Caso KAGGLE

Susana Sánchez Ropero,

1494978

APC

1-12-2022

Índice

Introducción

En esta tercera y última práctica de la asignatura de Aprendizaje Computacional usaremos una base de datos de la plataforma Kaggle.

Este proyecto constará de tres partes:

1. Una explicación detallada de los atributos más importantes de la base de datos y en especial del atributo objetivo, el que se va a predecir i/o clasificar.
2. Hacer una breve descripción del método de aprendizaje computacional aplicado, junto con los parámetros escogidos.
3. Y por último, una presentación de los resultados obtenidos.

En mi caso la base de datos con la que trabajaré trata sobre Adidas vs Nike. El un debate constante en la industria del deporte. Este conjunto de datos consiste en la información del producto de estas dos grandes empresas con información significativa.

Los datos de productos de Adidas y Nike se pueden utilizar para una serie de propósitos, como la investigación competitiva.

El link a kaggle es: Adidas vs Nike | Kaggle.

GitHub: https://github.com/susana99ssr/Caso-Kaggle-Aprendizage-Computacional.git.

Base de datos

Para empezar a adentrarme en la base de datos, primero analizo todos y cada uno de los atributos:

	Product Name	Product ID	Listing Price	Sale Price	Discount	Brand	Description	Rating	Reviews	Last Visited
0	Women's adidas Originals NMD_Racer Primeknit S...	AH2430	14999	7499	50	Adidas Adidas ORIGINALS	Channeling the streamlined look of an '80s rac...	4.8	41	2020-04-13T15:06:14
1	Women's adidas Originals Sleek Shoes	G27341	7599	3799	50	Adidas ORIGINALS	A modern take on adidas sport heritage, tailor...	3.3	24	2020-04-13T15:06:15
2	Women's adidas Swim Puka Slippers	CM0081	999	599	40	Adidas CORE / NEO	These adidas Puka slippers for women's come wi...	2.6	37	2020-04-13T15:06:15
3	Women's adidas Sport Inspired Questar Ride Shoes	B44832	6999	3499	50	Adidas CORE / NEO	Inspired by modern tech runners, these women's...	4.1	35	2020-04-13T15:06:15
4	Women's adidas Originals Taekwondo Shoes	D98205	7999	3999	50	Adidas ORIGINALS	This design is inspired by vintage Taekwondo s...	3.5	72	2020-04-13T15:06:15

Ilustración 1: Base de datos

Como se puede observar en la imagen anterior, dispone de 10 atributos y de 3268 entradas.

Los atributos son:

- Product Name: El nombre del producto.
  - Tipo: object.
- Product ID: Un ID único para cada producto.
  - Tipo: object.
- Listing Price: Precio de lista.
  - Tipo: int.
  - Rango: 0 – 29.999
- Sale Price: Precio de venta.
  - Tipo: int.
  - Rango: 449 – 36.500.
- Discount: El descuento que se le puede aplicar al producto.
  - Tipo: int.
  - Rango: 0 – 60.
- Brand: La marca del producto.
  - Tipo: object.
- Description: Una breve descripción del producto.
  - Tipo: object.
- Rating: Valoración del producto.
  - Tipo: float.
  - Rango: 0 – 5.
- Reviews: El número de reseñas que tiene el producto.
  - Tipo: int.
  - Rango: 0 - 223
- Last Visited: La fecha concreta de la última visita que tuvo el producto.
  - Tipo: object.

El atributo objetivo lo estableceremos más adelante, ya que no viene preestablecido.

Data Cleaning and Pre-Processing

Una vez sé de qué trata la base de datos, que atributos contiene y cuál es el atributo objetivo que voy a predecir/clasificar, procedo a limpiar los datos y a preprocesarlos.

Data Cleaning

Valores nulos

Primero de todo, miro si el dataset contiene valores nulos. Y efectivamente tiene, pero únicamente 3 entradas los contienen. Por esta razón he decidido eliminarlas directamente sin hacer ningún tipo de tratamiento especial, como por ejemplo sustituir el valor nulo por la media, ya que considero que no influirá en los resultados, no se pierde información.

Redundancias

Al indagar en la base de datos, me he dado cuenta de que en el atributo Brand hay dos marcas, en concreto 'Adidas ORIGINALS', 'Adidas ORIGINALS', que se refieren a la misma, es información redundante. Por lo que he decidido agruparlas en una sola marca: 'Adidas ORIGINALS'.

Pre-Processing

Matriz de correlación

Ilustración 2: Matriz correlación

Analizando la matriz de correlación, vemos que los atributos con más correlación son Sale Price y Listing Price con 0.31 y Reviews y Discount con 0.31 también.

Atributo Last Visited

El atributo Last Visited indica la fecha en la que un cliente ha visitado ese producto. Pero la información está en formato object.

He transformado este atributo object en datetime64 para que más adelante sea más cómodo trabajar con él.

Me he dado cuenta de que este atributo únicamente contiene fechas con 0 días y 00:36:43 de diferencia. Así que hay información redundante.

Para esto, he creado dos atributos nuevos llamados Minute y Second. Así solo tendré que consultar estos datos.

Atributo Listing Price

Al hacer dataset.describe() y analizarlo, me he dado cuenta que el mínimo de este es de 0. Y no concuerda que el precio sea de 0 euros, debe de ser un error. Por este motivo, lo reemplazo por el precio de venta.

Esto lo hago de la siguiente forma:

```
boolean_condition = dataset['Listing Price'] == 0
Column_Name ='Listing Price'
new_value = dataset['Sale Price']
dataset.loc[boolean_condition, Column_Name] = new_value
```

Exploratory data analysis – EDA

En este apartado analizaré la información de la base de datos. Y de esta forma averiguaré, únicamente centrándome en los datos de esta base de datos, que marca prefiere la clientela, si Nike o Adidas.

Descuentos Nike vs Adidas

Los descuentos son una forma de llamar la atención de los clientes y animarlos a comprar. Es un punto clave para saber que prefiere el comprador.

He mostrado en esta gráfica los descuentos que ofrecen las dos marcas, veamos que obtenemos:

Ilustración 3: descuentos Nike vs Adidas

Claramente se ve en el gráfico que Nike no ofrece ningún tipo de descuento. A diferencia de las tres marcas de Adidas que sí lo hacen. En concreto, Adidas CORE / NEO, con un 36,6% de descuentos. Aunque las tres submarcas de Adidas ofrecen alrededor del 30% de descuento.

Número de artículos en venta Nike vs Adidas

Nike vs Adidas

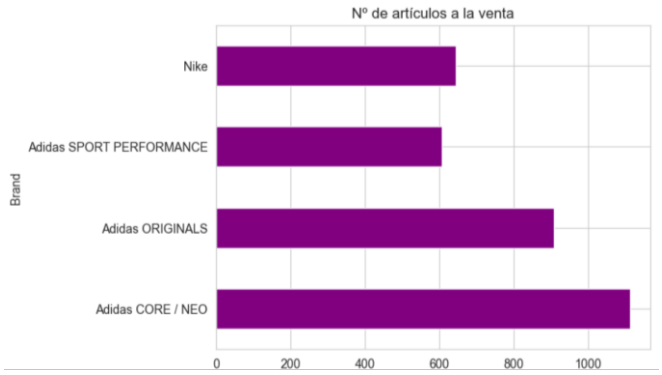


Ilustración 4: Nº artículos

Las tres marcas de Adidas tienen a la venta muchos más artículos que Nike, más de la mitad.

Esto puede hacer que el estudio no sea fiable, ya que tenemos las muestras de las diferentes marcas muy desbalanceadas.

Nike

Nike ofrece la siguiente lista de productos:

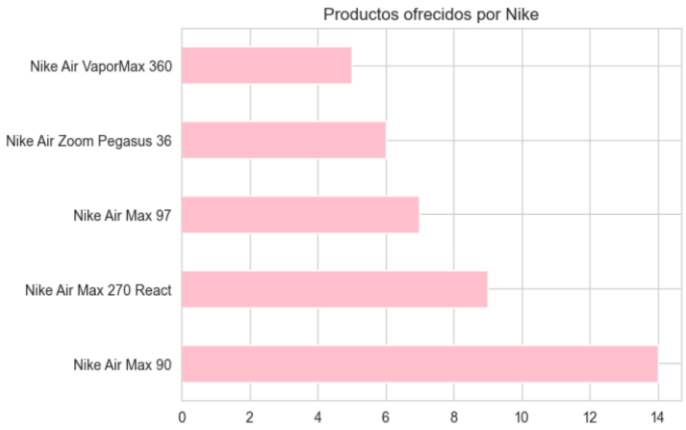


Ilustración 5: Productos ofrecidos por Nike

Nike ofrece 5 tipos diferentes de productos.

Adidas

Adidas ofrece la siguiente lista de productos:

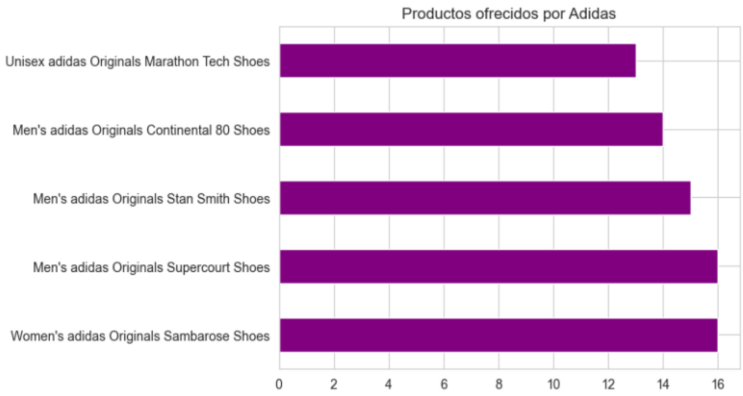


Ilustración 6: Productos ofrecidos por Adidas

Adidas ofrece 5 tipos de productos, igual que Nike.

Precios Adidas vs Nike

Vamos a indagar en los precios de las dos marcas, tanto de venta como de lista.

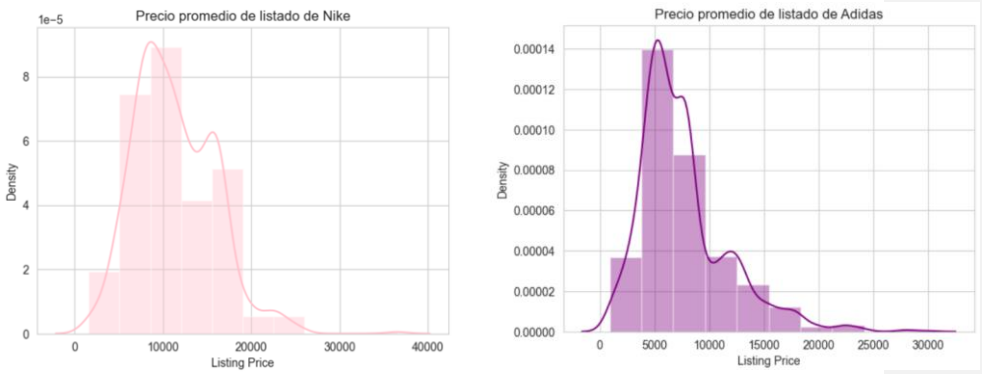


Ilustración 7: Precio listado

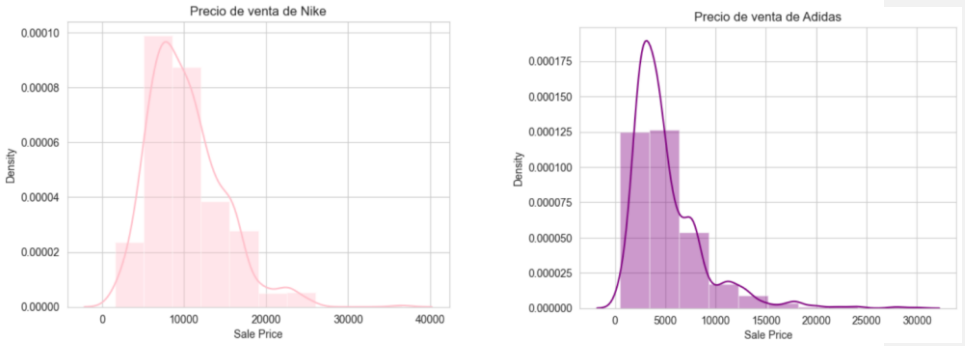


Ilustración 8: Precio de venta

Si observamos los gráficos del precio promedio de listado, en el caso de Nike ronda entre los 7.500 y 10.000. En Adidas va de los 5.000 y 7.500.

Si ahora nos fijamos en los de precio de venta, vemos que los precios de Nike rondan los 5.000 – 7.500. Y los de Adidas también.

Valoraciones y reseñas

Nike

	Rating	Reviews
count	643.000000	643.000000
mean	2.734837	7.181960
std	2.137756	15.968315
min	0.000000	0.000000
25%	0.000000	0.000000
50%	3.800000	1.000000
75%	4.600000	6.000000
max	5.000000	223.000000

Ilustración 9: Valoraciones y reseñas Nike

Las valoraciones de Nike tienen una media de 2.7 sobre 5, no son muy buenas, los clientes no están satisfechos con los productos que ofrecen.

Pero si nos fijamos en la desviación estándar de las reseñas vemos que es bastante significativa, lo que sugiere popularidad.

Adidas

	Rating	Reviews
count	2625.000000	2625.000000
mean	3.366362	48.725714
std	1.159873	28.926042
min	0.000000	0.000000
25%	2.700000	24.000000
50%	3.500000	49.000000
75%	4.300000	74.000000
max	5.000000	99.000000

Ilustración 10: Valoraciones y reseñas Adidas

Si estudiamos las valoraciones de Adidas vemos que de media son un 3.3 sobre 5, mejor que Nike. Esto podría deberse al hecho de que los productos Adidas son bastante baratos y ofrecen grandes descuentos en sus productos.

La desviación estándar para las reseñas es nuevamente buena en comparación con Nike, la posible razón de esto puede ser que Adidas ofrece una amplia gama de productos, mientras que los productos Nike son muy limitados.

Es de mencionar que Nike tiene 643 valoraciones y reseñas, mientras que Adidas tiene 2.625. Considero que la muestra no esta balanceada y esto puede hacer que los resultados no sean del todo fiables.

Clasificación

Una vez hecho el preprocesamiento de la base de datos y la EDA, comenzamos con la clasificación según el atributo target.

En este punto me di cuenta de que no sabía cuál era el atributo a predecir. Así que decidí probar con el que más obvio parecía, el atributo Sale Price.

Recalcar que todos los accuracy 's son sobre el conjunto de validación.

Atributo target Sale Price

Una vez escogido el target que me parecía el idóneo, probé con diferentes métodos de clasificación:

KNN

```
k=1: 424 errores de clasificación de un total de 654
k=2: 466 errores de clasificación de un total de 654
k=3: 484 errores de clasificación de un total de 654
k=4: 493 errores de clasificación de un total de 654
k=5: 503 errores de clasificación de un total de 654
k=6: 509 errores de clasificación de un total de 654
k=7: 518 errores de clasificación de un total de 654
k=8: 527 errores de clasificación de un total de 654
k=9: 528 errores de clasificación de un total de 654
k=10: 536 errores de clasificación de un total de 654
k=11: 542 errores de clasificación de un total de 654
k=12: 540 errores de clasificación de un total de 654
k=13: 549 errores de clasificación de un total de 654
k=14: 547 errores de clasificación de un total de 654
k=15: 546 errores de clasificación de un total de 654
k=16: 550 errores de clasificación de un total de 654
k=17: 555 errores de clasificación de un total de 654
k=18: 555 errores de clasificación de un total de 654
k=19: 555 errores de clasificación de un total de 654
k=20: 555 errores de clasificación de un total de 654
```

Ilustración 11: Clasificación con KNN

Como se puede observar en los resultados después de entrenar un modelo con KNN he obtenido resultados muy malos, ninguna de las K's obtiene un buen resultado y estas clasificaciones no son correctas.

Cómo no sabía cuál era la causa de esto, decidí ir probando con diferentes modelos de clasificación para comprobar si los resultados mejoraban o, por otro lado se quedaban igual.

Logistic Regression

```
Correct classification Logistic  0.5 % of the data:  0.13892288861689106
Model accuracy score: 13.89%
Correct classification Logistic  0.7 % of the data:  0.1620795107033639
Model accuracy score: 16.21%
Correct classification Logistic  0.8 % of the data:  0.154434250764526
Model accuracy score: 15.44%
```

Ilustración 12: Clasificación con Logistic Regression

Podemos ver que la predicción sigue siendo errónea como en el caso del KNN.

Random Forest Regressor

Decidí probar con Random Forest, y en este caso sí obtuve mejores resultados que con los clasificadores hechos con KNN y Regresión Logística.

```
Score RandomForest  0.5 % of the data:  0.9839021768224927
The accuracy of Random Forest is :  60.5875152998776 %
Score RandomForest  0.7 % of the data:  0.9834231713579566
The accuracy of Random Forest is :  65.85117227319061 %
Score RandomForest  0.8 % of the data:  0.9831820097043604
The accuracy of Random Forest is :  68.50152905198776 %
```

Ilustración 13: Clasificación con Random Forest Regressor

En este caso obtengo accuracy's de 0.6, 0.66 y 0.71. Muchísimo mejor que en los dos apartados anteriores.

Random Forest Classifier

```
Score RandomForest  0.5 % of the data:  0.7594859241126071
The accuracy of Random Forest is :  75.9485924112607 %
Score RandomForest  0.7 % of the data:  0.7849133537206932
The accuracy of Random Forest is :  78.49133537206932 %
Score RandomForest  0.8 % of the data:  0.8042813455657493
The accuracy of Random Forest is :  80.42813455657493 %
```

Ilustración 14: Clasificación Random Forest Classifier

En este caso vemos una mejoría en comparación con el Random Forest Regressor.

Aún sin estar contenta del todo con los resultados, aunque con el último clasificador he obtenido una mejora significativa, decidí probar con otro atributo a predecir: el atributo Brand.

Atributo target Brand

KNN

```
k=1: 61 errores de clasificación de un total de 654
k=2: 94 errores de clasificación de un total de 654
k=3: 95 errores de clasificación de un total de 654
k=4: 114 errores de clasificación de un total de 654
k=5: 115 errores de clasificación de un total de 654
k=6: 131 errores de clasificación de un total de 654
k=7: 142 errores de clasificación de un total de 654
k=8: 150 errores de clasificación de un total de 654
k=9: 148 errores de clasificación de un total de 654
k=10: 149 errores de clasificación de un total de 654
k=11: 149 errores de clasificación de un total de 654
k=12: 159 errores de clasificación de un total de 654
k=13: 158 errores de clasificación de un total de 654
k=14: 165 errores de clasificación de un total de 654
k=15: 164 errores de clasificación de un total de 654
k=16: 167 errores de clasificación de un total de 654
k=17: 170 errores de clasificación de un total de 654
k=18: 172 errores de clasificación de un total de 654
k=19: 172 errores de clasificación de un total de 654
k=20: 178 errores de clasificación de un total de 654
```

Ilustración 15: Clasificación KNN

Al cambiar el atributo a predecir, inmediatamente se nota una mejoría en la clasificación. Se puede observar que los errores en la clasificación han menguado drásticamente con KNN.

A ver si en los demás modelos de clasificación ha pasado lo mismo:

Logistic Regression

```
Correct classification Logistic  0.5 % of the data:  0.7515299877600979
The accuracy of Logistic is : 75.15%
Correct classification Logistic  0.7 % of the data:  0.7502548419979612
The accuracy of Logistic is : 75.03%
Correct classification Logistic  0.8 % of the data:  0.7155963302752294
The accuracy of Logistic is : 71.56%
```

Ilustración 16: Clasificación Logistic Regression

En este caso también notamos una gran mejoría, antes obteníamos un accuracy del 0.15 y ahora del 0.75.

Random Forest Regressor

```
Score RandomForest  0.5 % of the data:  0.9304924338920936
The accuracy of Random Forest is :  91.61566707466339 %
Score RandomForest  0.7 % of the data:  0.9444414132741499
The accuracy of Random Forest is :  93.17023445463812 %
Score RandomForest  0.8 % of the data:  0.9523203573632367
The accuracy of Random Forest is :  94.64831804281346 %
```

Ilustración 17: Clasificación con Random Forest Regressor

Con este target y un modelo de clasificación con Random Forest obtenemos un 94% de accuracy. Un muy buen resultado, el mejor hasta el momento.

Random Forest Classifier

```
Score RandomForest  0.5 % of the data:  0.9418604651162791
The accuracy of Random Forest is :  94.18604651162791 %
Score RandomForest  0.7 % of the data:  0.9622833843017329
The accuracy of Random Forest is :  96.2283384301733 %
Score RandomForest  0.8 % of the data:  0.9480122324159022
The accuracy of Random Forest is :  94.80122324159022 %
```

Ilustración 18: Clasificación con Random Forest Classifier

Esta a la par con el Random Forest Regressor.

Atributo target Discount

El otro atributo con mayor correlación es Discount, por lo tanto, hay que estudiar los modelos prediciéndolo:

KNN

```
k=1: 122 errores de clasificación de un total de 654
k=2: 135 errores de clasificación de un total de 654
k=3: 138 errores de clasificación de un total de 654
k=4: 152 errores de clasificación de un total de 654
k=5: 155 errores de clasificación de un total de 654
k=6: 159 errores de clasificación de un total de 654
k=7: 160 errores de clasificación de un total de 654
k=8: 159 errores de clasificación de un total de 654
k=9: 164 errores de clasificación de un total de 654
k=10: 167 errores de clasificación de un total de 654
k=11: 167 errores de clasificación de un total de 654
k=12: 169 errores de clasificación de un total de 654
k=13: 173 errores de clasificación de un total de 654
k=14: 182 errores de clasificación de un total de 654
k=15: 182 errores de clasificación de un total de 654
k=16: 182 errores de clasificación de un total de 654
k=17: 182 errores de clasificación de un total de 654
k=18: 184 errores de clasificación de un total de 654
k=19: 190 errores de clasificación de un total de 654
k=20: 192 errores de clasificación de un total de 654
```

Ilustración 19: Clasificación KNN

Podemos ver que la predicción tiene pocos errores, podría ser un candidato.

Logistic Regression

```
Correct classification Logistic  0.5 % of the data:  0.8494492044063647
The accuracy of Logistic is : 84.94%
Correct classification Logistic  0.7 % of the data:  0.8583078491335372
The accuracy of Logistic is : 85.83%
Correct classification Logistic  0.8 % of the data:  0.8547400611620795
The accuracy of Logistic is : 85.47%
```

Ilustración 20: Clasificación Logistic Regression

El accuracy de este modelo es del 0.85, muy bueno.

Random Forest Regressor

```
Score RandomForest  0.5 % of the data:  0.9646017214177286
The accuracy of Random Forest is :  93.39045287637698 %
Score RandomForest  0.7 % of the data:  0.977991870828687
The accuracy of Random Forest is :  93.98572884811416 %
Score RandomForest  0.8 % of the data:  0.9857593173077135
The accuracy of Random Forest is :  96.94189602446484 %
```

Ilustración 21: Clasificación Random Forest Regressor

Este modelo Random Forest clasificando el atributo Discout es el que mayor accuracy me ha dado, 0.97. Por lo tanto, este va a ser el taregt que voy a escoger finalmente.

Random Forest Classifier

```
Score RandomForest  0.5 % of the data:  0.916156670746634
The accuracy of Random Forest is :  91.61566707466339 %
Score RandomForest  0.7 % of the data:  0.9306829765545361
The accuracy of Random Forest is :  93.06829765545362 %
Score RandomForest  0.8 % of the data:  0.9510703363914373
The accuracy of Random Forest is :  95.10703363914374 %
```

Ilustración 22: Clasificación Random Forest Classifier

En este caso sigue dando mejores resultados el Random Forest Regressor.

SVM

Una vez he escogido el target, quería probar el clasificador SVM.

El mejor resultado me ha dado con los parámetros: C=10.000 y kernel='poly', y son los siguientes:

Correct classification SVM 0.5 % of the data: 0.9253365973072215
Correct classification SVM 0.7 % of the data: 0.9398572884811417
Correct classification SVM 0.8 % of the data: 0.9541284403669725

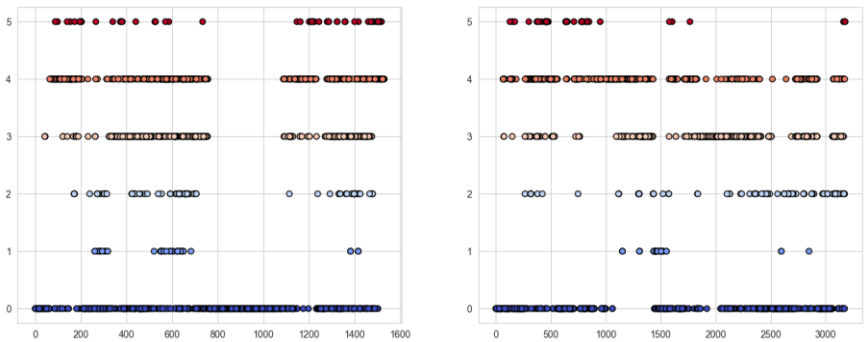


Ilustración 23: SVM

Este modelo clasifica mejor que los anteriores aunque, tarda más que el Random Forest anterior. Así que es mejor coger este último.

Comparación de modelos

Sale Price	Tiempo	C	Kernel	Accuracy 0.8	Best K
KNN	1.3 s	-	-	426/654 errores	1
Logistic Regression	15.3 s	-	-	12.84%	-
Random Forest Regressor	3.4 s	-	-	68.04%	-
Random Forest Classifier	8.3 s	-	-	80.42%	-
SVM	27 min 51 s	10.000	poly	33.94%	-

Brand	Tiempo	C	Kernel	Accuracy 0.8	Best K
KNN	1.3s	-	-	72/654 errores	1
Logistic Regression	0.7 s	-	-	73.09%	-
Random Forest Regressor	3.9 s	-	-	94.34%	-
Random Forest Classifier	2.4 s	-	-	94.80%	-
SVM	19 min 55 s	10.000	poly	79.61%	-

Discount	Tiempo	C	Kernel	Accuracy 0.8	Best K
KNN	1.19 s	-	-	114/654 errores	1
Logistic Regression	1.8 s	-	-	87.16%	-
Random Forest Regressor	3.4 s	-	-	96.94%	-
Random Forest Classifier	6.1 s	-	-	94.03%	-
SVM	3 min 56 s	10.000	poly	92.81%	-

He elegido estos parámetros de SVM ya que eran los que más accuracy daban. He probado C = 0.1, 1, 100, 1000, 10000 y kernel = poli, rbf.

Curva Precision Recall mejor modelo con Random Forest Classifier

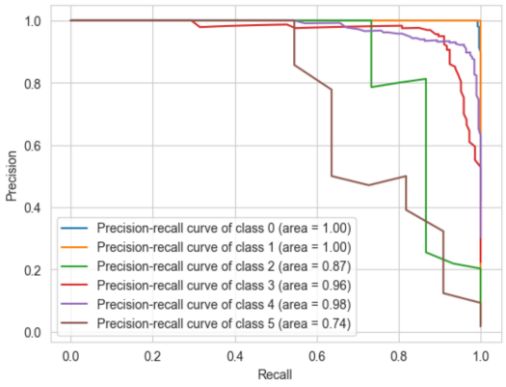


Ilustración 24: Curva PR mejor modelo Random Forest Classifier

Para entender un poco mejor las curvas, especificaré a que corresponden cada una de las 6 clases de los gráficos.

- Class 0 → 50% descuento
- Class 1 → 40% descuento
- Class 2 → 60% descuento
- Class 3 → 0% descuento
- Class 4 →30% descuento
- Class 5 → 20% descuento

En general, podemos ver que casi todas las clases tienen un área bastante cercana al 1. En concreto, los descuentos del 0% y 30% son los más cercanos a 1, siendo los descuentos del 50% y 40% de 1.

El que tiene una área más baja es el 20%

Curva ROC mejor modelo con Random Forest Classifier

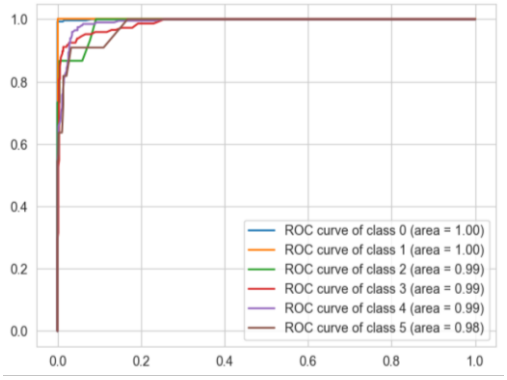


Ilustración 25: Curva ROC mejor modelo Random Forest Classifier

En el caso de las curvas ROC, volvemos a ver unas curvas muy cercanas a 1, de hecho, los descuentos del 50% y 40% son igual a 1, y los restantes son prácticamente 0.99.

Son muy buenos resultados los de las dos gráficas.

R2_score Random Forest Regressor

Tenemos un R2_socre de 0.965 con el modelo hecho con Random Forest Regressor y sobre la validación. Es muy buen valor.

En el modelo hecho con test obtengo un R2 de 0.913. Aunque está por debajo del anterior, sigue siendo bueno.

Gráficas Random Forest Regressor

Validation – Predicted

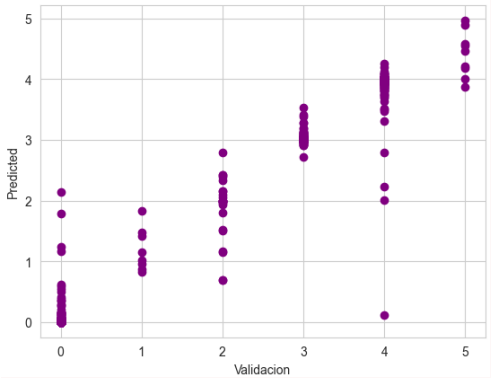


Ilustración 26: Validation Vs Predicted

Los elementos 0, 2 y 4 (50% descuento, 60% y 30% respectivamente) tienen un poco de overfiting mientras que los demás los predice bastante bien.

Test – Predicted

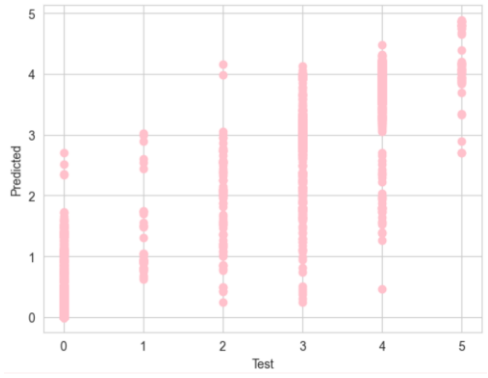


Ilustración 27: Test Vs Predicted

Comentado [s1]: Acabar

En estas gráficas podemos ver que se produce un poco de overfiting.

Los elementos 2, 3 y 4, que corresponden a 60%, 0% y 30% respectivamente, tienden a confundirse, mientras que los 1 y 5 que corresponden, al 40% y 20% respectivamente se predicen bastante bien.

Feature importance Random Forest Regressor

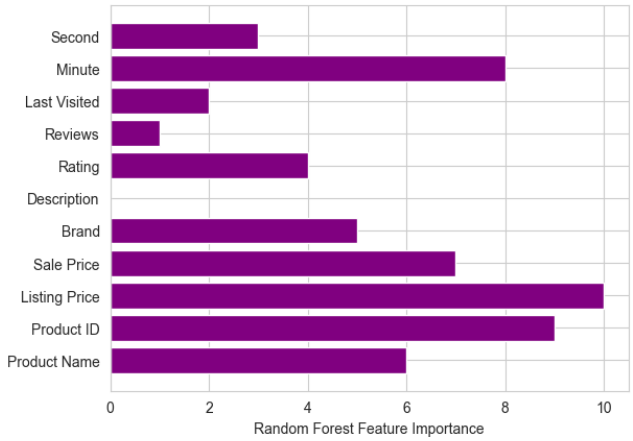


Ilustración 28: Feature Importance RF regressor

Como podemos ver en este gráfico de barras, Random Forest Regressor tiene mucho más en cuenta el atributo Listing Price en el modelo. Pero también tiene mucho en cuenta el Product ID y los Minutos de la última visita.

Estos son los tres atributos que más importancia tienen en este modelo.

Feature importance Random Forest Classifier

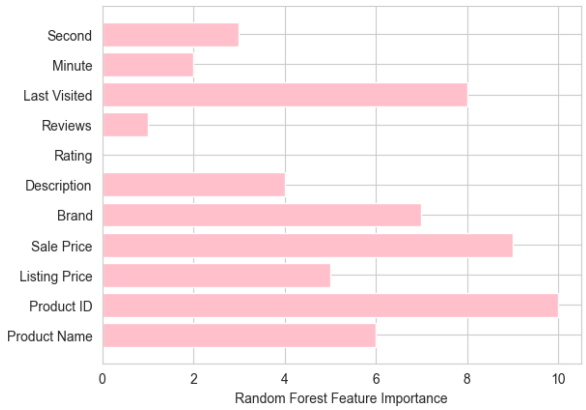


Ilustración 29: Feature Importance RF classifier

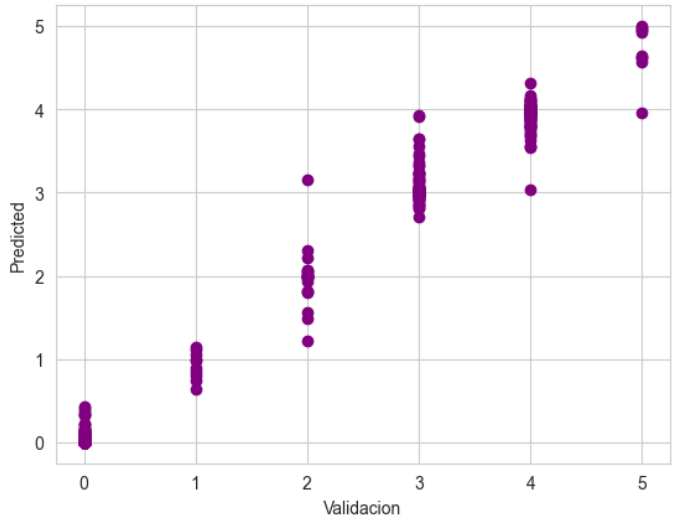
Como podemos ver en este gráfico de barras, Random Forest Classifier tiene mucho más en cuenta el atributo Description en el modelo. Pero también tiene mucho en cuenta las Descripciones de los Sale Price y Last Visited.

Estos son los tres atributos que más importancia tienen en este modelo.

Reducir Overfiting conjunto validation

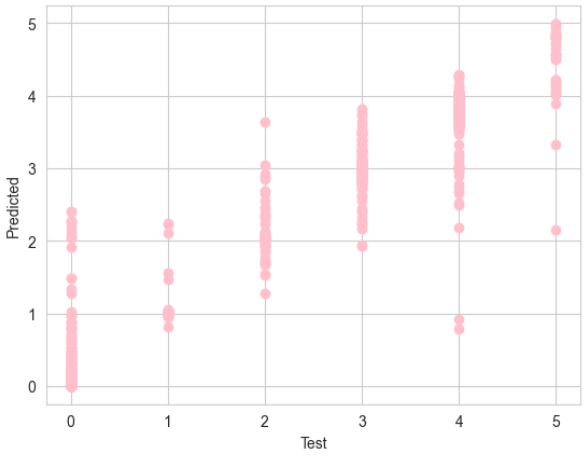
Para ello eliminaré los atributos que menos importancia tienen 'Description', 'Reviews', 'Last Visited'.

Este es el resultado:



Ahora he obtenido un accuracy del 98.47%, en partición train test del 30% - 80%. Superior a lo obtenido anteriormente.

Reducir Overfiting conjunto test



En este caso, he eliminado los atributos 'Rating', 'Reviews', 'Minute', 'Second', 'Description' que son los que menos importancia tienen.

En este caso, la partición 50% 50% de train y test aumentaba a 93.27% de accuracy, también superior a lo obtenido anteriormente, y es la que he usado para mostrar la gráfica. El resto de las particiones daban resultados de overfiting.

PCA

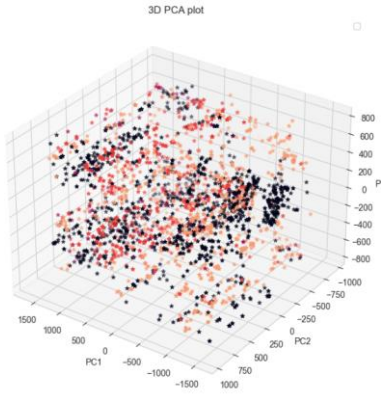


Ilustración 30: PCA

Si observamos el grafico 3D de la derecha que nos muestra el PCA, podemos ver que no sigue ningún tipo de patrón. Esto concuerda con los resultados no muy buenos del modelo de Regresor Logístico.

Así que este gráfico no nos da ningún tipo de información.

Conclusiones

Lo primero a comentar es que Nike está compitiendo con tres submarcas de Adidas y las muestras están muy desbalanceadas. No es un estudio justo.

Pero si me baso en los datos que tengo, veo que Adidas tiene unos precios más bajos y ofrece descuentos a sus clientes, a diferencia de Nike. Por esto, los compradores parecen contentos y lo expresan en las valoraciones y reseñas de la marca.

He encontrado un modelo Random Forest Regressor que obtiene más de un 96% de accuracy prediciendo el atributo Discount que he establecido como target.

El hecho de fijar un target me costó. Probé los que me parecieron más lógicos y me quedé con el que mejores resultados obtenía.

Al solucionar el problema del overfitting obtengo accuracys mejores.