INFORME PROYECTO 2

Nota Previa: *Este Proyecto he decidido realizarlo de manera individual*

[EX1]

Utilizamos *market_dt.info()* para mostrar el número de columnas, etiquetas de columna, tipos de datos de columna, uso de memoria, índice de rango y el número de celdas en cada columna (valores no nulos). Obteniendo la siguiente tabla:

Nombre de la variable	Cantidad de no nulos	Tipo de dato
City	13335	Object
Customer_Flag	13335	Integer (64)
Revenue	8589	Float (64)
Sector	13235	Float (64)
Legal_form_code	13229	Float (64)
CNT_EMPLOYEE	13335	Integer (64)
CNT_CB_DENSITY	10265	Float (64)
CNT_CB_MOB_DENSITY	10265	Float (64)
CNT_CB_FN_DENSITY	10265	Float (64)
Mobile_potential	13335	Float (64)

Podemos observar que el número de entradas es de 13335, por que la variable que contendrá más nulos es *Revenue* con un total de 4746.

Además, también podemos ver como las variables *City, Customer_Flag, CNT_EMPLOYEE* y *Mobile_potential* tienen una cantidad de 13335 no nulos, es decir, no contienen ningún valor nulo.

Podemos considerar *City* como objeto debido a que esta agrupa todo el conjunto de variables.

[EX3]

Una vez hemos creado los datasets customer_dt y noncustomer_dt basandonos en la flag. Continuamos creando cada uno de los bloxpots *(Revenue, CNT_EMPLOYEE, Mobile_potential y CNT_CB_DENSITY)* para ambos datasets.

Observando los boxplots podemos ver que son bastante similares entre ellos ,pero podemos encontrar algunas diferencias.

Si nos fijamos en la variable CNT_CB_DENSITY vemos que es la variable donde se pueden apreciar mayores diferencias, donde en customer_dt obtenemos valores mucho mayores, menos varianza y menos outliers que en noncustomer_dt.

Una de las similitudes que podemos apreciar es como en ambos dataset obtenemos el mismo máximo en cuanto a la variable CNT_EMPLOYEE (55)

En cuanto a los outliers, he usado la Regla 1,5(IQR), es decir, considerar outliers las que no se encuentran en el rango [Q1 – 1.5(IQR), Q3 +1,5(IQR)] , donde IQR = Q3 – Q1. Podemos apreciar como en el caso de noncustomer_dt obtenemos una mayor cantidad de outliers, concretamente para la variable de Revenue obtenemos 95 outliers en customer_dt y un total de 569 en noncustomer_dt.

Con los resultados obtenido en los cálculos de los cuartiles podemos construir las siguientes tablas respecto a las variables Revenue y Mobile Potential

Customers

	Revenue	Mobile Potential
Q1	1047500	2090.6967…
Q2	2200000	2401.4646…
Q3	4195000	2826.2351…

Non-Customers

	Revenue	Mobile Potential
Q1	902986	1975,5165…
Q2	1750000	2277,9727…
Q3	3501123.5	2631.9261…

[EX5]

En la siguiente tabla podremos ver los resultados obtenidos:

Se puede apreciar como en el caso de los customers la ciudad con mayor ratio es München con 0.024059 , mientras en el caso de los Non-Customers es Köln con 0.016093

Customer	Non-Customer
München 0.024059 Köln 0.020921 Chemnitz 0.017782 Dresden 0.015690 Berlin 0.015690 ... Hiddenhausen 0.001046 Eppelheim 0.001046 Flörsheim 0.001046 Bad Abbach 0.001046 Wedemark 0.001046 Name: City, Length: 557, dtype: float64	Köln 0.016093 Bremen 0.009982 Stuttgart 0.009778 Dortmund 0.009371 Dresden 0.009167 ... Salzbergen 0.000204 Bonstetten 0.000204 Neusäß 0.000204 Wolfhagen 0.000204 Solms 0.000204 Name: City, Length: 2126, dtype: float64

[EX6]

Hemos obtenido los siguientes valores con un valor de test_size de 0,20:

- *Longitud de X_train: 4692*
- *Longitud de X_test: 1173*
- *Longitud de final_dataset (20%): 1173,0*
- *Longitud de final_dataset (80%): 4692.0*

Podemos ver como el 20% del dataset final coincide con la longitud del dataset de test y el 80% restante coincide con la longitud del dataset de entrenamiento.

[EX7]

Observando los histogramas se puede ver claramente cómo en ambos casos (y_train, y_test) tenemos muchos más datos que pertenecen a la clase 0 que a la clase 1, lo que crea que los datasets no estén balanceados, para que estuvieran balanceados deberían tener una cantidad similar de datos.

Al tener una mayor cantidad de información sobre la clase 0 respecto a la clase 1 implicará de forma negativa a nuestro clasificador, ya que nuestro modelo aprenderá a clasificar mejor los datos de la clase 0 que los de la clase 1 , concretamente podría afectar a la precisión, recall y accuracy.

[EX12]

Podemos observar cómo ambos modelos, una vez balanceados, tienen unos valores similares tanto de precision, recall, f1-score y de accuracy. A continuación, vamos a comparar los resultados obtenidos en este apartado con el anterior.

SVC	Decision Tree
precision recall f1-score support 0 0.55 0.73 0.63 165 1 0.62 0.43 0.51 171 accuracy 0.57 336 macro avg 0.58 0.58 0.57 336 weighted avg 0.59 0.57 0.56 336 SVC Accuracy: 0.5744047619047619	precision recall f1-score support 0 0.58 0.59 0.59 165 1 0.60 0.58 0.59 171 accuracy 0.59 336 macro avg 0.59 0.59 0.59 336 weighted avg 0.59 0.59 0.59 336 DT Accuracy: 0.5892857142857143
Confusion Matrix	
[120 45] [98 73]	[98 67] [71 100]

Comparación balanceado con No balanceado

Cuando estamos en los modelos balanceados no existe la discriminación de ninguna de las clases, y aunque si que obtenemos mejores resultados para la clase 1 en ambos casos (SVC y DT), esto implicará que la clase 0 se vea afectada , obteniendo un peor valor en el balanceado respecto al no balanceado

Podemos ver como el modelo balanceado de Decision Tree obtiene unos valores más equilibrados y un poco mayores de recall y de accuracy que el SVC balanceado, además obtiene una cantidad mínimamente inferior de falsos positivos y negativos, por el que el modelo que sería más recomendable para clasificar ambas clases sería el Decision Tree.

[EX13]

Comparando con los resultados obtenidos en los apartados anteriores podemos ver que obtenemos una mayor cantidad de falsos negativos y positivos, además podemos ver como el precisión y recall esta mas desequilibrado respecto a los modelos anteriores y la accuracy es menor

	precision	recall	f1-score	support
0	0.53	0.84	0.65	187
1	0.66	0.29	0.40	196
accuracy			0.56	383
macro avg	0.59	0.57	0.53	383
weighted avg	0.60	0.56	0.52	383

Accuracy: 0.5587467362924282

Confusion Matrix:
[[158 29]
[140 56]]

[EX14]

Vemos como ambas clases están clasificando con una precisión y recall muy similares en ambas clases y mayores respecto a los ejercicios anteriores, también se puede observar como la accuracy es mayor a los casos anteriores . Además, observando la confusión matrix podemos ver que la cantidad de falsos positivos y negativos es menor también, por lo que podemos concluir que con este modelo obtenemos mejores resultados.

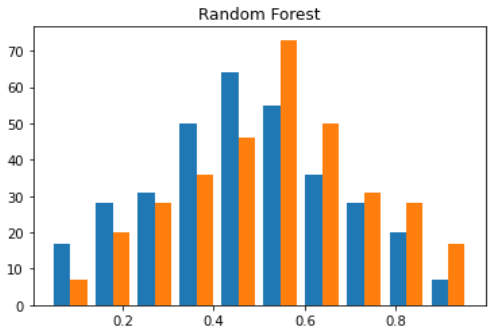
	precision	recall	f1-score	support
0	0.62	0.61	0.62	187
1	0.64	0.65	0.64	196
accuracy			0.63	383
macro avg	0.63	0.63	0.63	383
weighted avg	0.63	0.63	0.63	383

Accuracy: 0.6318537859007833

Confusion Matrix:
[[115 72]
[69 127]]

[EX15]

El histograma obtenido de las probabilidades resultantes de la predicción del modelo Random Forest para clase 0 y clase 1 es el siguiente:



[EX16]

Comparando con los valores obtenidos en cada uno de los apartados anteriores podemos ver como obtenemos unos valores equilibrados y muy parecidos a los del ejercicio 14 en cuanto a precision, recall y accuracy.

	precision	recall	f1-score	support
0	0.61	0.56	0.58	187
1	0.61	0.65	0.63	196
accuracy			0.61	383
macro avg	0.61	0.61	0.61	383
weighted avg	0.61	0.61	0.61	383

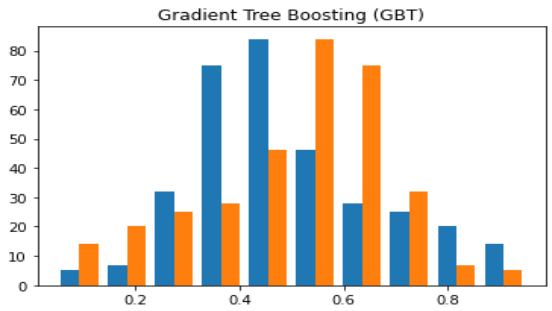
Accuracy: 0.608355091383812

Confusion Matrix:
[[105 82]
[68 128]]

Por otro lado, podemos ver que en este apartado tenemos mas casos donde encontramos un falso positivo, es decir, hemos clasificado a un non-customer como customer, lo que nos indicaría una mayor cantidad de posibles futuros clientes.

[EX17]

Comparando el histograma con el que hemos obtenido en el ejercicio 15 podemos observar como en el caso del GTB tenemos que separa las clases 0 y 1 mejor, ya que se puede apreciar como las distribuciones de las probabilidades están más separadas



[EX18]

Observando los diferentes valores que hemos obtenidos, como nos interesa predecir mejor los non customers, para así saber que los falsos positvos serán más fiables, elegiremos un valor de cutoff de 0,5, que es donde obtenemos unos valores de precisión y recall más equilibrados

También se puede observar como el número de noncustomers que enviaremos al manager, con un cutoff de 0,5 es de 211 noncustomers.

[EX19]

Podemos observar como las variables con mayor importancia son:

- CNT_CB_DENSITY
- Mobile_potential
- Revenue

[EX20]

La variable target nos permite diferenciar los dos tipos de clientes:

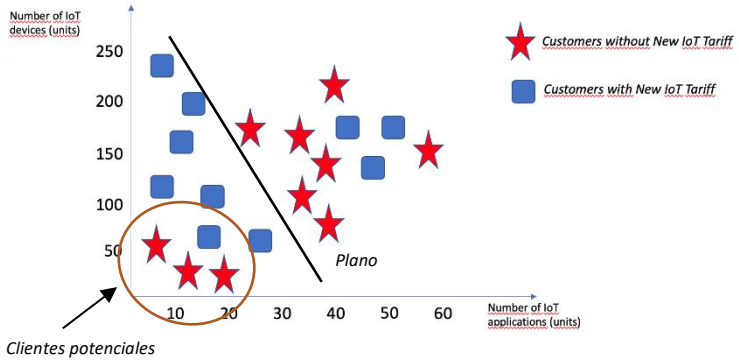
- Clientes que disponen de la nueva tarifa de IoT → representados en el gráfico con un cuadrado azul, diremos que pertenecen a la clase 1
- Clientes que no disponen de la nueva tarifa IoT → representados en el gráfico con una estrella roja, diremos que pertenecen a la clase 0

Si añadimos las variables data,voice consumption y mobile expense el coste computacional incrementará debido a que nuestro modelo deberá tener en cuenta estas variables a la hora de realizar el entrenamiento. Por lo que añadir o no estas variables dependerá de la información que aporten a nuestro modelo, si estas no aportan una gran información no merecerá la pena añadirlas , pero en el caso donde sí que aportaran información relevante , deberíamos sacrificar el coste computacional para obtener una mayor cantidad y calidad de información sobre los clientes. Otra idea sería si viéramos que alguna de estas nuevas variables aporta más información que alguna ya existente, también podríamos sustituirlas para que así el coste computacional se vea menos afectado.

Por otro lado, podemos observar como el dataset está equilibrado entre la cantidad de datos que pertenecen a la clase 1 (con la nueva tarifa IoT) y a la clase 0 (sin la nueva tarifa IoT) , teniendo la clase 0 una mayor parte de los datos, lo que nos dice que la mayoría de clientes no suelen contratar la tarifa móvil

Observando la representación de los datos en función del número de aplicaciones IoT y el número de dispositivos IoT podemos ver como que los clientes que tienen la tarifa móvil adquirida tienden a tener un menor número de aplicaciones IoT y unos valores similares de IoT devices

El plano que divide ambas clases será el siguiente:



Los clientes que será más conveniente llamar serían aquellos que han sido clasificados en la clase 1 siendo realmente de la clase 0 *(estrellas rojas por debajo del plano),* estos clientes tendrán características de número de aplicaciones y device IoT similares a los clientes que realmente si tienen la nueva tarifa IoT, por lo que la probabilidad de que adquieran la tarifa es mayor.

Para calcular la precisión y el recall nos ayudaremos de la matriz de confusión:

		0	**1**	**Total**	
True Negative	**0**	7	3	*10*	*False Positive*
False Negative	**1**	3	7	*10*	*True Positive*
	Total	*10*	*10*	*20*	

Podemos calcular el recall y la precision de la siguiente manera:

Recall de 1 $= \frac{True\ Positive}{True\ Positive + False\ Positive} = 7/10$

Recall de 0 $= \frac{True\ Positive}{True\ Positive + False\ Positive} = 7/10$

Precisión de 1 $= \frac{True\ Positive}{True\ Positive + False\ Negative} = 7/10$

Precisión de 0 $= \frac{True\ Positive}{True\ Positive + False\ Negative} = 7/10$

I hereby declare that, except for the code provided by the course instructors, all my code, report, and figures were produced by myself.