Desafío Final de Análisis de Datos

"Google Analytics Customer Revenue Prediction"

Analistas

Cristhian Angelo Alcántara López

Dario Tonarelli

María Benavente Gómez

Mathias Maximiliano Amarillo Lemos

Miguel Ángel Mesa González

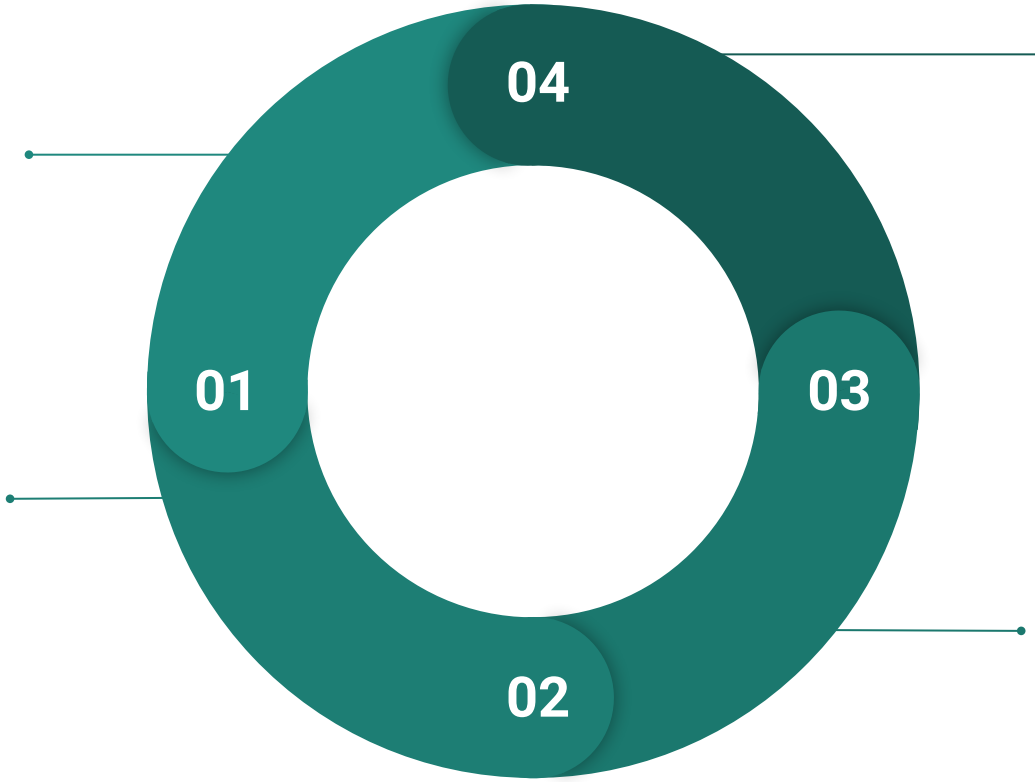
Introducción

Limpieza de Datos

Iteramos sobre este paso varias veces con el objetivo de refinar el tratamiento de datos o de hacer nuestro conjunto de datos más completo

Análisis de Datos

Se analiza la información que contienen los datos, desde cuántos y qué outliers hay, hasta cómo se comportan los usuarios y compradores teniendo en cuenta diferentes ámbitos (localización, hora, plataforma, etc). También se analiza la relación de todas estas variables con la variable objetivo: *totalTransactionRevenue*



Recapitulación

En la sección final se incluyen los problemas encontrados durante la realización de la práctica, los siguientes pasos a dar, así como las conclusiones del proyecto

Modelos de predicción

Explicamos cómo hemos probado diferentes modelos para intentar encontrar unos resultados realistas y que cumpliesen con las métricas adecuadas para este problema en concreto

División de Tareas

	Cristhian	Dario	María	Mathias	Miguel
Limpieza			X		X
Análisis		X		X	
Predicción	X		X		
Otras	X	X	X	X	X

Datos

- Los datos con los que contamos son datos de clientes de Google Merchandise Store, los cuales están en el periodo de 1/8/16 a 15/10/18.

- Los **datos** que proporciona la competición son:
  - *fullVisitorId*
  - *channelGrouping*
  - *date*
  - *device*
  - *geoNetwork*
  - *socialEngagement Type*
  - *trafficSource*
  - *visitId*
  - *visitNumber*
  - *visitStartTime*
  - *hits*
  - *customDimensions*
  - *totals*

- De los cuales *device*, *geoNetwork*, *totals*, *trafficSource* y *hits* son columnas en formato *json*, que necesitan un **tratamiento especial**.

Limpieza de Datos

- **Información relevante** de todas las columnas, sin incluir *hits*
- **Rellenar valores vacíos**, como en el caso de región, que se sustituye por el país si no está disponible
- **Transformaciones** de los datos:
  - Hora, día de la semana, semana del año, etc.
  - ¿Ha comprado previamente?
- Las mismas transformaciones se aplican a los datos de **test.csv**

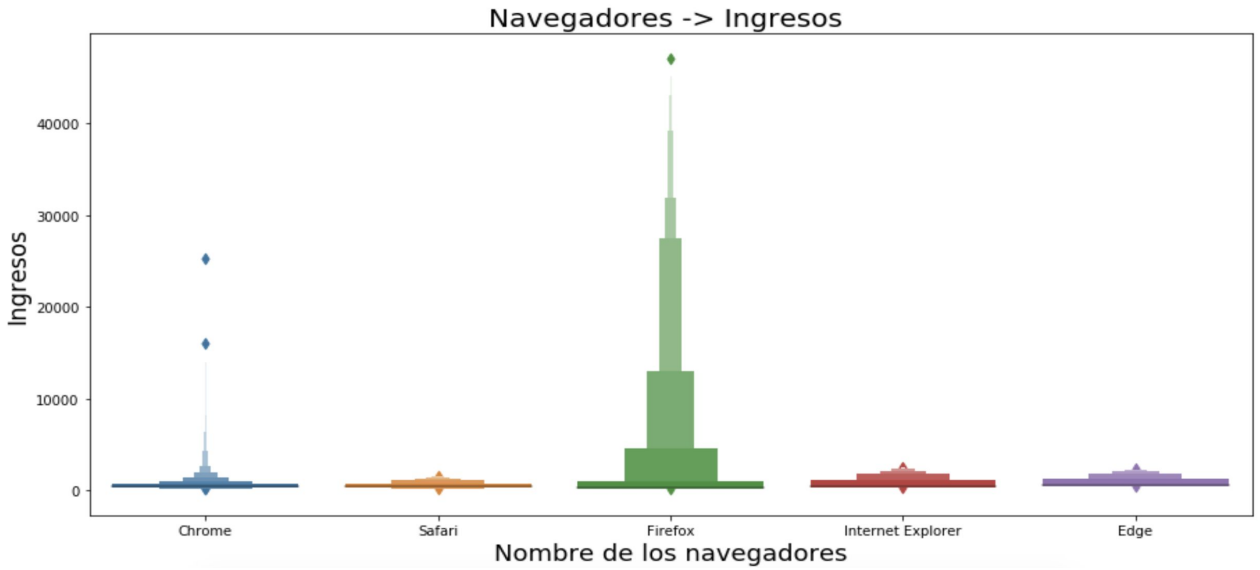

```
{
  "Index": 0,
  "channelGrouping": "Organic Search",
  "fullVisitorId": "5389281631145449 21",
  "visitNumber": 1.0,
  "visitStartTime": "2017-05-29 16:07:23",
  "browser": 1.0,
  "deviceCategory": 3.0,
  "operatingSystem": 3.0,
  "city": "not available in demo dataset",
  "country": "United States",
  "networkDomain": "rr.com",
  "region": "United States",
  "subContinent": "Northern America",
  "hits": 15,
  "newVisits": 1,
  "sessionQualityDim": "",
  "timeOnSite": 178,
  "totalTransactionRevenue": ,
  "transactionRevenue": 0,
  "transactions": "",
  "adContent": 1,
  "adwordsClickInfo_slot": "",
  "campaign": "(not set)",
  "campaignCode": "",
  "isTrueDirect": "",
  "medium": "organic",
  "source": "google",
  "prevPurchase": 0,
}
```

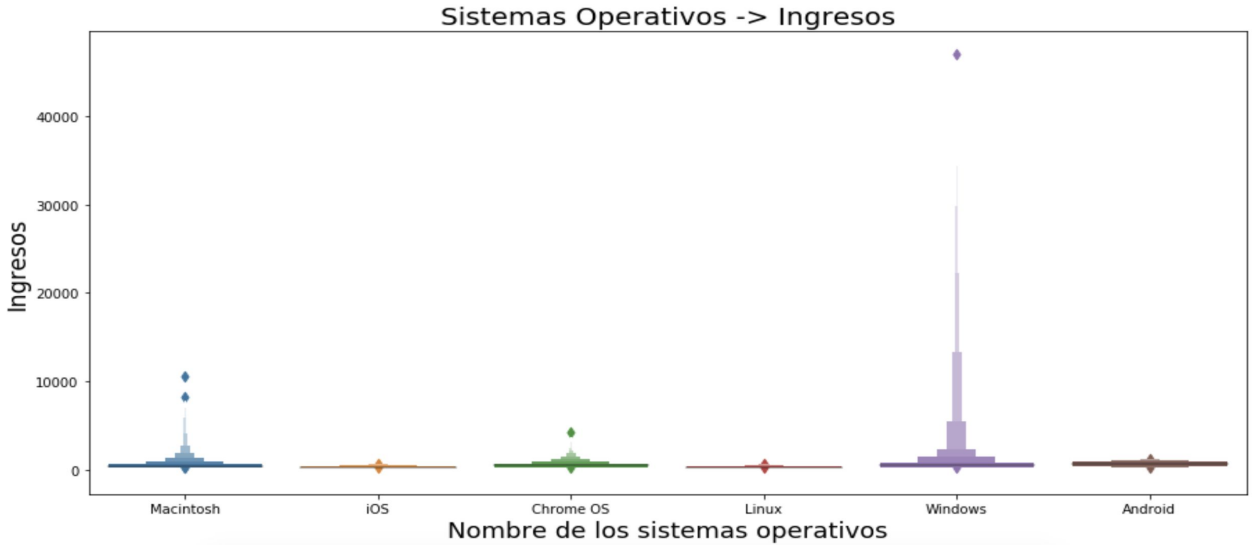
Análisis outliers

Hemos hecho la hipótesis de que los outliers pueden estar causados por dos motivos principalmente:

- Errores de sistema
- Clientes con perfiles particulares

Errores de sistema



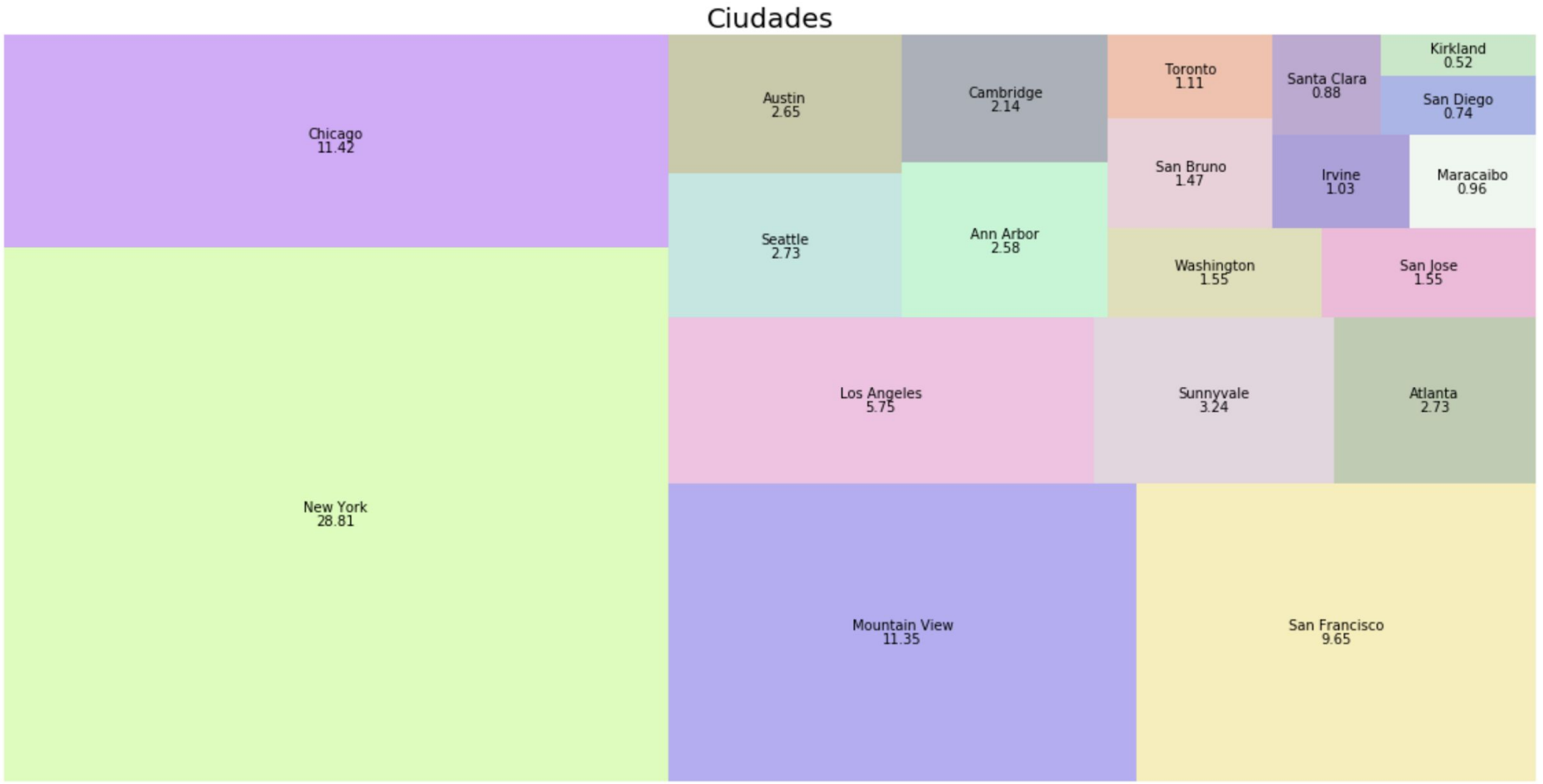



```
1      2011
2       159
3        24
4        11
5         8
6         6
7         5
8         2
12        2
10        1
21        1
15        1
25        1
Name: transactions,
```

Clientes

	channelGrouping	fullVisitorId	visitStartTime	browser	deviceCategory	operatingSystem	country	timeOnSite	totalTransactionRevenue	transactions	date	time	day	month	weekday
268089	Display	1,95746E+18	30/6/17 13:49	Firefox	desktop	Windows	United States	16.7	1205.74	1	30/6/17	13:49:53	30	6	4
278101	Direct	1,95746E+18	30/10/17 13:39	Firefox	desktop	Windows	United States	15.28	1153.13	2	30/10/17	13:39:19	30	10	0
519146	Display	1,95746E+18	18/4/17 15:05	Firefox	desktop	Windows	United States	39.23	32153.82	3	18/4/17	15:05:22	18	4	1
532270	Display	1,95746E+18	14/6/17 17:17	Firefox	desktop	Windows	United States	9.33	2341.56	1	14/6/17	17:17:24	14	6	2
541395	Display	1,95746E+18	19/4/17 15:40	Firefox	desktop	Windows	United States	25.13	3677.92	1	19/4/17	15:40:59	19	4	2
545612	Display	1,95746E+18	2/6/17 19:38	Firefox	desktop	Windows	United States	6.3	1428.25	1	2/6/17	19:38:40	2	6	4
555194	Display	1,95746E+18	28/6/17 17:22	Firefox	desktop	Windows	United States	19.66	2499.0	1	28/6/17	17:22:44	28	6	2
608078	Display	1,95746E+18	7/4/17 20:35	Firefox	desktop	Windows	United States	4.66	3030.2	1	7/4/17	20:35:58	7	4	4
608189	Display	1,95746E+18	7/4/17 15:20	Firefox	desktop	Windows	United States	91.1	6831.96	1	7/4/17	15:20:02	7	4	4
625106	Display	1,95746E+18	1/6/17 18:35	Firefox	desktop	Windows	United States	8.03	242.4	1	1/6/17	18:35:43	1	6	3
661872	Display	1,95746E+18	5/4/17 20:19	Firefox	desktop	Windows	United States	22.78	47082.06	2	5/4/17	20:19:40	5	4	2
662477	Display	1,95746E+18	9/6/17 18:33	Firefox	desktop	Windows	United States	29.8	51.59	2	9/6/17	18:33:27	9	6	4
901577	Display	1,95746E+18	24/3/17 18:36	Firefox	desktop	Windows	United States	4.51	8680.83	1	24/3/17	18:36:00	24	3	4
928438	Direct	1,95746E+18	13/9/17 13:56	Firefox	desktop	Windows	United States	31.25	13231.4	1	13/9/17	13:56:28	13	9	2
1030294	Display	1,95746E+18	4/8/17 17:23	Firefox	desktop	Windows	United States	9.71	613.07	1	4/8/17	17:23:27	4	8	4
1172898	Direct	1,95746E+18	14/2/17 18:30	Firefox	desktop	Windows	United States	48.26	17859.5	1	14/2/17	18:30:28	14	2	1
1218404	Display	1,95746E+18	23/8/17 15:12	Firefox	desktop	Windows	United States	38.66	12297.0	1	23/8/17	15:12:37	23	8	2
1243534	Display	1,95746E+18	8/6/17 18:16	Firefox	desktop	Windows	United States	2.85	27.96	1	8/6/17	18:16:54	8	6	3
1243547	Display	1,95746E+18	8/6/17 16:21	Firefox	desktop	Windows	United States	7.56	882.77	3	8/6/17	16:21:28	8	6	3
1354385	Direct	1,95746E+18	27/9/17 15:02	Firefox	desktop	Windows	United States	13.0	5395.44	1	27/9/17	15:02:18	27	9	2
1478528	Direct	1,95746E+18	11/1/18 15:17	Firefox	desktop	Windows	United States	13.23	332.69	1	11/1/18	15:17:07	11	1	3
1524995	Display	1,95746E+18	2/8/17 17:20	Firefox	desktop	Windows	United States	63.7	28904.83	2	2/8/17	17:20:28	2	8	2
1585131	Display	1,95746E+18	6/9/17 18:25	Firefox	desktop	Windows	United States	18.88	341.54	2	6/9/17	18:25:23	6	9	2
1631903	Display	1,95746E+18	26/5/17 18:29	Firefox	desktop	Windows	United States	15.6	416.96	1	26/5/17	18:29:28	26	5	4

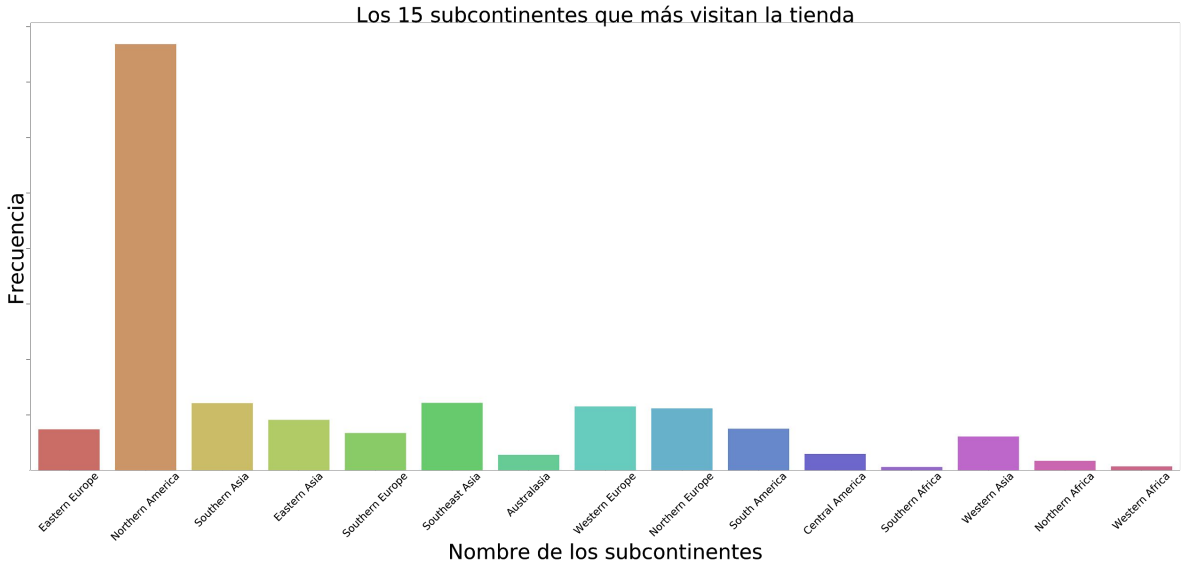
Estado y ciudad



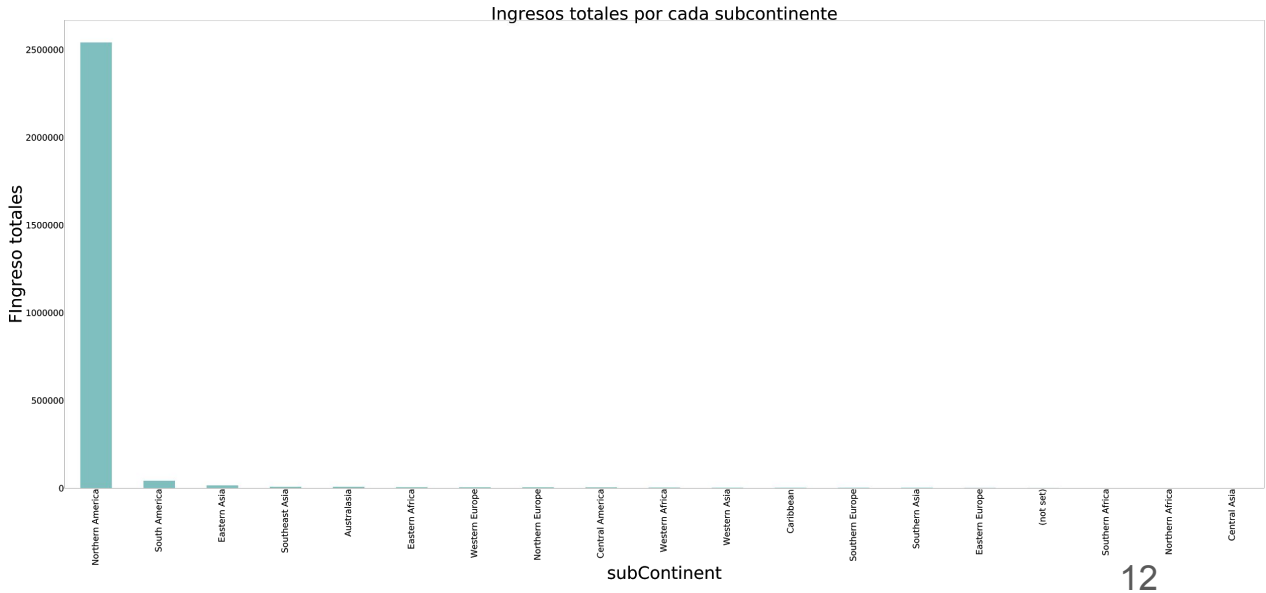
Análisis de datos

- ¿Desde dónde accede el cliente?
- ¿Cuáles son las características del dispositivo?
- ¿Desde que medio accede a la web?
- ¿Cómo se comporta el cliente?

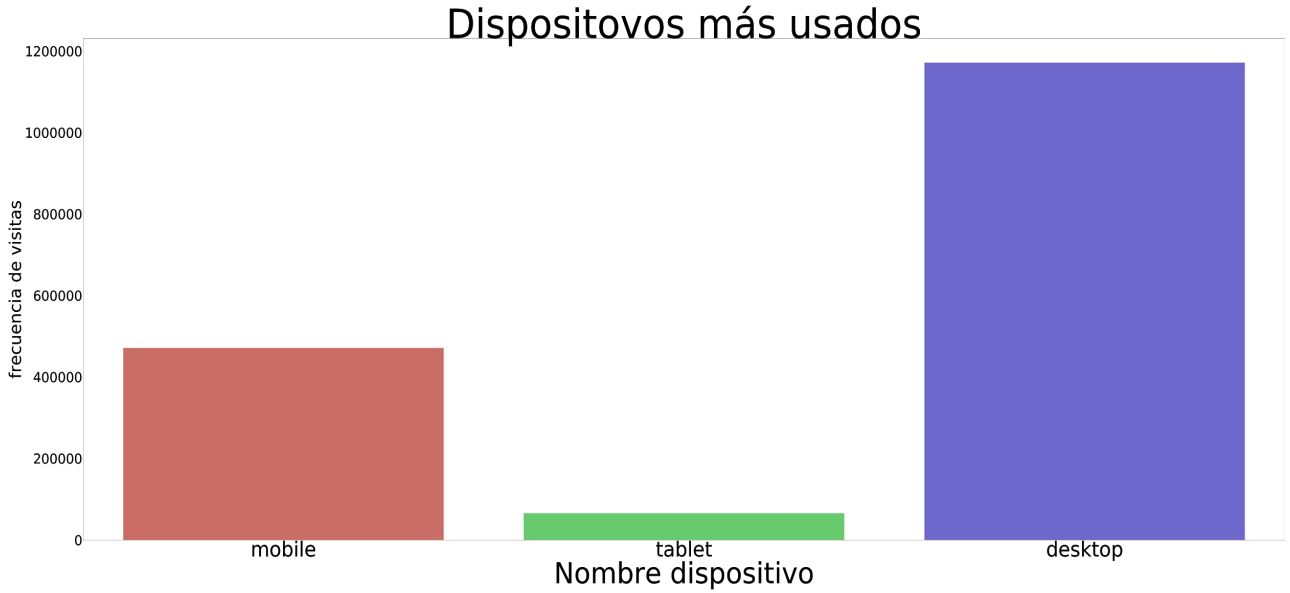
Localización



Porcentaje de compras provenientes de Norte América 96%



Dispositivo

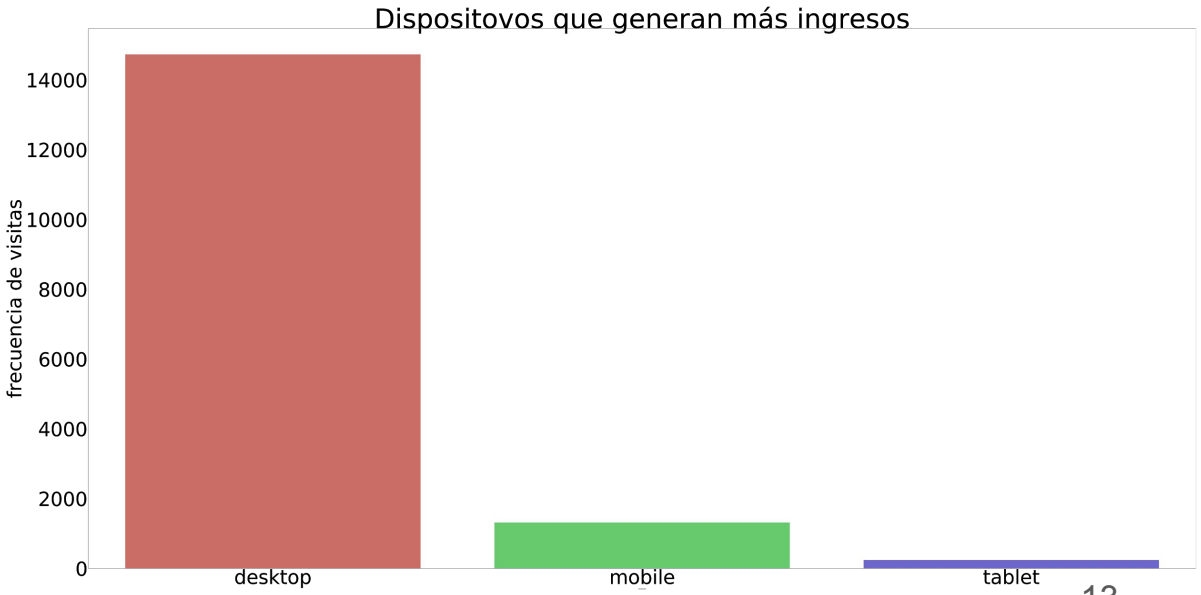


Probabilidad de que un usuario que visita la tienda termine realizando compra según dispositivo:

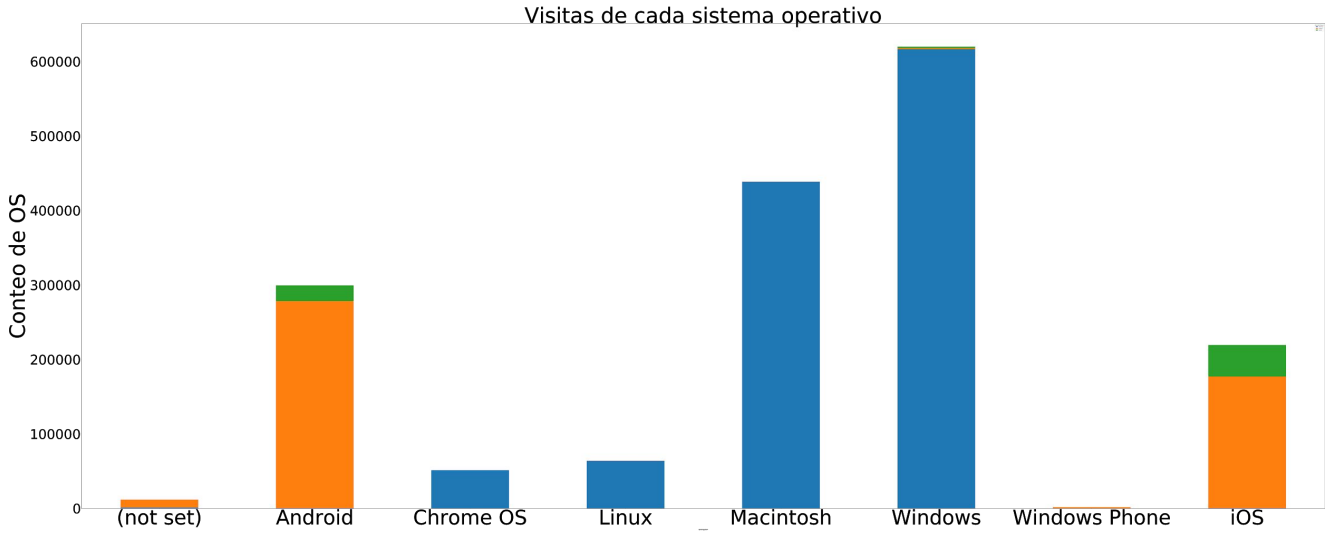
- desktop 1, 46%
- mobile 0,29%
- tablet 0,376%

Valor medio de cada visita dependiendo del dispositivo:

- desktop 0,90 USD por visita
- mobile 0,12 USD por visita
- tablet 0,20 USD por visita



Sistema Operativo

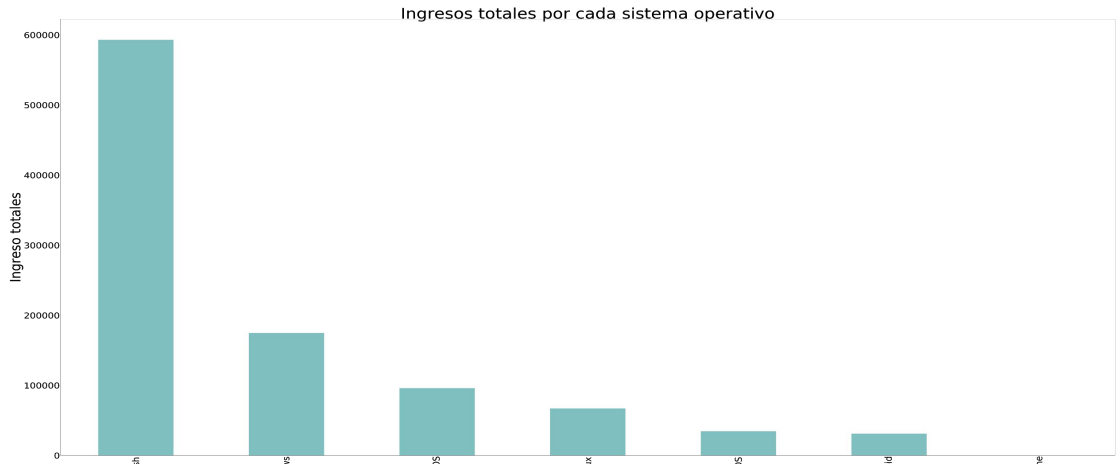


Probabilidad de que un usuario realice una compra dependiendo del sistema operativo que utiliza:

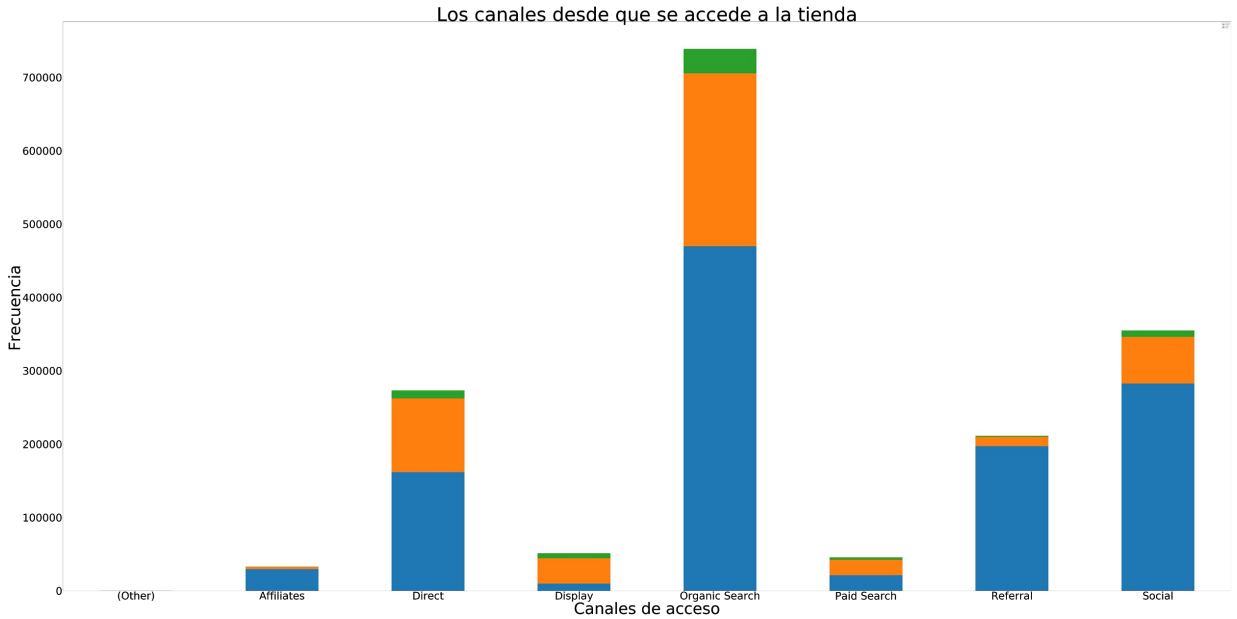
- Macintosh 2,42%
- Windows 0.58%
- Android 0,25%
- Chrome OS 3,43%

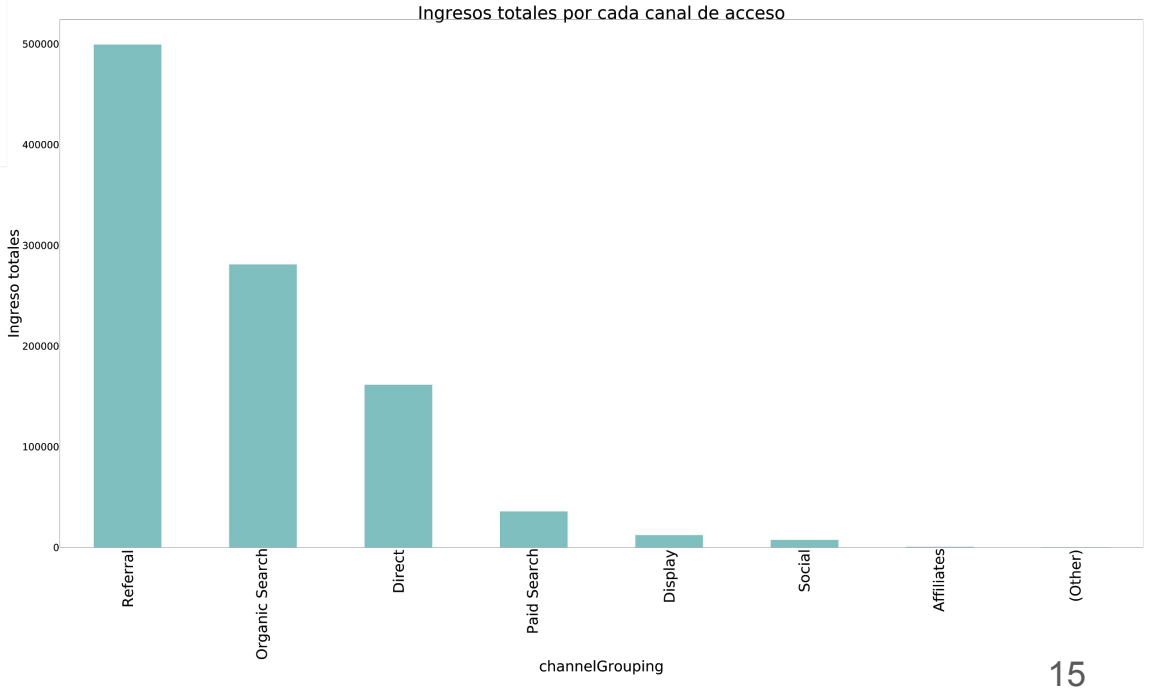
Valor medio de cada visita de cada sistema operativo:

- Macintosh: 1,60 USD por visita
- Windows: 0,33 USD por visita
- Android: 0,11 USD por visita
- Chrome OS: 2,39 USD por visita

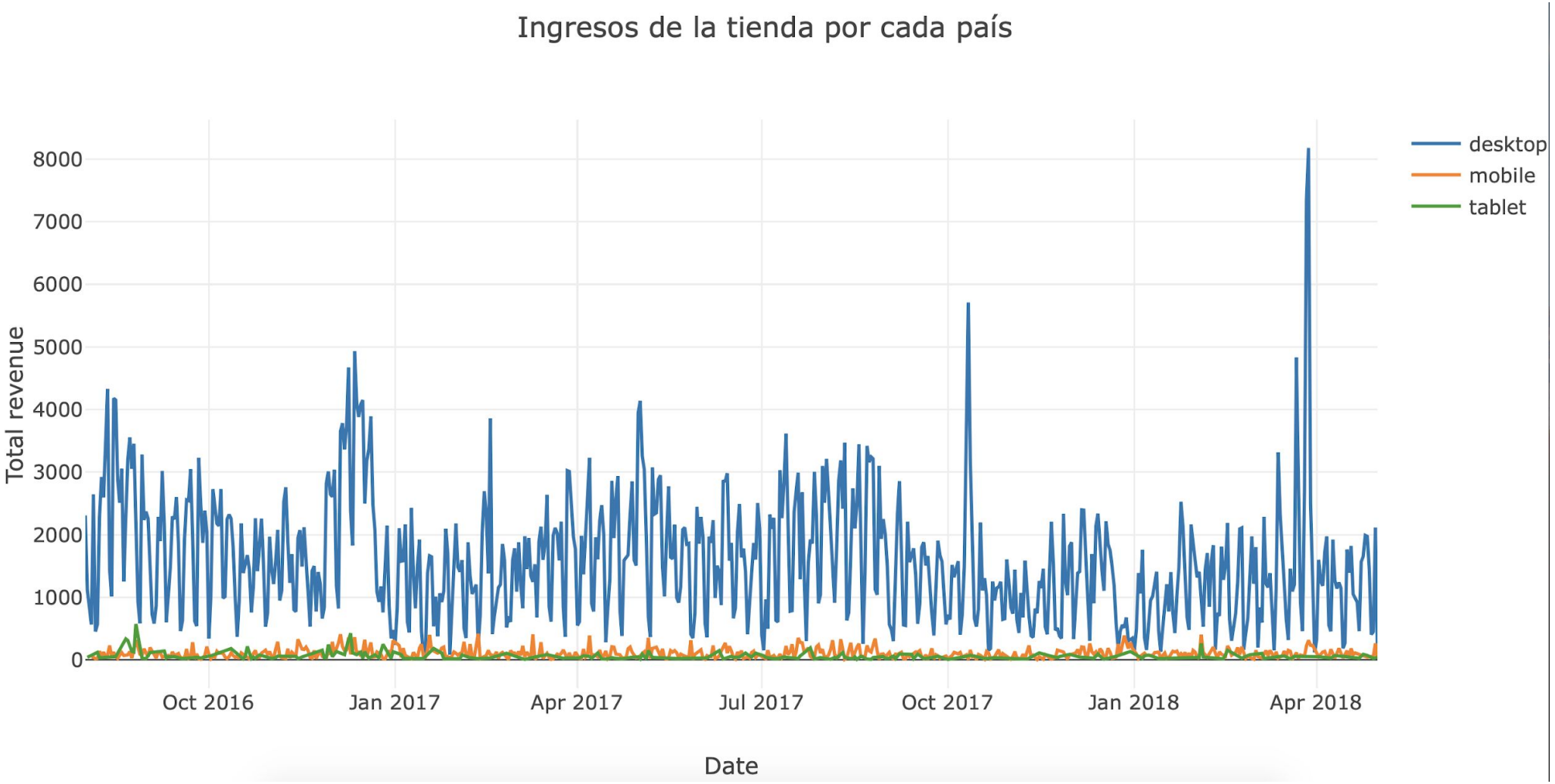


Canal de acceso





Serie temporal de ingresos



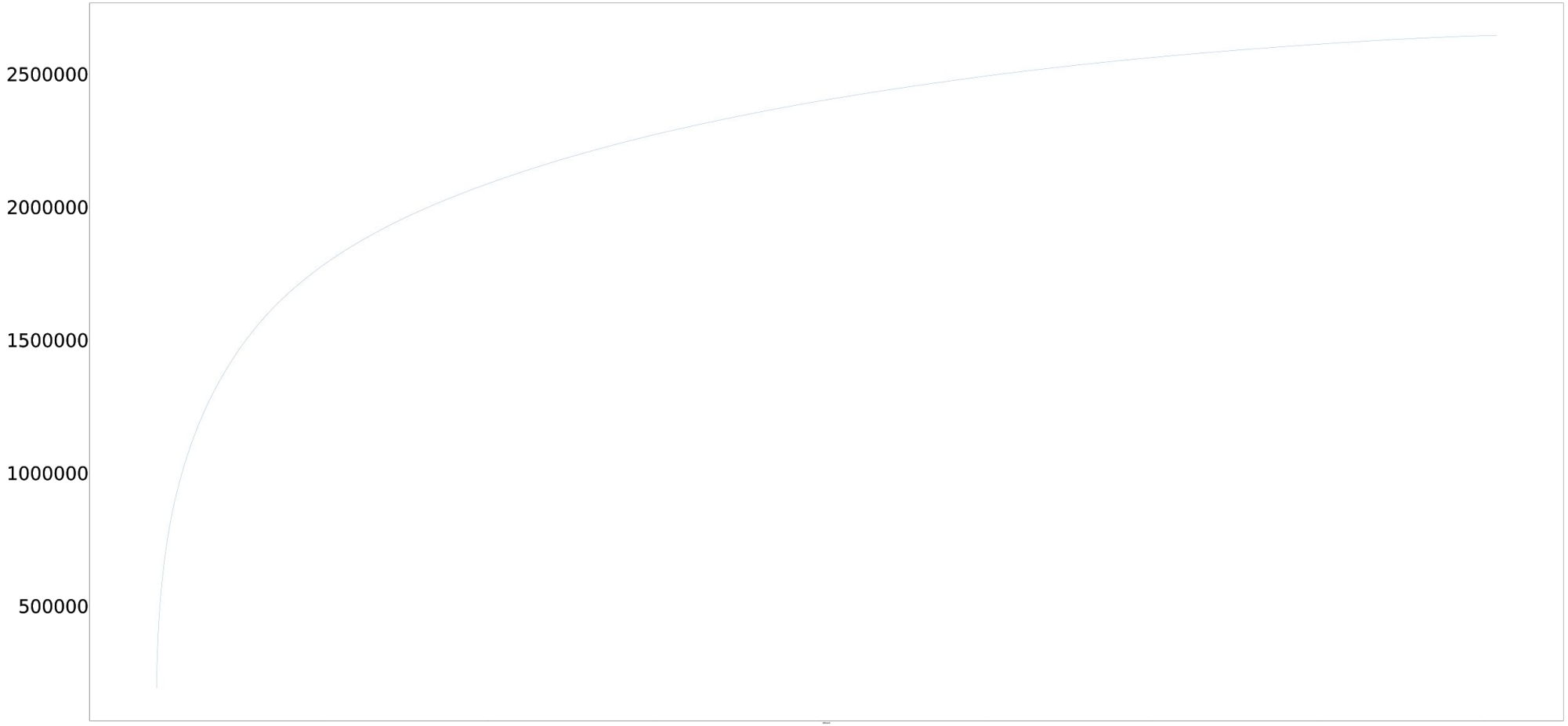
Horario de compra

Los horarios comprendidos en este mapa de valor están en formato UTC (Greenwich Mean Time, GMT)

month	1	2	3	4	5	6	7	8	9	10	11	12
hour												
0	8457.97	8344.15	12208.5	6246.74	7014.17	3381.47	4734.43	11690.1	7179.55	10688.5	8660	10348.3
1	7325.14	6119.89	6498.84	5574.5	3601.18	2679.84	3529.61	9340.83	7423.32	8792.99	7117.37	21167.2
2	6934.56	6753.28	6672.75	4932.67	3856.12	4269.58	2827.08	6195.53	4963.98	7287.33	8599.35	11472.7
3	6201.91	5213.01	8257.22	3607.78	2255.9	3315.85	2749.54	8741.41	6653.29	4615.97	6860.81	8745.08
4	3612.22	3691.8	6208.41	3765.99	4811.47	1771.01	2476.14	7306.06	4390.85	4069.44	7637.99	8636.95
5	2161.12	3932.42	4450.26	3222.11	2785.6	823.78	2341.02	6427	3434.29	2692.6	3161.17	7949.91
6	3525.39	1779.41	3388.2	725.51	660.94	494.66	226.43	2325.66	779.06	1686.36	4531.89	2701.33
7	1490.61	836.65	4696.05	1493.9	449.66	905.52	1396.56	3632.26	1106.13	502.85	207.85	2961.08
8	1102.96	1795.27	4428.08	518.75	207.79	158.87	209.87	1931.24	3797.63	1027.85	1556.12	2240.9
9	1199.86	78.47	1731.48	199.72	295.4	549.13	100.73	300.83	310.13	2907.41	595.38	251.34
10	193.33	336.93	580.18	1121.64	116.97	299.88	0	2618.89	339.45	2596.23	360.69	2596.19
11	37.43	976.25	1859.5	538.73	632.7	1031.02	1334.7	639.32	2564.99	1387.5	376.42	1940.01
12	1483.92	1036.37	2955.34	10257.2	1170.47	2833.9	2440.08	6806.91	1959.75	3171.47	2574.78	1721.36
13	4036.93	2273.39	4982.35	14935	6365.24	11256.3	16838.7	30402.1	24857.8	7358.34	3768.2	3880.33
14	6604.66	8281.07	12877.9	8725.48	8355.54	6109.57	7860.74	10782.8	30160.4	13476.3	9472.84	5599.2
15	12351.7	10927.1	16182.2	58776	8019.49	7025.69	9837.17	37210.3	21073.6	20777.2	14531.3	18381.9
16	14820.9	13984.8	14747.3	15480.2	7148.53	6663.78	8678.84	18151.5	20235.5	16338	18753.1	16798.2
17	12558	10509.4	17249.7	17010.2	10965.6	12806.5	15997.7	63701.9	16622.2	18492	18457.4	21537.4
18	19345.7	30951.9	27731.6	18524.8	13713.9	8949.59	12085.4	36530.6	15624.2	18384.2	12875.1	25744
19	15850.7	18192.2	22274	19810.4	13515	14415.8	31971.9	26718.1	23960.6	14946.6	15695.2	18298.3
20	11554.1	10105.7	23190.9	69019.8	18588.6	14930.6	8105.26	25696.4	19187.5	17301.2	14179.8	15481.3
21	15734.6	13115.2	24360.7	13858.8	7910.26	11596.9	11548.3	21083.3	9733.64	14705.7	13490	13581.5
22	14224.6	15380.2	12176.7	13670.5	8085.36	7419.78	7539.46	20687.6	12252.7	10080.8	15194.1	13871.4
23	11643.8	9860.03	11275.6	16191	6514.73	5054.07	4726.73	18614.2	7868.84	7569.93	9730.05	23485.5

weekday	0	1	2	3	4	5	6
hour							
0	12399.4	12431	17274.4	18929.9	16398.5	15269.1	6251.63
1	7106.45	21139.9	16243	11847.3	15418.3	11356.9	6058.91
2	7011.15	11149.5	13369.8	19409.5	10353.7	9479.44	3991.81
3	6054.76	9825.34	12109.1	13653.1	9745.9	9186.36	6643.2
4	6018.55	10720.6	13632.8	8741.84	7775.04	6131.88	5357.66
5	5039.83	7650.18	9137.11	7399.68	7038.07	4976.64	2139.77
6	3503.44	3215.2	4304.1	4300.69	3197.89	2209.43	2094.09
7	2344.98	1889.9	4481.04	3016.42	3139.83	3170.57	1636.38
8	5472.33	5504.42	1825.28	2251.97	1861.94	1035.16	1024.23
9	2947.46	1135.94	403.69	1109.74	1520.71	620.35	781.99
10	2079.07	671.71	5132.3	1113.62	1311.72	415.66	436.3
11	675.66	2023.3	4814.05	2824.2	1304.16	1115.02	562.18
12	4374.42	5291.1	5701.34	5445.8	14639.1	1011.97	1947.87
13	13001.2	10083.2	32837.6	53716.8	14743.2	3597.31	2975.36
14	20961.6	20264.7	21498.7	20978.9	37232.4	3991.85	3378.4
15	41724.2	61656	48822.3	33038.9	37948.1	4545.59	7358.59
16	19867.7	42864.2	37487	32514.1	29025.5	4918.95	5123.21
17	35883.3	37035	77603.9	42105.7	30190.6	6052.11	7037.51
18	44679.8	56339	41340.4	37819.6	48370.1	5869.06	6042.96
19	32096.5	66708.4	41366.4	34212.3	47584.2	8164.55	5516.51
20	40960.6	40014.7	90688.6	29571.2	30533.6	6220.1	9352.26
21	28263.8	27340.8	37878.2	36503.7	23952.5	7370.61	9409.12
22	34501	26882.7	27283.2	24374.6	22112.5	7575.01	7854.31
23	23166.1	21940.4	34369.8	22318.1	19410	6295.55	5034.58

Regla 20-80



Predicción

LGBM | Árbol de clasificación | Árbol de regresión | Modelo mixto | Elección modelo final

Estamos pronosticando el registro natural de la **suma de todas las transacciones por usuario**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

$$y_{user} = \sum_{i=1}^{n} transaction_{user_i}$$

$$target_{user} = \ln(y_{user} + 1)$$

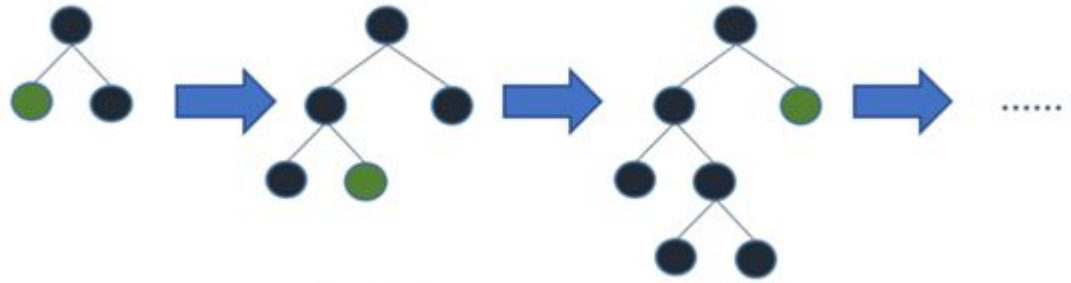
Primer Modelo: LGBM

¿Qué es LGBM (Light GBM)?

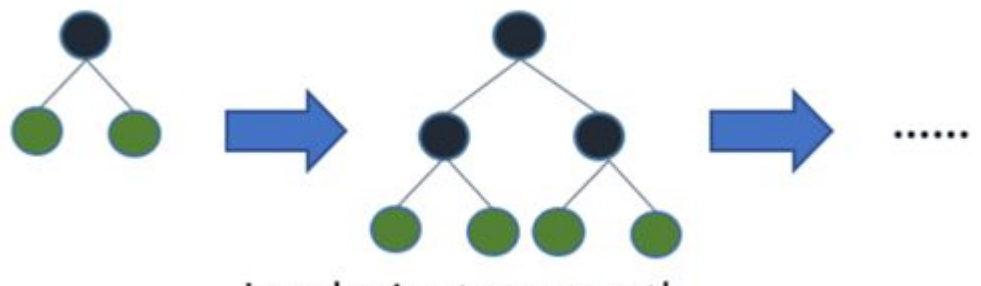
Es una plataforma de mejora de gradiente y de alto rendimiento que utiliza un algoritmo de aprendizaje basado en árboles.

¿En qué se diferencia de otro algoritmo basado en árboles?

Light GBM crece el árbol verticalmente, dividiendo la hoja del árbol con el mejor ajuste mientras que otro algoritmo crece árboles horizontalmente, dividiendo por nivel.



Leaf-wise tree growth



Level-wise tree growth

LGBM: Características

- Es capaz de manejar gran cantidad de datos
- Alta **velocidad**
- Requiere **menos memoria** para ejecutarse
- Se centra en la **precisión** de los resultados
- No es recomendable utilizar este modelo con conjuntos de datos pequeños (*sensible al exceso de información*)

LGBM: Resultados

Error cuadrático medio de entrenamiento y validación:

```
Training until validation scores don't improve for 500 rounds.
[100]   training's rmse: 0.428681       valid_1's rmse: 0.435985
[200]   training's rmse: 0.42755        valid_1's rmse: 0.435834
[300]   training's rmse: 0.426957       valid_1's rmse: 0.435833
[400]   training's rmse: 0.426439       valid_1's rmse: 0.435848
[500]   training's rmse: 0.425972       valid_1's rmse: 0.435898
[600]   training's rmse: 0.425577       valid_1's rmse: 0.435938
[700]   training's rmse: 0.425191       valid_1's rmse: 0.435964
Early stopping, best iteration is:
[228]   training's rmse: 0.427362       valid_1's rmse: 0.435825
```

LGBM: Resultados

Ejemplo de predicción sobre **test.csv**:

	fullVisitorId	PredictedRevenue
0	3404236376816187578	0.034679
1	4914017278166893777	0.072037
2	6232847580219746322	0.005122
3	067325434172403072	0.000000
4	180007823412001644	0.002022
5	5444311544138544790	0.159474
6	331641559902952240	0.062542
7	5335308362644345743	0.135537
8	6033022396879117264	0.224204
9	010912948631735769	0.115923

Problemas encontrados

Debido al **desbalance** de los datos (~99% no compra contra un ~1% que sí compra) nuestro modelo **predice que el gasto es 0** o valores cercanos a 0, por lo cual su **tasa de aciertos global es alta**:

- classification_report:

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	396995
1.0	0.00	0.00	0.00	4594
micro avg	0.99	0.99	0.99	401589
macro avg	0.49	0.50	0.50	401589
weighted avg	0.98	0.99	0.98	401589

- confusion_matrix para la clase 1 (compras)

	$\hat{y} = 0$	**$\hat{y} = 1$**
y = 0	396.992 **[TN]**	3 **[FP]**
y = 1	4.594 **[FN]**	0 **[TP]**

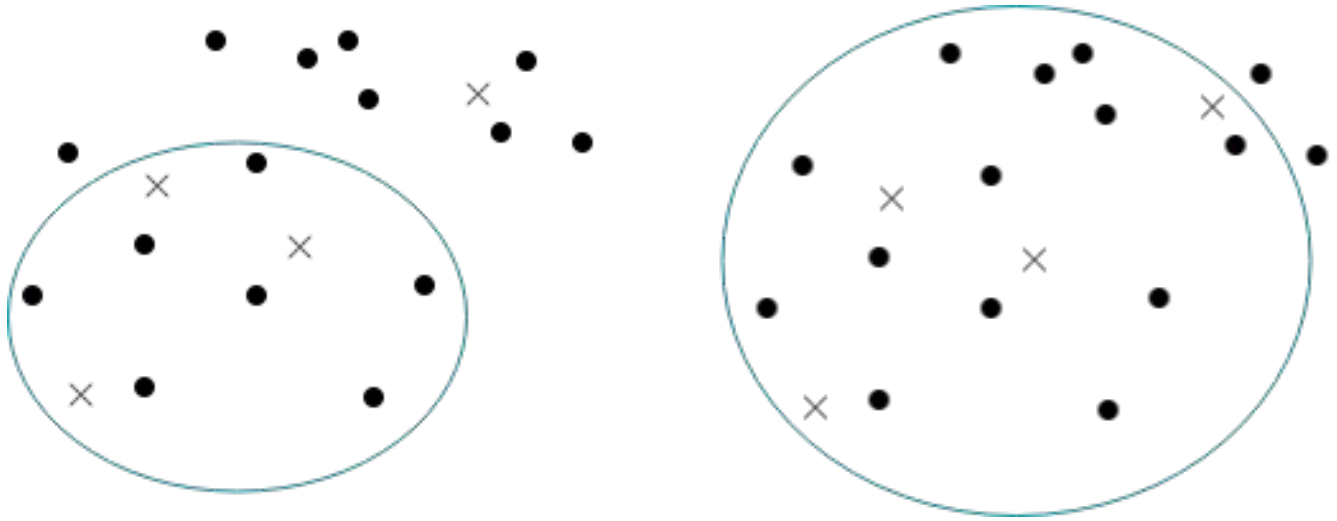
Predicción

LGBM | Árbol de clasificación

- Aplicamos el logaritmo natural más uno a la variable objetivo
- 99% de accuracy y cumple la fórmula, pero no devuelve resultados “reales”
- No tiene en cuenta el desbalance de los datos
- Predice que nunca se compra, por lo que descartamos el modelo

Árbol clasificador

- Intenta predecir si un usuario va a **comprar o no**
- Nos ayudó mucho a **entender el problema**, cómo enfrentarnos a él y cómo evaluar sus métricas
- Hay un *trade off* entre los **falsos positivos** y los **falsos negativos**, y tenemos que encontrar un **balance** entre esas dos métricas:



Árbol clasificador: resultados

- Predicción y métricas sobre **test.csv**:
  - classification_report:

	precision	recall	f1-score	support
0.0	0.99	0.90	0.94	396995
1.0	0.06	0.53	0.10	4594
avg / total	0.98	0.89	0.93	401589

- confusion_matrix para la clase 1 (compras)

	$\hat{y} = 0$	**$\hat{y} = 1$**
y = 0	355.414 **[TN]**	41.581 **[FP]**
y = 1	2.146 **[FN]**	2.448 **[TP]**

Predicción

~~LGBM~~ | **Árbol de clasificación** | Árbol de regresión

- Aplicamos el logaritmo natural más uno a la variable objetivo
- 99% de accuracy y cumple la fórmula, pero no devuelve resultados “reales”
- No tiene en cuenta el desbalance de los datos
- Predice que nunca se compra, por lo que descartamos el modelo

- Variable objetivo en binario: ¿compra o no compra?
- Empezamos a usar otras métricas de evaluación que nos ayudan a entender el problema
- El propio árbol gestiona el balanceo de los datos (*cost sensitive learning*)

Árbol de regresión

- Intenta predecir si un usuario compra, y en ese caso, **cuánto** va a comprar
- Balance manual de los datos (***undersampling*** de los usuario que no compran) para forzar al árbol a "**arriesgarse**" a predecir valores más altos a ~0
- Consideremos **ruido** todos los valores cercanos a cero (**menores a 1.3$**)
- Seguimos utilizando las métricas del árbol anterior, además del **error cuadrático medio**, para saber no sólo el error medio sino qué **porcentaje de acierto** tiene cuando predice compras o no

Árbol de regresión: resultados

- Predicción y métricas sobre **test.csv:**
  - **classification_report**:

	precision	recall	f1-score	support
0.0	1.00	0.62	0.77	396995
1.0	0.02	0.76	0.04	4594
avg / total	0.98	0.62	0.76	401589

  - **MSE**: 1.976

- **confusion_matrix** para la clase 1 (compras)

	ŷ = 0	**ŷ = 1**
y = 0	246.668 **[TN]**	150.327 **[FP]**
y = 1	1.125 **[FN]**	3.469 **[TP]**

Predicción

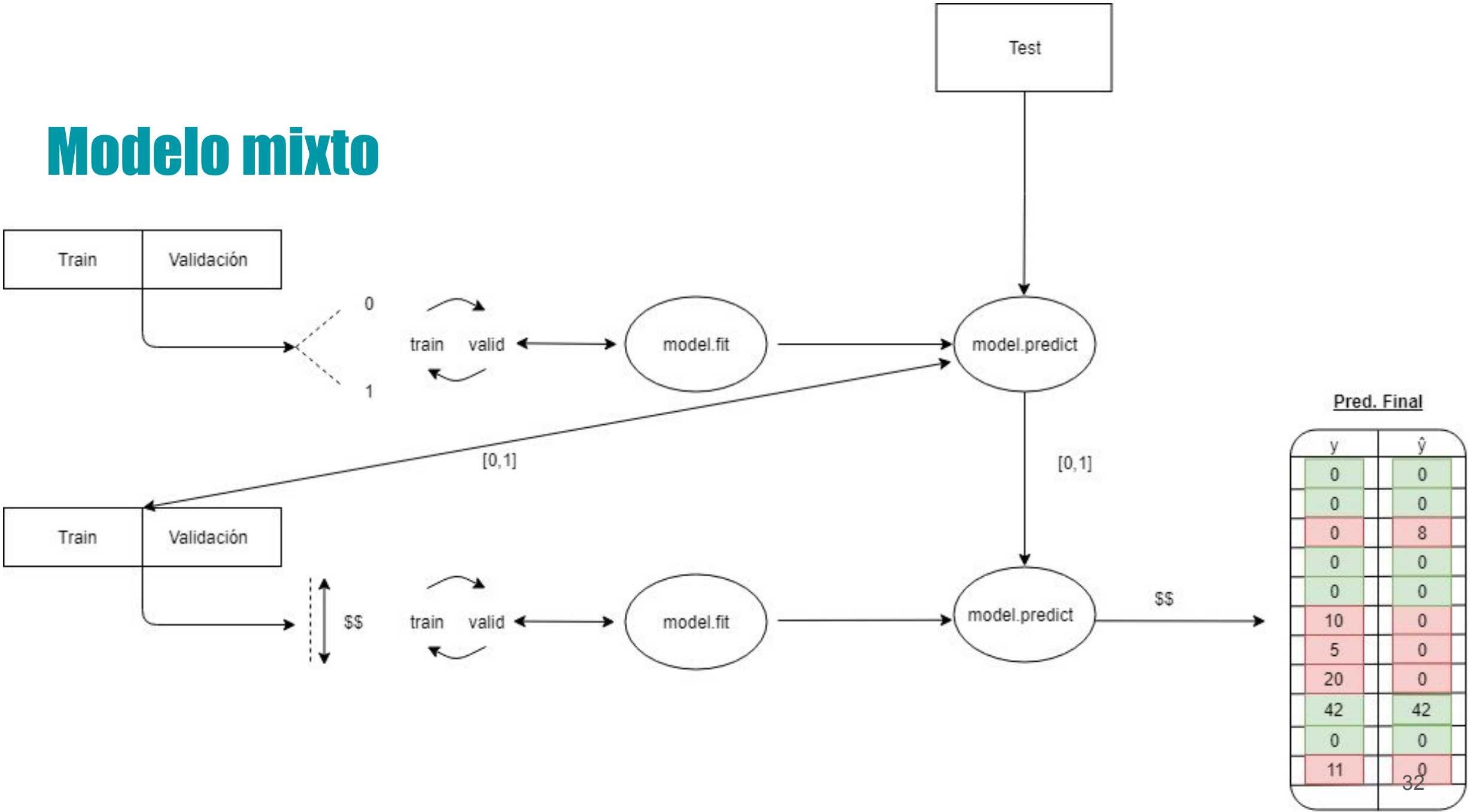
~~LGBM~~ | ~~Árbol de clasificación~~ | **Árbol de regresión** | Modelo mixto

- Aplicamos el logaritmo natural más uno a la variable objetivo
- 99% de accuracy y cumple la fórmula, pero no devuelve resultados “reales”
- No tiene en cuenta el desbalance de los datos
- Predice que nunca se compra, por lo que descartamos el modelo

- Variable objetivo en binario: ¿compra o no compra?
- Empezamos a usar otras métricas de evaluación que nos ayudan a entender el problema
- El propio árbol gestiona el balanceo de los datos (*cost sensitive learning*)

- Volvemos a normalizar la variable objetivo
- Balance manual de los datos (*undersampling*)
- Seguimos utilizando las métricas del árbol anterior, además del error cuadrático medio
- Consideremos ruido todos los valores cercanos a cero

Modelo mixto



Modelo mixto: resultados

- Intentamos mezclar los dos modelos para ver si los resultados de uno ayudan al otro a predecir
- El árbol de clasificación se entrena con outliers
- Predicción y métricas sobre **test.csv:**
  - **classification_report**:

	precision	recall	f1-score	support
0.0	1.00	0.58	0.73	396995
1.0	0.02	0.83	0.04	4594
avg / total	0.99	0.58	0.73	401589

  - **MSE**: 1.536
  - **confusion_matrix** para la clase 1 (compras)

	ŷ = 0	**ŷ = 1**
y = 0	230.173 **[TN]**	166.822 **[FP]**
y = 1	783 **[FN]**	3.811 **[TP]**

Predicción

~~LGBM~~ | ~~Árbol de clasificación~~ | Árbol de regresión | **Modelo mixto** | Elección modelo final

- Aplicamos el logaritmo natural más uno a la variable objetivo
- 99% de accuracy y cumple la fórmula, pero no devuelve resultados “reales”
- No tiene en cuenta el desbalance de los datos
- Predice que nunca se compra, por lo que descartamos el modelo

- Variable objetivo en binario: ¿compra o no compra?
- Empezamos a usar otras métricas de evaluación que nos ayudan a entender el problema
- El propio árbol gestiona el balanceo de los datos (*cost sensitive learning*)

- Volvemos a normalizar la variable objetivo
- Balance manual de los datos (*undersampling*)
- Seguimos utilizando las métricas del árbol anterior, además del error cuadrático medio
- Consideremos ruido todos los valores cercanos a cero

- Unir los dos árboles para hacer una predicción conjunta (*stacking*): el primero trata de orientar al siguiente modelo
- El árbol de clasificación se entrena con *outliers*
- Se descarta porque tiene resultados similares para una complejidad mayor

Modelo final: datos que utiliza

- Los datos más importantes son la hora, la semana del año, el número de visita y el día de semana
- Aquellos datos cuyos pesos son muy inferiores (aportan muy poco a la predicción) han sido eliminados
- Está claro que con estos datos es difícil poder hacer una predicción cuyo margen de error sea el mínimo, por lo que se considera que se tiene que hacer un mayor análisis para *feature engineering* y así encontrar nuevas columnas que ayuden a la predicción


```
{
  "hour" :  0.19258,
  "weekofyear" :  0.18735,
  "visitNumber" :  0.12890,
  "weekday" :  0.12370,
  "city" :  0.07018,
  "month" :  0.05047,
  "region" :  0.04771,
  "operatingSystem" :  0.046338,
  "channelGrouping" :  0.03909,
  "comprasAnteriores" :  0.031229,
  "country" :  0.026724,
  "deviceCategory" :  0.0166,
  "browser" :  0.015733,
  "isTrueDirect" :  0.015095,
  "adContent" :  0.0044,
  "subContinent" :  0.0038242,
}
```

Modelo final: interpretación

- Dimos una interpretación económica a los datos para saber si las métricas muestran un modelo viable o no (es decir, que pueda basarse una campaña de márketing siguiendo sus resultados):

	$\hat{y} = 0$	$\hat{y} = 1$
y = 0	246.668 **[TN]**	150.327 **[FP]**
y = 1	1.125 **[FN]**	3.469 **[TP]**

- **Coste** de hacer campaña a un cliente: 2$
- **Beneficio** medio por cliente: 142$
- **Coste** de nuestra campaña ≤ **beneficio** que reporta
- ∑ **CostePorCliente** * (FP+TP) ≤ ∑ **BeneficioPorCliente** * (TP)
- 2$ * (150327+3469) ≤ 142$ * (3469) => 307592 ≤ 492598
- Se esperan ~1,6$ de cada dólar de inversión

Predicción

~~LGBM~~ → ~~Árbol de clasificación~~ → **Árbol de regresión** → ~~Modelo mixto~~ → **Elección modelo final**

- Aplicamos el logaritmo natural más uno a la variable objetivo
- 99% de accuracy y cumple la fórmula, pero no devuelve resultados “reales”
- No tiene en cuenta el desbalance de los datos
- Predice que nunca se compra, por lo que descartamos el modelo

- Variable objetivo en binario: ¿compra o no compra?
- Empezamos a usar otras métricas de evaluación que nos ayudan a entender el problema
- El propio árbol gestiona el balanceo de los datos (*cost sensitive learning*)

- Volvemos a normalizar la variable objetivo
- Balance manual de los datos (*undersampling*)
- Seguimos utilizando las métricas del árbol anterior, además del error cuadrático medio
- Consideremos ruido todos los valores cercanos a cero

- Unir los dos árboles para hacer una predicción conjunta (*stacking*): el primero trata de orientar al siguiente modelo
- El árbol de clasificación se entrena con *outliers*
- Se descarta porque tiene resultados similares para una complejidad mayor

- Resultados similares sólo con el árbol de regresión
- Priorizamos un modelos más simple y que funciona parecido
- Hace falta mejorar los datos para obtener mejores resultados
- ¡La interpretación económica del resultado final reporta beneficios!

Problemas encontrados

- Encontrar una forma ordenada de trabajar colaborativamente a nivel de gestión de configuración:
    - ¿Combinar el trabajo de los notebooks de *Jupyter*?
    - ¿Trabajar en *Colab*?
- Archivos iniciales muy grandes, necesitaron un tratamiento especial. Además, había muchas entradas incompletas y se usaban términos muy diferentes para referirse a los datos no existentes
- Trabajar con datos tan poco balanceados

Conclusiones

- Compran un 1% de los clientes
- El cliente modelo de esta tienda proviene de Estados Unidos, es usuario de ordenador *desktop* con el sistema operativo de *macintosh*
- Los *outliers* de las compras parecen ser tiendas comprando *stock*
- Casi se cumple la regla 20/80, ya que un 20% de los compradores reportan el 70% de los beneficios totales
- Las campañas no han parecido muy efectivas (sólo dos de ellas reportan ganancias) y el sistema de referenciación funciona mayoritariamente en *desktop*
- Se ha conseguido desarrollar un modelo (mejorable) pero que es económicamente viable: se estima devolver el doble del dinero invertido en la campaña

Siguientes pasos

- Limpieza
    - Incluir nuevos datos:
        - Utilizar algunos valores de hits para hacer un análisis más completo, como los productos que se compran y sus precios
        - Añadir datos externos como festividades, días de envío gratis u ofertas
    - Generar nuevas transformaciones y caracteristicas:
        - Número de visitas por día y por semana
        - Densidad de visitas
- Análisis:
    - Uso de *clustering* para entender grupos de clientes
    - Análisis de series temporales para ver tendencias de compra
    - Definir el perfil del 20% que más beneficio nos reporta
- Predicción:
    - Mejorar el modelo *stacking* especializándolo más y usando intervalos de confianza
    - Utilizar un regresor especializado en datos categóricos como *Catboost*, ya que son mayoría