**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

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 | 1621,0556… |
| Q2 | 2200000 | 1948.4376… |
| Q3 | 4195000 | 2116.4740… |

Non-Customers

|  |  |  |
| --- | --- | --- |
|  | **Revenue** | **Mobile Potential** |
| Q1 | 902986 | 1513.3835… |
| Q2 | 1750000 | 1797.0542… |
| Q3 | 3501123.5 | 2035.0828… |

**[EX5]**

En la siguiente tabla podremos ver los resultados obtenidos:

|  |  |
| --- | --- |
| **Customer** | **Non-Customer** |
| **München 0.025060**  Köln 0.020286  Chemnitz 0.019093  Frankfurt 0.014320  Stuttgart 0.013126  ...  Enger 0.001193  Pfullingen 0.001193  Gilching 0.001193  Neu-Isenburg 0.001193  Wedemark 0.001193  Name: City, Length: 503,  dtype: float64 | **Köln 0.016139**  Stuttgart 0.010139  Bremen 0.010139  Dortmund 0.009518  Frankfurt 0.008897  ...  Bergtheim 0.000207  Atzelgift 0.000207  Salzbergen 0.000207  Bonstetten 0.000207  Solms 0.000207  Name: City, Length: 2110,  dtype: float64 |

Se puede apreciar como en el caso de los customers la ciudad con mayor ratio es München con 0.025060 , mientras en el caso de los Non-Customers es Köln con 0.016139

**[EX6]**

Hemos obtenido los siguientes valores con un valor de test\_size de 0,20:

* Longitud de X\_train: 4536
* Longitud de X\_test: 1135
* Longitud de final\_dataset (20%): 1134.2
* Longitud de final\_dataset (80%): 4536.8

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 como 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 , concretamente podría afectar a la precisión, recall y accuracy.

**[EX8]**

Como he comentado en el apartado anterior, el algoritmo está clasificando los datos de los diferentes datasets de manera muy desbalanceada, esto es debido a que en la clase 1 tenemos muchos menos datos que en la clase 0, para que estuvieran bien balanceados deberian tener una cantidad de datos similares.

En nuestro caso, vemos como tenemos una accuracy muy grande para la clase 0 pero una precision y recall muy bajos para la clase 1

Con ayuda de la confusion matrix , tanto para el test como el train, podemos observar como para una clase 0 el algoritmo funciona muy bien mientras que para la otra no debido al desbalance de los datos comentado previamente.

EX 10 🡪 revisar std de SVM

Ex 11🡪 segunda opcion, generear mas muestras de customers?

**[EX12]**

Los resultados obtenidos han sido los siguientes:

|  |  |
| --- | --- |
| **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] |

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.

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 resultado 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 un poco mayores de recall y de accuracy que el SVC balanceado, por el que el modelo que sería más recomendable para clasificar ambas clases sería el Decision Tree.

**[EX13][CODE][REPORT] Build a voting ensemble formed by a SVM and Decision Tree and train it with the balanced training dataset. Calculate the precision, recall and confusion matrix of the new classifier. Is it better than any of the previous baseline models? Justify your answer.**

El resultado obtenido es el siguiente:

**precision**  **recall** f1-score support

0 **0.54 0.86** 0.66 165

1 **0.68 0.29** 0.40 171

accuracy 0.57 336

macro avg 0.61 0.57 0.53 336

weighted avg 0.61 0.57 0.53 336

**Accuracy: 0.5684523809523809**

**Confusion Matrix: [142 23]**

**[122 49]**

Comparando con los resultados obtenidos en los apartados anteriores podemos ver que obtenemos un recall bastante mayor en la clase 0

Cual es mejor?

**[EX14]**

El resultado obtenido es el siguiente:

**precision recall** f1-score support

0 **0.65 0.56** 0.60 165

1 **0.62 0.70** 0.66 171

accuracy 0.63 336

macro avg 0.64 0.63 0.63 336

weighted avg 0.64 0.63 0.63 336

**Accuracy: 0.6339285714285714**

**Confusion Matrix: [93 72]**

**[51 120]**

Vemos como ambas clases están clasificando con una precision y recall muy similares, también se puede observar como la accuracy es mayor a los casos anteriores , por lo que podemos concluir que con este modelo obtenemos mejores resultados.

**[EX15][REPORT] Plot the histograms of the probabilities resulting of the prediction of the Random Forest model for class 0 and class 1.**

Gráfico, Gráfico de barras

Descripción generada automáticamente

Explicacion?

**[EX16][CODE][REPORT] Build a Boosting ensemble based on Gradient Tree Boosting (GBT). There are several boosting algorithms as Adaboost, etc. Train the GBT with the balanced training dataset, i.e. X\_train and y\_train. Calculate the precision, recall and confusion matrix of the new classifier. Is it better than any of the previous baseline models? Justify your answer.**

El resultado obtenido es el siguiente:

**precision recall** f1-score support

0 **0.65 0.53** 0.58 165

1 **0.61 0.73** 0.66 171

accuracy 0.63 336

macro avg 0.63 0.63 0.62 336

weighted avg 0.63 0.63 0.62 336

**Accuracy: 0.6279761904761905**

**Confusion Matrix: [87 78]**

**[47 124]**

Cual es mejor? Similitud al Random Forest

**[EX17][REPORT] Plot the histograms of the probabilities resulting of the prediction of the GBT model for class 0 and class 1. Compare it with histogram of Random Forest. Which one classifies better from your point of view? Why?**

Gráfico, Gráfico de barras

Descripción generada automáticamente  
Comparación con Random forest, cual clasifica mejor ?

**[EX18][CODE][REPORT] Execute the prediction for the selected model. Adjust the cutoff value to optimize the classifier if you consider necessary. How many non customers are you going to send to the sales managers to sell our products to them?**

**[EX19][CODE][REPORT] Order the features by importance. Which are the top 3 features to discriminate between non customers and customers?**

Podemos observar como el orden de importancia de las variables para discriminar entre customers y non customers es el siguiente:

1. **City\_coded**
2. **Mobile\_potential**
3. **Revenue**
4. CNT\_CB\_FN\_DENS
5. CNT\_CB\_MOB\_DENSITY
6. Sector
7. CNT\_EMPLOYEE
8. Legal\_Form\_Code
9. CNT\_CB\_DENSITY

Vemos como en las 3 variables que más importancia tienen son City\_coded, Mobile\_potential y revenue

CANVIAAA cuando ejecutas ?

**[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 una estrella azul, diremos que pertenecen a la clase 1
* Clientes que no disponen de la nueva tarifa IoT 🡪 representados en el gráfico con un círculo naranja, diremos que pertenecen a la clase 0

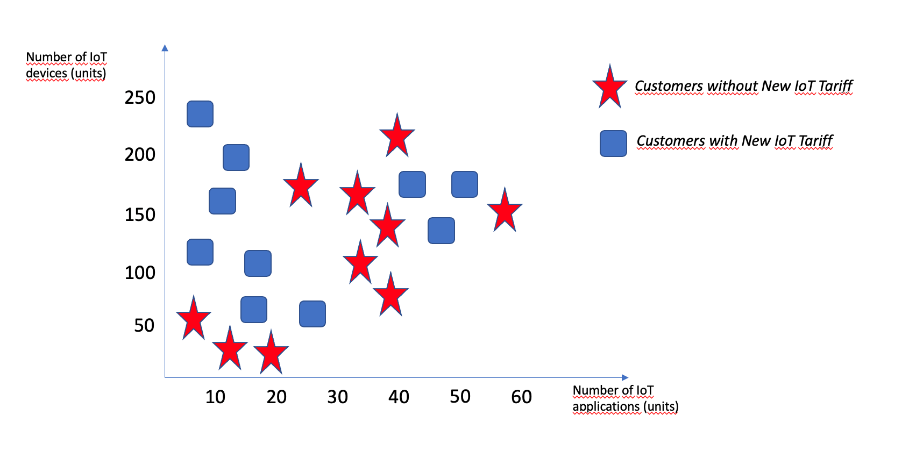
Si añadimos las variables data,voice consumption y mobile expense el coste computacional incrementará debido a que nuestro modelo deberá tener en cuenta estas variables a la hora de realizar el entrenamiento. Por lo que añadir o no estas variables dependerá de la información que aporten a nuestro modelo, 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 substituirlas 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 0 (con la nueva tarifa IoT) y a la clase 1 (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

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 numero 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.

El plano que divide ambas clases será el siguiente:



*Clientes potenciales*

*Plano*

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **0** | **1** |  |  |
| *True Negative* | **0** | 32 | 4 | *36* | *False Positive* |
| *False Negative* | **1** | 2 | 9 | *11* | *True Positive* |
|  |  | *34* | *13* | *47* |  |

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

***Recall de 1*** *=**9/11* ***Recall de 0*** *= 32/36*

***Precisión de 1*** *= 9/13* ***Precisión de 0*** *= 32/34*

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