

Modeling II

David Guzmán Leyva - A01706417
Enrique Santos Fraire - A01705746
Leonardo Alvarado Menéndez - A01705998
Oscar Enrique Delgadillo Ochoa - A01705935

Índice

Índice	2
Bitácora de cambios	4
Técnica de modelado	4
Modelo 1	5
Seleccionar la técnica de modelado	5
Supuestos de modelado	5
Generar el diseño de las pruebas	5
Diseño de las pruebas	5
Construir del modelo	7
Ajuste de parámetros	7
Modelo	7
Descripción del modelo	7
Evaluar el modelo	8
Evaluación del modelo	8
Parámetros revisados y ajustados	10
Modelo 2	11
Seleccionar la técnica de modelado	11
Supuestos de modelado	11
Generar el diseño de las pruebas	11
Diseño de las pruebas	11
Primera configuración de parámetros	11
Ajuste de parámetros	11
Modelo	11
Descripción del modelo	12
Evaluar el modelo	12
Evaluación del modelo	12
Parámetros revisados y ajustados	13
Segunda configuración de parámetros	13
Ajuste de parámetros	13
Modelo	13
Descripción del modelo	14
Evaluar el modelo	14
Evaluación del modelo	14
Parámetros revisados y ajustados	15
Modelo 3	16

Seleccionar la técnica de modelado	16
Supuestos de modelado	16
Generar el diseño de las pruebas	16
Diseño de las pruebas	16
Primera configuración de parámetros	16
Ajuste de parámetros	16
Modelo	16
Descripción del modelo	16
Evaluar el modelo	17
Resultados de evaluación del modelo	17
Parámetros revisados y ajustados	18
Segunda configuración de parámetros	19
Ajuste de parámetros	19
Modelo	19
Descripción del modelo	19
Evaluar el modelo	19
Resultados de evaluación del modelo	19
Parámetros revisados y ajustados	20
Comparación de modelos	21
Referencias	22

Bitácora de cambios

Versión	Fecha	Autor(es)	Modificaciones
1.0	11/11/2022	David Guzmán Leyva Enrique Santos Fraire Leonardo Alvarado Menéndez Oscar Enrique Delgadillo Ochoa	Línea base
2.0	27/11/2022	David Guzmán Leyva Enrique Santos Fraire Leonardo Alvarado Menéndez Oscar Enrique Delgadillo Ochoa	Segunda iteración

Técnica de modelado

Recordemos que nuestro objetivo es identificar aquellos indicadores que tengan correlación con la presencia de home-office, así como el porcentaje del mismo por comuna. Por lo tanto, estamos ante un problema de predicción, siendo las posibles técnicas de modelado para llegar a una solución[1]:

- Análisis de regresión
- Árboles de regresión
- Redes neuronales
- K Nearest Neighbor
- Box-Jenkins methods
- Algoritmos genéticos

Modelo 1

Seleccionar la técnica de modelado

Regresión lineal

Al tratar con un problema de predicción, vamos a empezar haciendo un análisis de regresión debido a su simplicidad con respecto a otros métodos que también abordan este tipo de problema. Este método se tratará como benchmark para analizar el comportamiento general de los datos.

Supuestos de modelado

- Debido a que este modelo es un benchmark, obtendremos una predicción sencilla, que será la base para predecir el home-office.

Generar el diseño de las pruebas

Diseño de las pruebas

Dado que estamos bajo un problema de regresión, la manera de calcular la precisión de nuestros modelos es circunstancialmente diferente a algo como una clasificación, no es posible predecir un valor exacto pero sí saber que tan cerca es nuestra predicción frente al valor real.

Dentro de las diversas métricas para evaluar los modelos de regresión contamos con 3 principales[2]:

La R cuadrada (R^2)

Es una medida relativa, mide que tanta variabilidad en las variables dependientes puede ser explicada por el modelo. Se obtiene del coeficiente de correlación (R).

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Como se puede observar en la fórmula, se calcula dividiendo la sumatoria del cuadrado de la predicción entre la sumatoria del cuadrado de los valores reales.

No toma en cuenta el problema del sobreajuste (*overfitting*), si se cuenta con muchas variables independientes es probable que se ajuste bien a los datos de entrenamiento pero con un desempeño inferior durante las pruebas. En estos casos se utiliza la R cuadrada ajustada, pues penaliza las variables independientes adicionales y ajusta las métricas para prevenir el sobreajuste.

Mean Square Error (MSE)

Es una medida absoluta de qué tan bueno es el ajuste del modelo.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Se calcula por la suma del cuadrado del error de las predicciones (valor real menos valor predicho) dividido entre el número de datos. Devuelve el valor de qué tanto el resultado predicho se desvía del resultado original. De primera instancia no se pueden sacar demasiadas conclusiones de un solo resultado, sin embargo se cuenta con un valor real que se puede comparar contra otros modelos y así seleccionar el más adecuado.

En otras ocasiones se utiliza el Root Mean Square Error (RMSE), siendo la raíz del MSE, pues su valor puede ser demasiado grande para compararse con facilidad, además que al ser la raíz vuelve a estar al mismo nivel que el error de predicción, por lo que su interpretación es más comprensible.

Mean Absolute Error (MAE)

Es similar al MSE, pero en vez de ser calculado por medio de la sumatoria del cuadrado del error, es la sumatoria del valor absoluto del error.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

El MAE es una representación más directa de la sumatoria de errores. Mientras que el MSE penaliza en mayor medida los errores grandes de predicción al elevarlo al cuadrado, el MAE trata todos los errores de la misma manera.

Dicho esto y dada sus utilidades, estas métricas son las que utilizaremos durante nuestro primer modelo.

Una vez obtenido el dataset para el modelo se establecieron criterios para las pruebas, entre ellos:

- Contar con al menos un r^2 superior (mayor) al modelo Benchmark, durante las pruebas.
- Llegar a un Mean Squared Error (MSE) menor a 5 en el set de prueba.
- Contar con un Mean Absolute Error (MAE) menor a 2 en el set de prueba.

Para el entrenamiento del modelo se definió la separación de datos en train y test de la siguiente manera:

- Train - 80%
- Test - 20%

Construyendo el modelo con el conjunto de train y evaluando con test.

Construir del modelo

Ajuste de parámetros

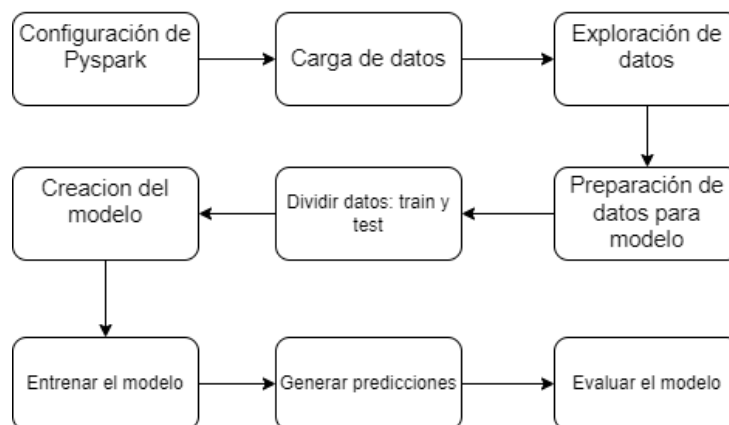
maxIter = 100

Modelo

- Linear Regression

Descripción del modelo

Para el modelo se siguieron los siguientes pasos generales:



- *Linear Regression:*
 - Descripción:

La regresión lineal permite predecir el comportamiento de una variable (dependiente o predicha) a partir de otra (independiente o predictora). Básicamente se busca modelar matemáticamente la variable desconocida o dependiente y la variable conocida o independiente como una ecuación lineal. En el mejor de los casos este modelo es muy efectivo cuando los datos se encuentran con un patrón casi lineal, por el lado contrario si los datos son valores muy dispersos será mucho más difícil generar un modelo eficaz.
 - Stack tecnológico: Apache Spark, Colab, Google Drive
 - Librerías: Pyspark, Spark ML.

Evaluar el modelo

Evaluación del modelo

- *Linear Regression:*

```
RMSE: 5.262
MSE: 27.685
MAE: 3.008
r2: -0.403
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	60.2216	1.348	44.660	0.000	57.478	62.965
a	-2.053e-05	8.32e-05	-0.247	0.807	-0.000	0.000
b	-0.0122	0.009	-1.305	0.201	-0.031	0.007
c	5.161e-05	0.000	0.184	0.855	-0.001	0.001
d	-0.0154	0.010	-1.593	0.121	-0.035	0.004
e	-0.0003	0.001	-0.316	0.754	-0.002	0.002
f	-0.0350	0.028	-1.243	0.223	-0.092	0.022
g	-0.0002	0.001	-0.145	0.885	-0.003	0.002
h	-0.0297	0.040	-0.746	0.461	-0.111	0.051
i	-0.0002	0.001	-0.263	0.794	-0.002	0.001
j	-0.0426	0.069	-0.618	0.541	-0.183	0.097
k	-7.081e-05	0.000	-0.243	0.810	-0.001	0.001
l	0.0001	0.000	0.434	0.667	-0.000	0.001
m	-0.0001	0.000	-0.441	0.662	-0.001	0.000
n	-0.0001	0.000	-0.434	0.667	-0.001	0.000
o	0.0113	0.007	1.556	0.129	-0.003	0.026
p	0.0096	0.006	1.727	0.094	-0.002	0.021
q	1.789e-06	9.27e-05	0.019	0.985	-0.000	0.000
r	1.215e-05	5.02e-05	0.242	0.810	-9e-05	0.000
=====						
Omnibus:	18.476		Durbin-Watson:		2.370	
Prob(Omnibus):	0.000		Jarque-Bera (JB):		128.281	
Skew:	0.055		Prob(JB):		1.39e-28	
Kurtosis:	10.694		Cond. No.		4.25e+06	
=====						

Variables del modelo:

Y = %_of_homeoffice

a = Cantidad de Conexiones de internet fijas

b = Número de empresas sin ventas

c = Cantidad de trabajadores en empresas sin ventas

d = Número de empresas Micro 1

e = Cantidad de trabajadores en empresas Micro 1

f = Número de empresas Pequeña 1

g = Cantidad de trabajadores en empresas Pequeña 1

h = Número de empresas Mediana 1

i = Cantidad de trabajadores en empresas Mediana 1

j = Número de empresas Grande 1

k = Cantidad de trabajadores en empresas Grande 1

l = Consumo de Electricidad en Servicios Comunitarios en M\$

m = Consumo de Electricidad en M\$

n = Consumo de Electricidad Dependencias Municipales en M\$

o = Empresas informáticas

p = Empresas no informáticas

q = Trabajadores en empresas informáticas

r = Trabajadores en empresas no informáticas

La relación que tiene cada variable con el porcentaje de home office de la comuna, la podemos ver en su coeficiente, por ejemplo el 1.22% del número de empresas sin ventas afecta al porcentaje de forma negativa y el 1.13% del número de empresas informáticas afecta de forma positiva, de esta forma podemos ver según el coeficiente de cada variable como afecta al cálculo.

Igualmente podemos ver que tan descriptivo es el valor de la variable según su p-value, donde podemos decir que si un p-value es mayor a 0.05 no es muy significativo. [3] En nuestro caso, no contamos con ninguna variable con un p-value aceptable, pero al ser un modelo benchmark esto no afecta en gran medida.

Por el lado de la multicolinealidad, usando la librería `variance_inflation_factor` (VIF), podemos ver que todas nuestras variables son dependientes entre ellas, ya que todas cuentan con valores por encima de 5, por lo que no todas las variables son tan descriptivas en el caso de la regresión logística.






Estos resultados y el proceso seguidos, se encuentran en este código, en la sección de OLS


<https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Modeling/SparkML.ipynb>

De acuerdo al Business Understanding los objetivos esperados son:

- Identificar los indicadores y su correlación con la presencia de home-office a través del entrenamiento del modelo.
- Conocer el porcentaje de home-office por cada comuna según los indicadores obtenidos previamente a través del modelo. Resultando así en las comunas con mayor presencia de home-office y qué depende de ello.

A continuación se muestra la evaluación de los criterios de éxito de minería de datos.

Criterios de éxito de minería de datos	Cumple con el objetivo (Si ) / (No )	Justificación
Se llega a un Mean Squared Error (MSE) menor a 5 en el set de prueba.		El resultado es 27.685 y el esperado era menor a 5.
Un Mean Absolute Error (MAE) menor a 2		El resultado es 3.008 y el esperado era menor a 2.
Un r^2 superior al modelo benchmark.		El resultado es -0.403, siendo el base como benchmark

Los indicadores establecidos mantienen una correlación verificable con el porcentaje de home-office del área.		Si podemos obtener esta información, pero debido a el resto de criterios, los cuales miden la eficacia del modelo, son altos, la relación de los indicadores con el home-office no sería acertada
---	---	---

Debido a que este modelo es un benchmark, lo que haremos ahora, es buscar un modelo el cual obtenga mejores resultados a este, por lo que vamos a implementar un nuevo modelo, para intentar obtener una mejora.

Parámetros revisados y ajustados

El ajuste de parámetros permite encontrar la mejor configuración de los métodos de optimización ante un determinado problema. Sin embargo, para la implementación de este modelo no se hizo un ajuste de parámetros ya que con la configuración por defecto podemos entrenar el modelo sin ningún inconveniente [4].

Modelo 2

Seleccionar la técnica de modelado

Redes neuronales

Se seleccionaron redes neuronales de forma arbitraria para probar su funcionamiento y cómo se desempeña con respecto al modelo anterior. Esta no fue la mejor decisión metodológica, pero los detalles se abordarán en la fase de evaluación.

Supuestos de modelado

- Por la poca cantidad de instancias que poseemos el modelo tendrá un grado no tan alto de predicción.

Generar el diseño de las pruebas

Diseño de las pruebas

Utilizaremos las mismas métricas presentadas en el [modelo 1](#) para comprobar la efectividad de los diferentes modelos.

Primera configuración de parámetros

Ajuste de parámetros

Learning rate: 0.01

Epochs: 10

Layers

- *Dense => 3:*
 1. Neuronas: 180
Activación: relu
 2. Neuronas: 512 Activación: relu
 3. Neuronas: 256 Activación: relu
 4. Neuronas: 1 Activación: linear
- *Dropout => 2:*
 1. Porcentaje: 20%
 2. Porcentaje: 20%

Modelo

- *Neural Networks*

Descripción del modelo

- *Neural Networks (NNs)*:
 - Descripción:
Las Redes neuronales son un método de la inteligencia artificial que enseña a las computadoras a procesar datos de una manera que está inspirada en la forma en que lo hace el cerebro humano con más de dos capas que le permiten hacer predicciones con una gran precisión.
 - Stack tecnológico: Anaconda, Visual Studio, Jupyter notebook
 - Librerías: Tensorflow, pandas, numpy

Evaluar el modelo

Evaluación del modelo





- Neural Network (NNs) primera configuración:


```
RMSE: 52380.125
MSE: 4647275008.0
MAE: 42886.359375
r2: -1941478980.3054454
```

De acuerdo al Business Understanding los objetivos esperados son:

- Identificar los indicadores y su correlación con la presencia de home-office a través del entrenamiento del modelo.
- Conocer el porcentaje de home-office por cada comuna según los indicadores obtenidos previamente a través del modelo. Resultando así en las comunas con mayor presencia de home-office y qué depende de ello.

A continuación se muestra la evaluación de los criterios de éxito de minería de datos.

Criterios de éxito de minería de datos	Cumple con el objetivo (Si ) / (No )	Justificación
Se llega a un Mean Squared Error (MSE) menor a 5 en el set de prueba.		El resultado es 4647275008 y el esperado era menor a 5.
Un Mean Absolute Error (MAE) menor a 2		El resultado es 42886.3 y el esperado era menor a 2.
Un r2 superior al modelo benchmark.		El resultado es -1941478980.3, siendo bastante menor a lo obtenido

		en el benchmark (-0.403).
Los indicadores establecidos mantienen una correlación verificable con el porcentaje de home-office del área.		No, debido a que los valores obtenidos anteriormente nos dan resultados con los que no es posible evaluar esta métrica.

A partir de la evaluación que realizamos arriba vemos que no se cumple con ningún criterio de minería de datos, en consecuencia no se logra ningún objetivo de minería de datos, por lo que vamos a implementar un nuevo modelo con diferentes parámetros, para intentar obtener una mejora utilizando este método.

Parámetros revisados y ajustados

En la red neuronal utilizó un learning rate de 0.01, para que el aprendizaje lo realice más rápido, utilizando 10 iteraciones porque como es el primer modelo de este tipo a implementar observamos cómo será su funcionamiento. Tenemos 4 capas densas que son en las que se realiza el entrenamiento, en la primera un valor de 180 neuronas con función de activación relu para evitar negativos y esto se aplica en las 2 capas siguientes, en la segunda usamos 512 neuronas, en la tercera 256 y en la última capa 1 con función de activación lineal para que el dato de salida se mantenga con lo entrenado en las capas anteriores, la cantidad de neuronas en las primeras 3 capas, se decidió arbitrariamente. Además se usan 2 capas de dropout que “apaga” cierto número de neuronas para evitar el sobreajuste de nuestro modelo, es decir, evita que el modelo “memorice” los datos, en ambas capas usamos un valor de 0.2 o 20% de las neuronas, elegido arbitrariamente.

Segunda configuración de parámetros

Ajuste de parámetros

Learning rate: 0.001

Epochs: 20

Layers

- *Dense* = 3:
 5. Neuronas: 160
Activación: relu
 6. Neuronas: 480 Activación: relu
 7. Neuronas: 256 Activación: relu
 8. Neuronas: 1 Activación: linear
- *Dropout* = 2:
 3. Porcentaje: 20%
 4. Porcentaje: 20%

Modelo

- *Neural Networks*

Descripción del modelo

- *Neural Networks (NNs):*
 - Descripción:
Las Redes neuronales son un método de la inteligencia artificial que enseña a las computadoras a procesar datos de una manera que está inspirada en la forma en que lo hace el cerebro humano con más de dos capas que le permiten hacer predicciones con una gran precisión.
 - Stack tecnológico: Anaconda, Visual Studio, Jupyter notebook
 - Librerías: Tensorflow, pandas, numpy

Evaluar el modelo

Evaluación del modelo




- Neural Network (NNs) segunda configuración:




```
RMSE: 30622.205078125
MSE: 1582216320.0
MAE: 25411.759765625
r2: -660986423.0481906
```

De acuerdo al Business Understanding los objetivos esperados son:

- Identificar los indicadores y su correlación con la presencia de home-office a través del entrenamiento del modelo.
- Conocer el porcentaje de home-office por cada comuna según los indicadores obtenidos previamente a través del modelo. Resultando así en las comunas con mayor presencia de home-office y qué depende de ello.

A continuación se muestra la evaluación de los criterios de éxito de minería de datos.

Criterios de éxito de minería de datos	Cumple con el objetivo (Si ) / (No )	Justificación
Se llega a un Mean Squared Error (MSE) menor a 5 en el set de prueba.		El resultado es 1582216320 y el esperado era menor a 5.

Un Mean Absolute Error (MAE) menor a 2		El resultado es 25411.75 y el esperado era menor a 2.
Un r^2 superior al modelo benchmark.		El resultado es -660986423.04, siendo bastante menor a lo obtenido en el benchmark (-0.403).
Los indicadores establecidos mantienen una correlación verificable con el porcentaje de home-office del área.		No, debido a que los valores obtenidos anteriormente nos dan resultados con los que no es posible evaluar esta métrica.

A partir de la evaluación que realizamos arriba vemos que se logró una mejora, sin embargo se sigue sin cumplir con ningún criterio de minería de datos, en consecuencia no se logra ningún objetivo de minería de datos por lo que vamos a implementar un nuevo modelo con una técnica diferente de modelado.

Parámetros revisados y ajustados

En la red neuronal mejorada utilizó un learning rate de 0.001, para que el aprendizaje lo realice con más conciencia y evitar así que la red “memorice” los datos, incrementando el número de iteraciones a 20 para que pueda entrenar durante más tiempo.

Se utilizan 4 capas densas como en el modelo anterior, en la primera un valor se disminuyó el número de neuronas de 180 a 160 para disminuir los pesos, en la segunda se hizo lo mismo de 512 a 480, mientras que la tercera capa permaneció con el mismo valor de 256 y en la última capa 1, porque es la cantidad de valores esperados, manteniendo las funciones de activación que tiene todas las capas.

Además se usan 2 capas de dropout con el mismo valor que en el modelo anterior que fue de 0.2.

Modelo 3

Seleccionar la técnica de modelado

Aprendizaje de árboles de decisión

Decidimos aplicar un modelo basado en árboles de regresión, ya que en cada iteración, aprende de los resultados anteriores, y al nosotros tener pocos datos para realizar el modelo, es útil tener esta memoria de las iteraciones anteriores, a diferencia de los modelos implementados anteriormente.

Supuestos de modelado

- El árbol de regresión funcionará mejor que los modelos previamente implementados.
- El boosting para este modelo ocupará modelos de aprendizaje automático muchos más débiles como los árboles de decisiones y cada árbol mejorará a su antecesor.
- Se pueden obtener predicciones más cercanas a las reales ajustando algunos parámetros del modelo antes de entrenarlo.

Generar el diseño de las pruebas

Diseño de las pruebas

Utilizaremos las mismas métricas presentadas en el [modelo 1](#) para comprobar la efectividad de los diferentes modelos.

Primera configuración de parámetros

Ajuste de parámetros

maxDepth = 5
maxIter = 20
maxBins = 32

Modelo

- Gradient-Boosted Trees (GBTs)

Descripción del modelo

Gradient boosting o Potenciación del gradiente produce un modelo predictivo en forma de un conjunto de modelos de predicción débiles, típicamente árboles de decisión. Construye

el modelo de forma escalonada como lo hacen otros métodos de boosting, y los generaliza permitiendo la optimización arbitraria de una función de pérdida diferenciable. Dichos métodos los conocimos el semestre pasado para la solución del reto, por ello decidimos utilizarlo gracias a que podemos hacer que el modelo mejore cambiando el número de árboles o ajustando otros hiperparametros.

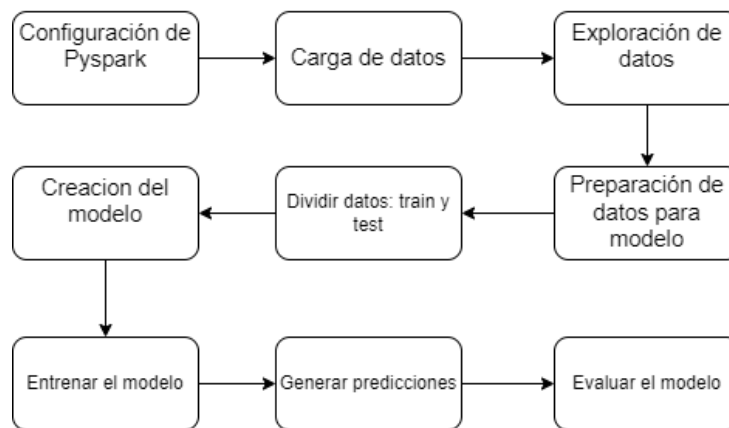
Stack tecnológico:

- Apache Spark
- Colab
- Google Drive

Librerías:

- Pyspark
- Spark ML

Para el modelo se siguieron los siguientes pasos generales:



Evaluar el modelo

Resultados de evaluación del modelo







- Gradient-Boosted Trees (GBTs) 1.0:

```
RMSE: 3.027  
MSE: 9.165  
MAE: 2.259  
r2: -5.730
```

De acuerdo al Business Understanding los objetivos esperados son:

- Identificar los indicadores y su correlación con la presencia de home-office a través del entrenamiento del modelo.
- Conocer el porcentaje de home-office por cada comuna según los indicadores obtenidos previamente a través del modelo. Resultando así en las comunas con mayor presencia de home-office y qué depende de ello.

A continuación se muestra la evaluación de los criterios de éxito de minería de datos.

Criterios de éxito de minería de datos	Cumple con el objetivo (Si ) / (No )	Justificación
Se llega a un Mean Squared Error (MSE) menor a 5 en el set de prueba.		El resultado es 9.16 y el esperado era menor a 5.
Un Mean Absolute Error (MAE) menor a 2		El resultado es 2.25 y el esperado era menor a 2.
Un r^2 superior al modelo benchmark.		El resultado es -5.73, siendo menor a lo obtenido en el benchmark (-0.403).
Los indicadores establecidos mantienen una correlación verificable con el porcentaje de home-office del área.		Si, con el modelo creado, podemos conocer la relevancia de cada indicador para producir el home-office, pero como aún no se cumplen los valores de las métricas establecidas, estos valores pueden no ser tan reales.

A partir de la evaluación que realizamos arriba vemos que solo se cumple con un criterio de minería de datos, en consecuencia no se logran todos los objetivos de minería de datos, por lo que vamos a implementar un nuevo modelo con diferentes parámetros, para intentar obtener una mejora utilizando este método.

Parámetros revisados y ajustados

El ajuste de parámetros permite encontrar la mejor configuración de los métodos de optimización ante un determinado problema. Sin embargo, para la implementación de este modelo no se hizo un ajuste de parámetros ya que con la configuración por defecto podemos entrenar el modelo sin ningún inconveniente [5].

Segunda configuración de parámetros

Ajuste de parámetros

maxDepth = 21
maxIter = 40
maxBins = 18

Modelo

- Mismo que la primera [configuración](#).

Descripción del modelo

Descripción, librerías, stack y pasos iguales que en la primera [configuración](#).

Hiperparámetros:

- `maxIter = Param(parent='undefined', name='maxIter', doc='max number of iterations (>= 0).')`
- `maxDepth = Param(parent='undefined', name='maxDepth', doc='Maximum depth of the tree. (>= 0) E.g., depth 0 means 1 leaf node; depth 1 means 1 internal node + 2 leaf nodes. Must be in range [0, 30].')`
- `maxBins = Param(parent='undefined', name='maxBins', doc='Max number of bins for discretizing continuous features. Must be >=2 and >= number of categories for any categorical feature.'`

Evaluar el modelo

Resultados de evaluación del modelo







- Gradient-Boosted Trees (GBTs) 2.0:

```
RMSE: 2.122  
MSE: 4.503  
MAE: 1.714  
r2: 0.634
```

De acuerdo al Business Understanding los objetivos esperados son:

- Identificar los indicadores y su correlación con la presencia de home-office a través del entrenamiento del modelo.
- Conocer el porcentaje de home-office por cada comuna según los indicadores obtenidos previamente a través del modelo. Resultando así en las comunas con mayor presencia de home-office y qué depende de ello.

A continuación se muestra la evaluación de los criterios de éxito de minería de datos.

Criterios de éxito de minería de datos	Cumple con el objetivo (Si ) / (No )	Justificación
Se llega a un Mean Squared Error (MSE) menor a 5 en el set de prueba.		El resultado es 4.503 y el esperado era menor a 5.
Un Mean Absolute Error (MAE) menor a 2		El resultado es 1.17 y el esperado era menor a 2.
Un r^2 superior al modelo benchmark.		El resultado es 0.634, siendo mejor a lo obtenido en el benchmark (-0.403).
Los indicadores establecidos mantienen una correlación verificable con el porcentaje de home-office del área.		Si, con el modelo creado, podemos conocer la relevancia de cada indicador para producir el home-office.

Debido a que ya cumplimos con todos nuestros criterios de éxito de minería de datos, este será el modelo que usaremos para la predicción.

Parámetros revisados y ajustados

Para mejorar los resultados del modelo inicial en el que se usó los parámetros por defecto del modelo, debemos ajustar unos cuantos parámetros que se adecuen mejor a nuestro dataset. Se cambió la profundidad de los árboles de decisión del modelo con “maxDepth = 21” (número significativo a nuestras edades) de “maxDepth = 5” para mejorar los resultados obtenidos por cada árbol de decisión, aumentamos el número máximo de iteraciones al doble para mejorar la generación de predicciones por la posibilidad de usar más iteraciones con “maxIter = 40” comparado al modelo inicial con “maxIter = 20” y cambiamos “maxBins = 18” de “maxBins = 32” que había en el modelo anterior por defecto debido a que tenemos 18 variables para generar las predicciones.

Comparación de modelos

A continuación se muestra una tabla comparativa de los modelos, describiendo los parámetros especificados, el link al repositorio, así como sus métricas.

Modelos	Configuración	RMSE	MSE	MAE	r2
Linear Regression	maxIter = 100	5.2616	27.6845	3.0077	-0.4027
Neural Network (NN) 1.0	Learning rate: 0.01 Epochs: 10 Layers <ul style="list-style-type: none"> Dense = 3: <ul style="list-style-type: none"> 9. Neuronas: 180 Activación: relu 10. Neuronas: 512 Activación: relu 11. Neuronas: 256 Activación: relu 12. Neuronas: 1 Activación: linear Dropout = 2: <ul style="list-style-type: none"> 5. Porcentaje: 20% 6. Porcentaje: 20% 	52380.125	4647275008.0	42886.35	-1941478980.30
Neural Network (NN) 2.0	Learning rate: 0.001 Epochs: 20 Layers <ul style="list-style-type: none"> Dense = 3: <ul style="list-style-type: none"> 13. Neuronas: 160 Activación: relu 14. Neuronas: 480 Activación: relu 15. Neuronas: 256 Activación: relu 16. Neuronas: 1 Activación: linear Dropout = 2: <ul style="list-style-type: none"> 7. Porcentaje: 20% 8. Porcentaje: 20% 	30622.20	1582216320.0	25411.75	-660986423.04
Gradient-Bosted Trees (GBTs) 1.0	maxDepth = 5 maxIter = 20 maxBins = 32	3.0273	9.1646	2.2593	-5.7304
Gradient-Bosted Trees (GBTs) 2.0	maxDepth = 21 maxIter = 40 maxBins = 18	2.1220	4.5032	1.7139	0.6335

Como se puede observar en la tabla de arriba, contamos con una comparación de los resultados de todos los modelos aplicados, siendo los más destacados aquellos que utilizaron la técnica de árboles de regresión (Gradient-Boosted Tree), particularmente durante su segunda configuración de parámetros, cumpliendo satisfactoriamente todos nuestros criterios de éxito de minería de datos, llegando a un 63% de precisión de predicción.

Referencias

1. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. SPSS inc, 9(13), 1-73.
2. Wu, S. (2020, May 23). 3 Best metrics to evaluate Regression Model? - Towards Data Science. Medium; Towards Data Science.
<https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b>
3. Arias, M., & Arias, M. (2017). ¿Qué significa realmente el valor de p? Pediatría Atención Primaria, 19(76), 377–381.
https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1139-76322017000500014
4. Apache Spark (S.N). LinearRegression.
<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.regression.LinearRegression.html>
5. Apache Spark (S.N). GBRegressor.
<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.regression.GBRegressor.html>
6. Brownlee, J. (2016, 9 septiembre). A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. Machine Learning Mastery.
<https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/>
7. Dagnino, J. (2017, 2 diciembre). REGRESIÓN LINEAL. Revista Chilena de Anestesia. <https://revistachilenadeanestesia.cl/regresion-lineal/>