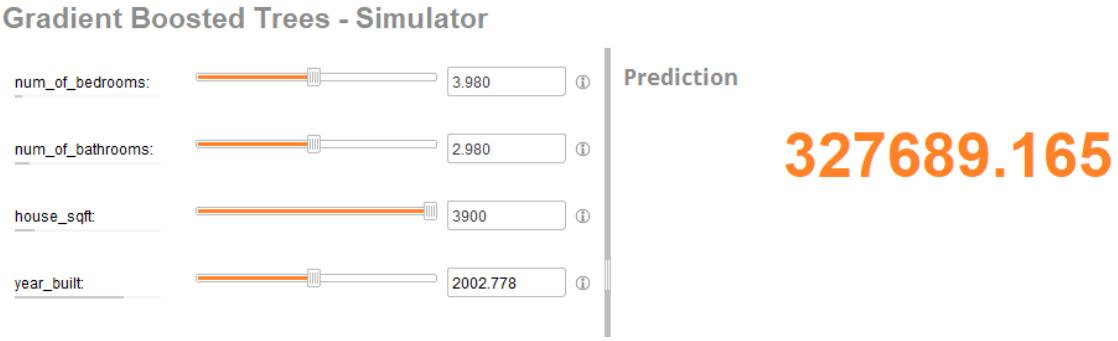
5. Con alto número de baños



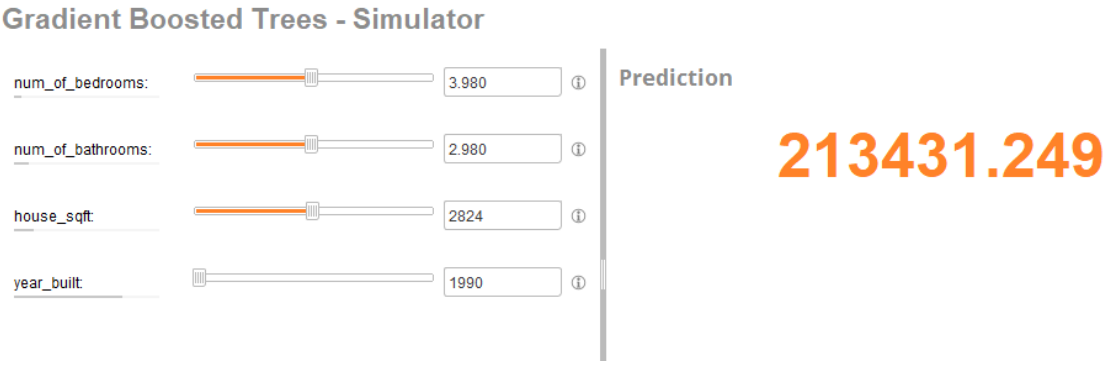
6. Con poca superficie



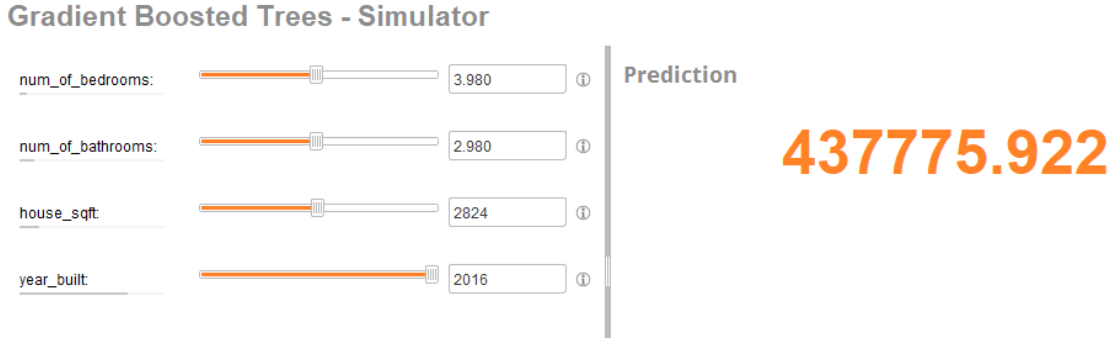
7. Con gran superficie



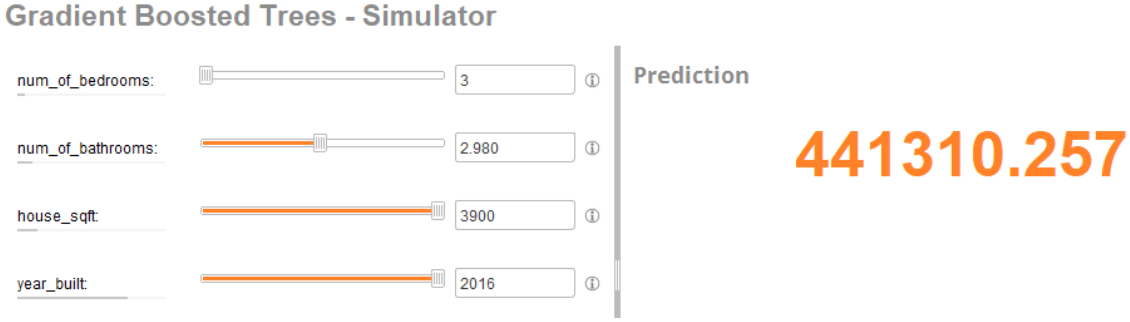
8. Casa antigua



9. Casa moderna

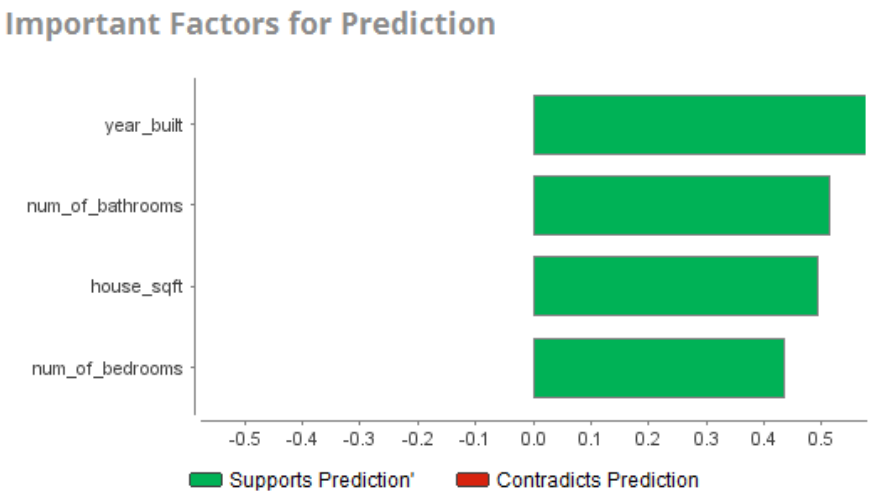


10. Casa con pocos cuartos, construida actualmente y con gran superficie



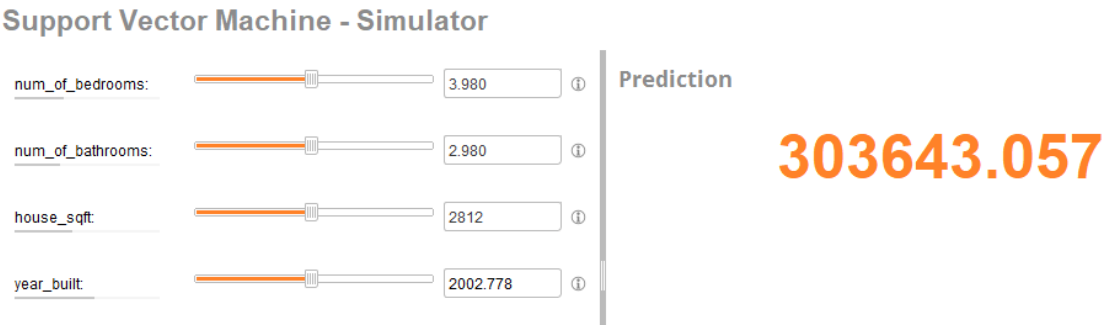
F. Algoritmo: Support Vector Machine

Factores:

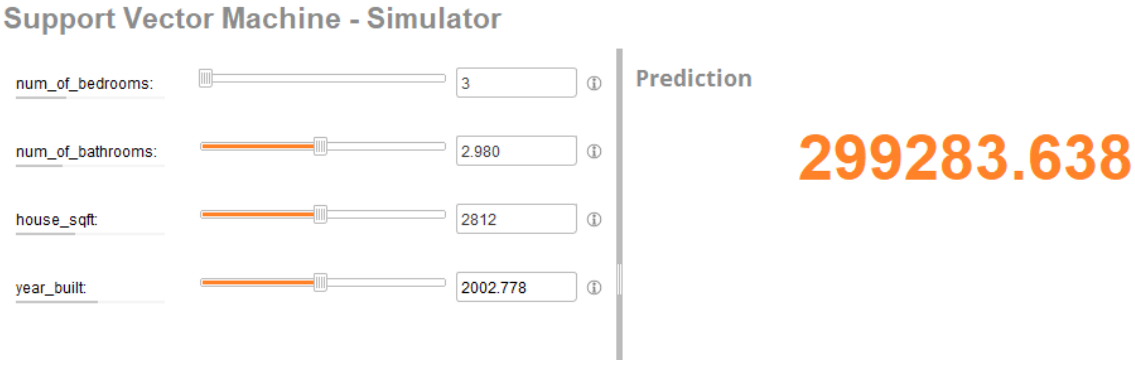


Escenarios:

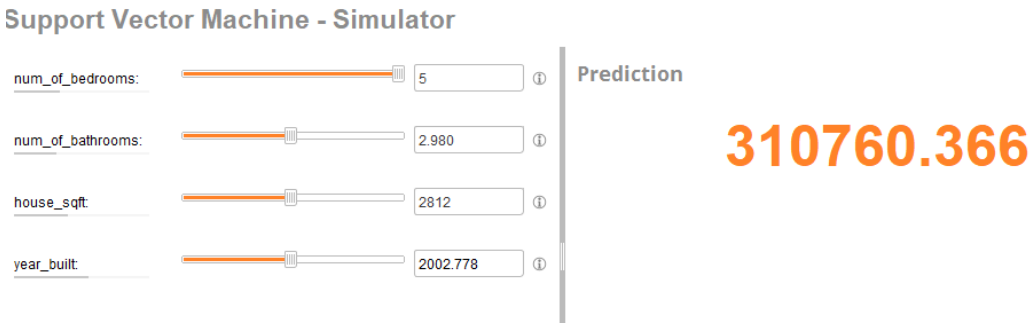
1. Con número promedio de datos



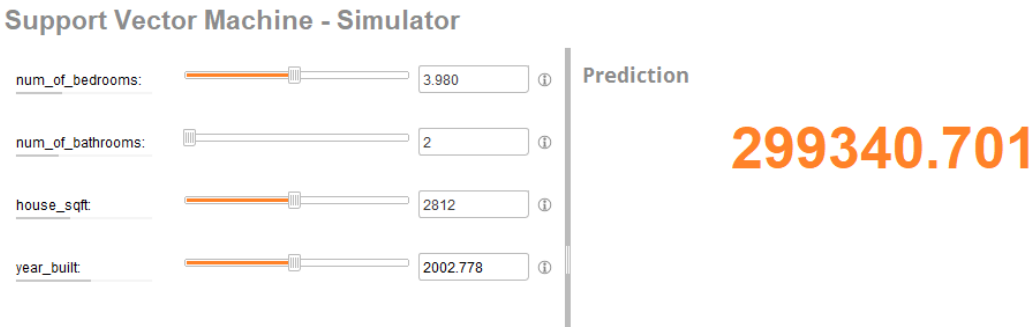
2. Con bajo número de cuartos



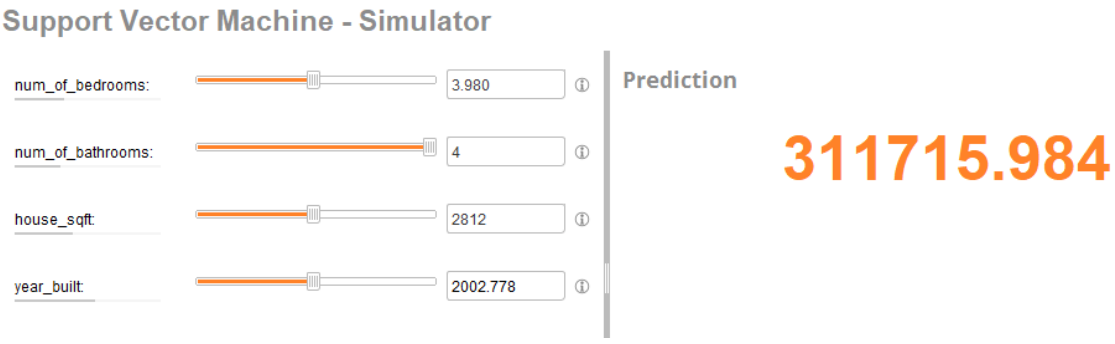
3. Con alto número de cuartos



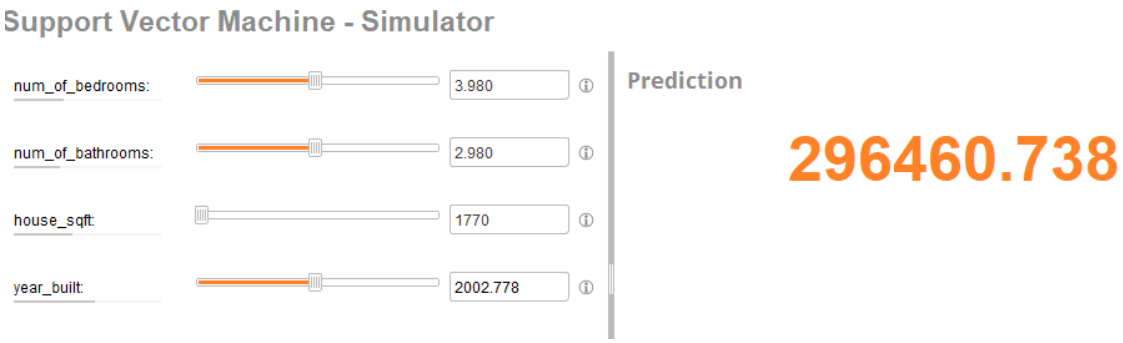
4. Con bajo número de baños



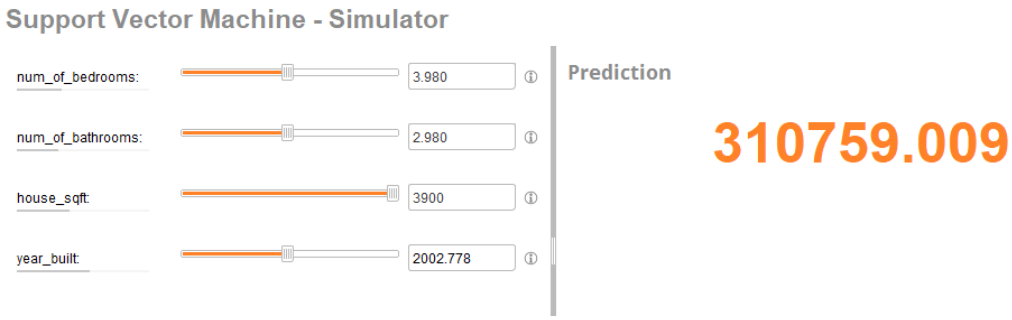
5. Con alto número de baños



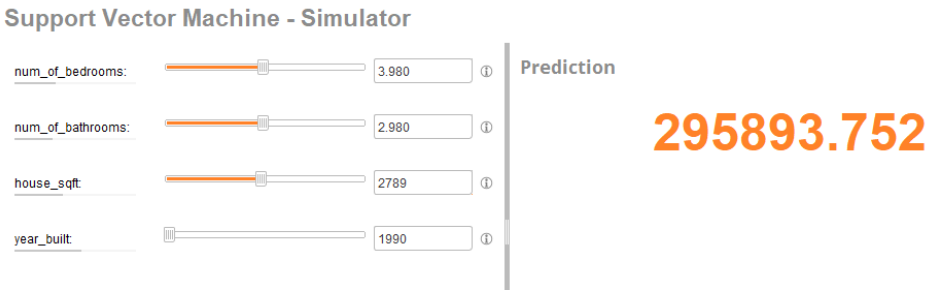
6. Con poca superficie



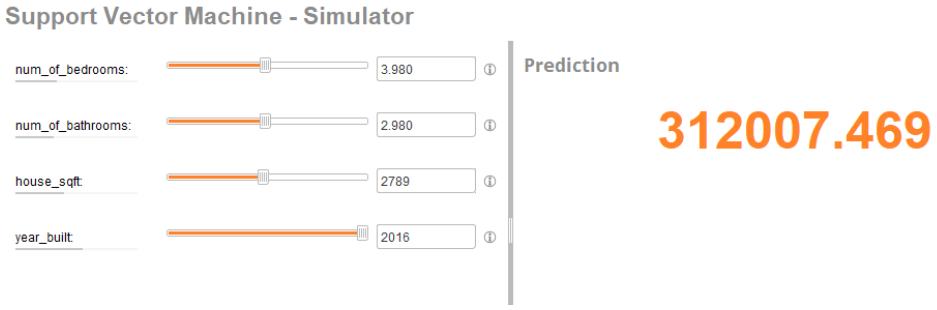
7. Con gran superficie



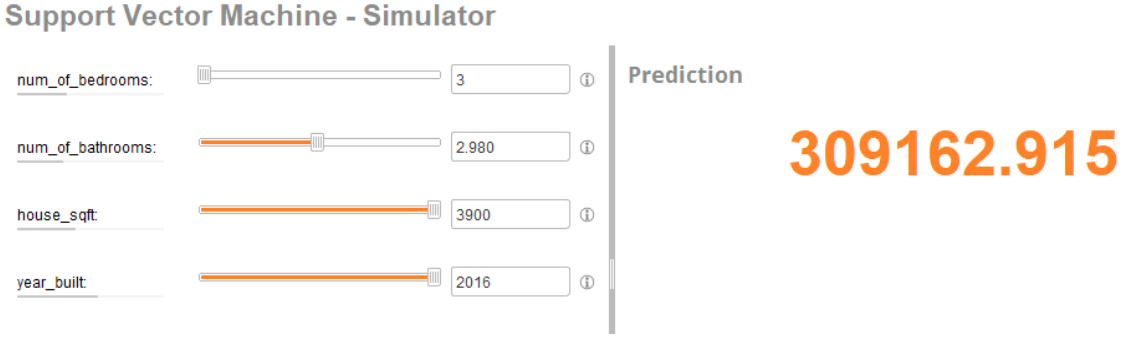
8. Casa antigua



9. Casa moderna



10. Casa con pocos cuartos, construida actualmente y con gran superficie



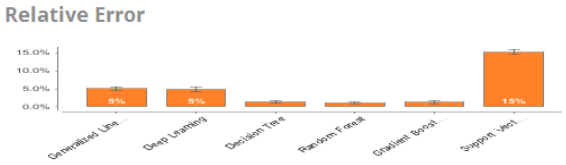
Análisis de resultados:

Generando un mapa de calor sobre cada conjunto de escenarios se establece el verde como el menor valor y el rojo como el mayor valor.

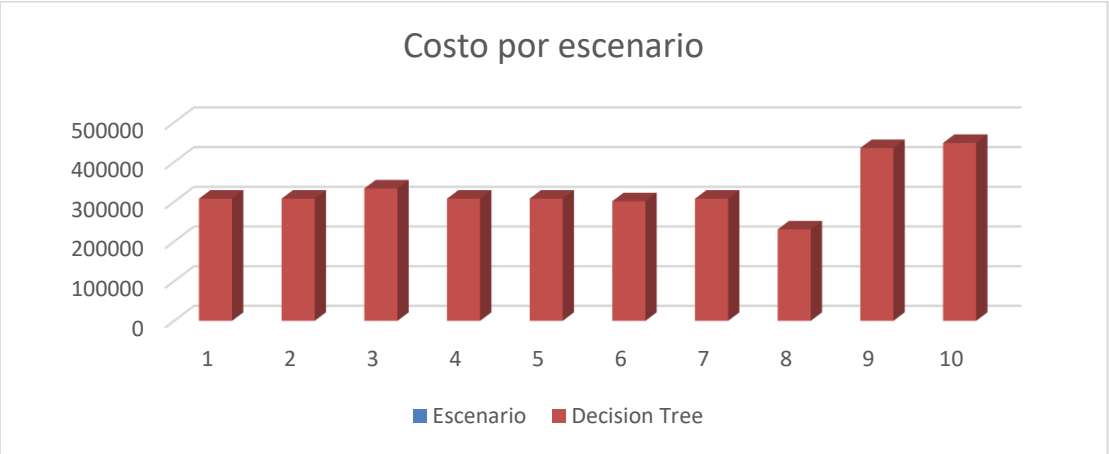
Casi todos los algoritmos (Excepto Support Vector Machine) marcan como el escenario 9 y 10 (Moderna, Con alta superficie, bajo número de cuartos pero construida recientemente) como claves para detectar el patrón que más incrementa el precio y el escenario 8 (Casa Antigua) como el que más disminuye el precio de las casas.

	Algoritmo					
Escenario	G. Linear Model	Deep Learning	Decision Tree	Random Forest	Gr. B. Trees	S. Vector Machine
1	338115,03	324001,49	307775,80	334173,73	303783,17	303643,06
2	338115,03	324328,47	307775,80	334173,73	303783,17	299283,64
3	338115,03	329832,86	333419,50	342996,86	312337,48	310760,37
4	362211,07	349464,36	307775,80	334156,44	303783,17	299340,70
5	310537,66	301744,64	307775,80	342698,07	310880,61	311715,98
6	273077,50	260952,60	301303,00	304227,01	306830,08	296460,74
7	390301,90	382543,38	307775,80	331807,49	327689,17	310759,01
8	247685,58	246634,64	230345,00	247992,38	213431,25	295893,75
9	412102,77	422240,04	435287,33	436980,42	437775,92	312007,47
10	470764,51	437065,87	448046,00	420910,12	441310,26	309162,92

Se recuerda que, Support Vector Machine tuvo la mayor cantidad de fallo respecto, por lo que se descarta como algoritmo detector de patrones por su inestabilidad en los resultados como se ve en el mapa de calor.



Se grafica con barras los datos del mejor algoritmo (Decision Tree) para representar mejor la idea del patrón:



Conclusiones y recomendaciones:

- El mapa de calor sirve para representar visualmente los puntos altos y bajos de un conjunto de datos.
- El algoritmo de Decision Tree de Rapid Miner tiene el mejor desempeño en predicción de precios de casas. Y este determinó que una casa antigua es mucho más barata que una casa moderna con moderada superficie.
- El número de baños influyo in significativamente sobre el precio de una casa.
- El número de cuartos influyó moderadamente en la estimación precio de una casa.
- Los porcentajes de error de los mejores algoritmos rodean al 1% lo cual se considera aceptable. Si se requiere mejorar ese porcentaje es necesario utilizar programación desde 0.
- Se logró utilizar el auto modelador de la herramienta Rapid Miner.
- Se predijo el precio de una casa a partir de un conjunto de datos.
- Se logró identificar el patrón que influye en la compra de una casa.

- Se recomienda utilizar procesamiento en la nube para grandes volúmenes de datos ya que, como se vio en la práctica algunos algoritmos pueden tardar mucho tiempo en testearse y entrenarse.
- Es recomendable utilizar una gráfica para visualizar mejor los resultados y entender el patrón.

Bibliografía

[1] Minitab, «Minitab,» [En línea]. Available: https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/logistic-regression/what-is-a-generalized-linear-model/. [Último acceso: 17 Noviembre 2019].

[2] Mathworks, «Mathworks,» [En línea]. Available: https://la.mathworks.com/discovery/deep-learning.html. [Último acceso: 17 Noviembre 2019].

[3] G. Tutoriales, «Gestión de Operaciones,» 7 Marzo 2016. [En línea]. Available: https://www.gestiondeoperaciones.net/procesos/arbol-de-decision/. [Último acceso: 17 Noviembre 2019].

[4] M. Estévez, «Inteligencia Analítica,» 4 Septiembre 2017. [En línea]. Available: https://inteligencia-analitica.com/random-forest-python/. [Último acceso: 17 Noviembre 2019].

[5] Rapid Miner, «Rapid Miner,» [En línea]. Available: https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/gradient_boosted_trees.html. [Último acceso: 17 Noviembre 2019].

[6] M. Riquelme, «Web y Empresas,» 2 Septiembre 2013. [En línea]. Available: https://www.webyempresas.com/que-es-el-support-vector-machine-en-la-inteligencia-de-negocios/. [Último acceso: 17 11 2019].