Aprendizaje Automático Proyecto Final

Andrés Racines Bolaños.
Maestría En Ingeniería
Pontificia Universidad Javeriana
Cali-Colombia
Email:racines@javerianacali.edu.co

***Resumen*- El presente es el proyecto final del curso de Aprendizaje Automático, El objetivo del proyecto es resolver un problema de aprendizaje automático con el uso de 4 técnicas estudiadas a lo largo del curso y comparar el desempeño de las diferentes técnicas empleadas.**

I. Descripción de Problema

Para la implementación del proyecto final se seleccionó uno de los conjunto de datos del repositorio UCI https://archive.ics.uci.edu/ml/datasets/Bank+Marketing# este conjunto de datos está relacionado con campañas de marketing de institución bancaria portuguesa con datos tomados entre Mayo 2008 a Noviembre 2010, estas campañas están basadas en información obtenidas de llamadas telefónicas, dichas campañas a menudo requerían más de un contacto con el cliente para saber si accede a un crédito bancario, la clase objetivo es conocer si accede o no al crédito, se describe como un tarea de clasificación en la cual se emplearan dos conjuntos de datos (bank-full.csv con 45.211 instancias y 17 atributos y bank-full-adicion.csv con 41.189 instancias y 21 atributos), la ficha técnica se muestra a continuación.

Data Set Characteristics:	Multivariate	**Number of Instances:**	45211	**Area:**	Business
Attribute Characteristics:	Real	**Number of Attributes:**	17	**Date Donated**	2012-02-14
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	707414

Adicional a la tarea de clasificación existe un objetivo implícito en el planteamiento del problema, por tal razón el autor incluyen dos conjuntos de datos, uno con 17 atributos y 45.211 instancias "bank-full.csv" y otro con 41.189 instancias y 21 atributos de los cuales 17 atributos son similares a conjunto original contienen información relación con el banco y cliente, los 4 atributos adicionales del conjunto de datos "bank-full-adicion" atributos del contexto social y económico. El objetivo implícito es conocer con cuál de los dos conjunto de datos usando las mismas técnicas de clasificación obtiene el mejor resultado en la clasificación.

II. Pre-procesamiento de Datos

Las actividades de pre-procesamiento se realizaron de igual forma con los dos conjunto de datos descargados del repositorio UCI, los conjunto de datos son (bank-full.csv y bank-full-adicion.csv).

Primero se realiza el cargue de los datos y se verifica la correcta conversión atributos multivariados en numéricos.

```
ds_ = pd.read_csv('bank-full.csv', sep=';', na_values=" ?" )# se carga el cojunto de datos csv
ds = ds_.dropna() # se elimina las filas/columnas con valores null
ds = ds.rename(columns = {ds.columns.values[16]:'C'}) # renombramos la ultima columna por "C" de clase
le = preprocessing.LabelEncoder() # Label encoder de sci-kit
ds = ds.apply(le.fit_transform) # Convertimos los valores de object a numericos
ds.head() # se verifica el resultado de cargue de los datos
```

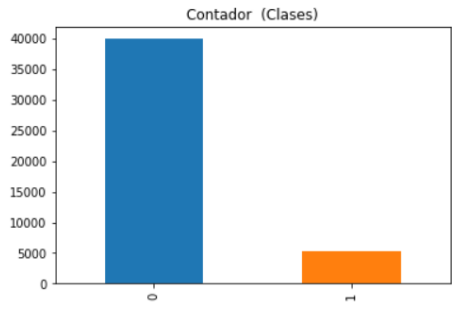
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	C
0	40	4	1	2	0	3036	1	0	2	4	8	261	0	0	0	3	0
1	26	9	2	1	0	945	1	0	2	4	8	151	0	0	0	3	0
2	15	2	1	1	0	918	1	1	2	4	8	76	0	0	0	3	0
3	29	1	1	3	0	2420	1	0	2	4	8	92	0	0	0	3	0
4	15	11	2	3	0	917	0	0	2	4	8	198	0	0	0	3	0

Una vez cargado el conjunto de datos original se procede a dividir el conjunto de datos de la siguiente forma
train 70% = 31.647 de conjunto de datos original
test 30% = 13.564 de conjunto de datos original
test-validación 10% = 4.522 de conjunto de datos original, este conjunto es utilizado para ajuste de configuración de algunas técnicas de aprendizaje automático las cuales tienen un alto costo computacional como las SVM y MLP.

```
Conteo de clases conjunto train Counter({0: 27909, 1: 3738})
Conteo de Clases conjunto test  Counter({0: 12013, 1: 1551})
Conteo de Clases conjunto train Validacion  Counter({0: 35939, 1: 4750})
Conteo de Clases conjunto test Validacion   Counter({0: 3983, 1: 539})
```

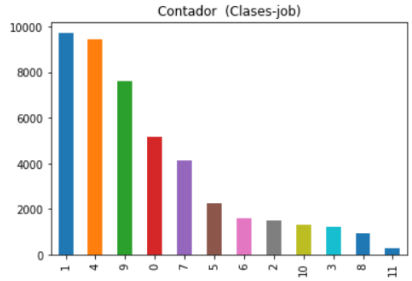
Se realiza un histograma para conocer la distribución y proporción de las clases.

```
Clase 0: 39922
Clase 1: 5289
Proporcion: 7.55 : clase[0] / clase[1]
```

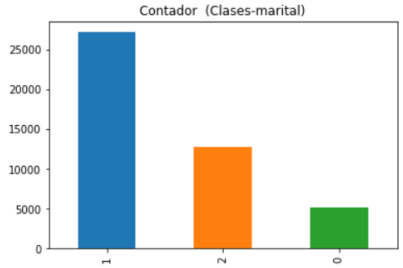


Se verifica que en todo el cojunto de datos no se presenten atributos con valores atipicos "outliers", a continuacion se incluyen algunas graficas de este proceso.

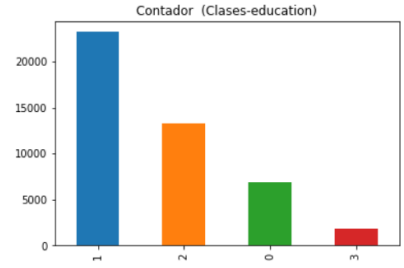
```
Clase 0: 5171
Clase 1: 9732
Proporcion: 0.53 : clase[0] / clase[1]
```




```
Clase 0: 5207
Clase 1: 27214
Proporcion: 0.19 : clase[0] / clase[1]
```



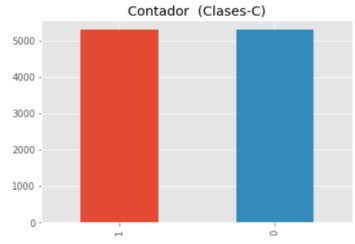

```
Clase 0: 6851
Clase 1: 23202
Proporcion: 0.3 : clase[0] / clase[1]
```



Como se puede observar en el primer histograma la clase objetivo esta desbalanceada la clase (0) tiene 39.922 instancias mientras que la clase (1) tiene 5.289 instancias, se realizaron iteraciones con el conjunto de datos en desbalance y las cuatro técnicas seleccionadas y se pudo constatar que esta que la condición del desbalance entre las clases afecta de forma drástica la capacidad de generalización de los diferentes clasificadores probados. Por tal motivo se procedió a balancear los las clases usando la técnica de under-sampling en el conjunto original,

El under-sampling consiste en igualar la clase mayoritaria a la clase minoritaria, a continuación se presenta el grafico y distribución de las clases después de aplicar el under-samplig:

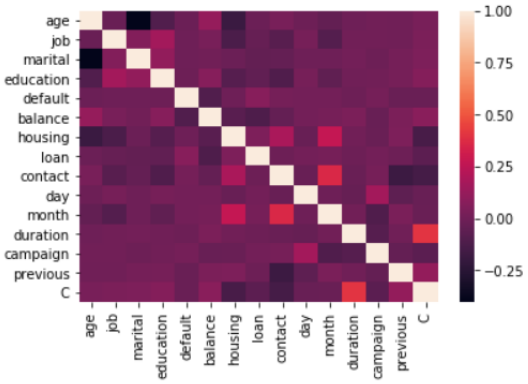
```
Clase 0: 5289
Clase 1: 5289
Proporcion: 1.0 : clase[0] / clase[1]
```



Nuevamente se realiza la division de los datos entre (train, test, validacion)

```
Conteo de clases conjunto train Counter({1: 3721, 0: 3683})
Conteo de Clases conjunto test  Counter({0: 1606, 1: 1568})
Conteo de Clases conjunto train Validacion  Counter({0: 3333, 1: 3330})
Conteo de Clases conjunto test Validacion   Counter({1: 391, 0: 350})
```

Se procede a realizar una matriz de correlación con el propósito de evidenciar la presencia de atributos con alta correlación los cuales podrían ser candidatos a eliminarse en una siguiente etapa del experimento. Como se puede observar el atributo “duration” y la clase “C” altamente correlacionados.



Selección de técnicas

Dado que la tarea es una tarea de clasificacion es binaria, la cual no es separable linealmente, se seleccionaron 4 cuatro tecnicas de aprendizaje automatico (SVM, K-NN, RNA y Arboles de decisión). EL experimento 1 consiste en probar las 4 tecnicas con el conjunto de datos “bank-full” de 45.211 instancias y 17 atributos, que luego de realizar el under sampling quedo con 10.578 instancias y 17 atributos.

El experimento 2 probar el conjunto de datos “bank-full-adcion” con 41.189 instancias y 21 atributos, los cuatro atributos adicionales corresponde a características socio económico de los clientes. Se probaran las mismas técnicas en los dos experimentos con las mismas configuraciones.

III. Experimento 1

En el experimento 1 se probaron las 4 técnicas de clasificacion, con el conjunto de datos “bank-full” de 10.578 instancias y 17 atributos, durante esperimento se realizaron varias iteraciones probando diferentes configuraciones y tecnicas que condujeron a mejorar cada una de las prediciones, solo se presentaran para el proyecto los mejores resultados obtenidos, se realizaran algunos comentarios de cada tecnica probada.

IV. Técnica de Clasificación # 1.1 (SVM)

Debido al alto costo computacional de esta técnica se utilizó los conjuntos de datos train.validacion y test-

validacion para el ajuste de parametros como el costo y kernel. Para esta técnica se realizaron dos iteraciones una con el conjunto de datos desbalanceado y otra con el conjunto de datos balanceado, primero se presentan los resultados obtenidos con el conjunto desbalanceados 45.211 instacias, {train=31.647 donde clase (0) = 27.909 y la clase (1) = 3.738}, {test=13.569 clase(0) = 12.031, clase(1) = 1.551} los resultados se presentan a continuacion:

```
############## Accuracy Linear SVM FINAL ##################
0.8903715718077263
[[11833   180]
 [ 1307   244]]
              precision    recall  f1-score   support

           0       0.90      0.99      0.94     12013
           1       0.58      0.16      0.25      1551

   micro avg       0.89      0.89      0.89     13564
   macro avg       0.74      0.57      0.59     13564
weighted avg       0.86      0.89      0.86     13564
```

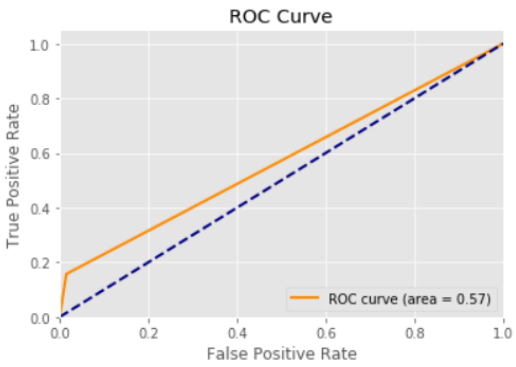
Inicialmente se puede observar que el accuracy llego hasta el 89% con lo cual se podría pensar que el clasificador tiene buena capacidad de generalización, lo cual no es cierto porque el accuracy solo esta indicando los elementos (TP+TN/VP+VN+FP+FN) que fueron bien clasificados y como en este caso la clase mayoritaria es 8 veces la clase minoritaria, dicha metrica no es suficiente para evaluar el clasificador, es necesario mirar otras metricas como la precision, recall y f1-score.

Al observar la precision se puede detallar que para la clase (0) se alcanzo el 90% precision. mientras que para la clase (1) el 58%, la precision se puede interpretar como cuantos valores verdaderos positivos (VP/VP+FP) fueron bien clasificados.

El recall alcanzo para la clase (0) 99% mientras que para la clase (1) 16%, el recall es conocida como la tasa o sensibilidad positiva verdadera.

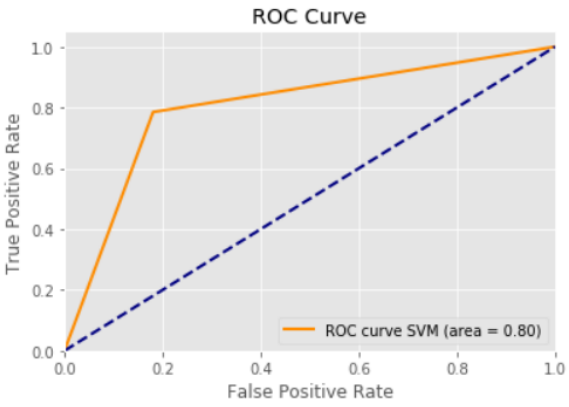
Otra métrica relacionada es la F1-score con 94% para clase (0) y 25% para la clase (1), en estas metrica se toma encuenta tanto la precision como el recall. El F1-score es la media armónica de las dos metricas y se calcula como: F1 = 2 (precisión x recall) / (precisión + recall). El puntaje de F1-score es la manera de resumir la clasificación de la precision y el recalla en un solo número.

También se inspecciono la tasa de positivos verdaderos en comparación con la tasa de falsos positivos, en la curva de característica de operación del receptor (ROC) y el valor correspondiente del área bajo la curva (AUC). Cuanto más cerca esté esta curva de la esquina superior izquierda, mejor será el rendimiento del clasificador (es decir, maximizar la tasa de verdaderos positivos y minimizar la tasa de falsos positivos). Las curvas que están cerca de la diagonal de la trama, resultan de clasificaciones que tienden a hacer predicciones que están cerca de adivinar al azar.



Ahora se presentan los resultados despues de la under-sampling con las clases balanceadas, el archivo original luego de igualar la clase mayoritaria a la minoritaria quedo de 10.578 instancias y 17 atributos. Distribuidos de la siguiente forma {train= 7.404 clase (0) 3721 , clase (1) 3.683}, {test=3174 clase(0) 1.606 clase (1) 1.568}

```
############## Accuracy Linear SVM FINAL ##################
0.8024574669187146
[[1316  290]
 [ 337 1231]]
              precision    recall  f1-score   support

           0       0.80      0.82      0.81      1606
           1       0.81      0.79      0.80      1568

   micro avg       0.80      0.80      0.80      3174
   macro avg       0.80      0.80      0.80      3174
weighted avg       0.80      0.80      0.80      3174
```



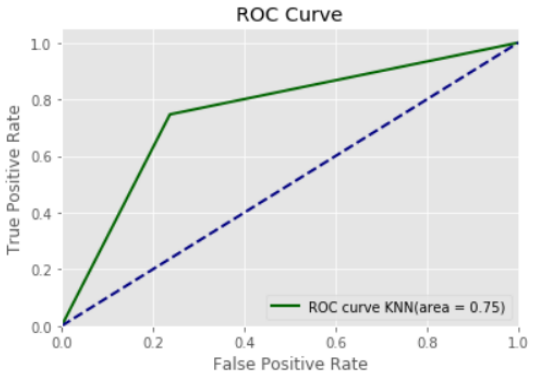
Como se puede apreciar el modelo luego de under-samplig tiene mayor capacidad de generalización y área bajo la curva ROC es del 80%, mientras que la medida del accuracy bajo de 89% a 80%, las demás métricas se igualaron a valores cercanos al 80% . Lo cual indica que el modelo está generalizando bastante bien.

Dados los resultados de las metricas obtenidas luego del under-sampling se aplicara este mismo conjunto de datos balanceado a las demas metricas para observar la capacidad de generalización, se incluira por cada técnica las metricas (precision, recall, F1-score) y la curva ROC, para comparar el comportamiento de las demas técnicas, al finalizar el experimento 1 se incluira una gráfica con la integración de todas las curvas ROC para apreciar cual de las técnicas de clasificacion presento mejor desempeño.

V. Técnica de Clasificación # 1.2 (K-NN)

Datos original 10.578 instancias y 17 atributos. Distribuidos de la siguiente forma {train= 7.404 clase (0) 3721 , clase (1) 3.683}, {test=3174 clase(0) 1.606 clase (1) 1.568}

```
############## Accuracy KNN ##################
0.7548834278512917
[[1225  381]
 [ 397 1171]]
              precision    recall  f1-score   support

           0       0.76      0.76      0.76      1606
           1       0.75      0.75      0.75      1568

   micro avg       0.75      0.75      0.75      3174
   macro avg       0.75      0.75      0.75      3174
weighted avg       0.75      0.75      0.75      3174
```

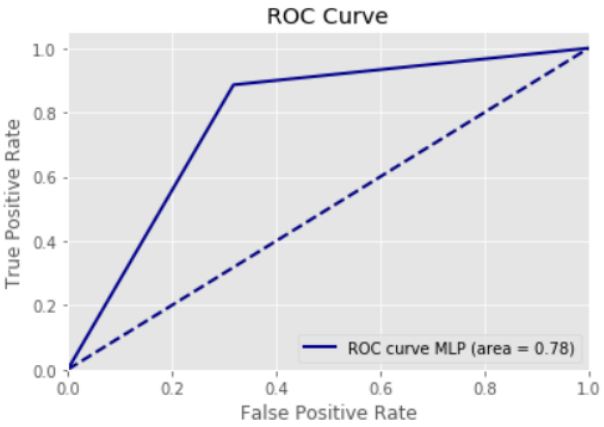


Como se puede observar en la curva ROC, esta técnica obtuvo un área bajo la curva inferior al de la SVM.

VI. Técnica de clasificación # 1.3 (Redes Neuronales MLP)

Datos original 10.578 instancias y 17 atributos. Distribuidos de la siguiente forma {train= 7.404 clase (0) 3721 , clase (1) 3.683}, {test=3174 clase(0) 1.606 clase (1) 1.568}

```
############## Accuracy MLP ##################
0.782608695652174
[[1095  511]
 [ 179 1389]]
              precision    recall  f1-score   support

           0       0.86      0.68      0.76      1606
           1       0.73      0.89      0.80      1568

   micro avg       0.78      0.78      0.78      3174
   macro avg       0.80      0.78      0.78      3174
weighted avg       0.80      0.78      0.78      3174
```

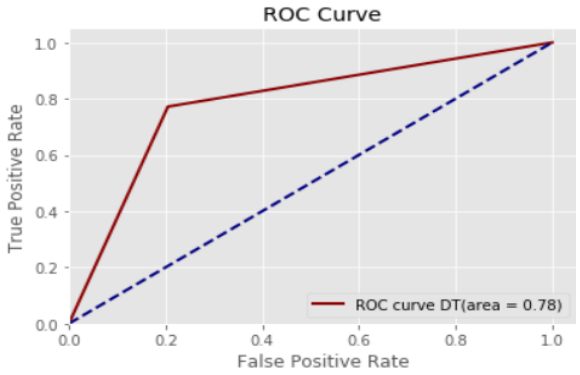


La curva ROC de MLP supera la técnica KNN pero no logra superar a la SVM y a pesar que pareciera que el área bajo la curva fuese mayor .

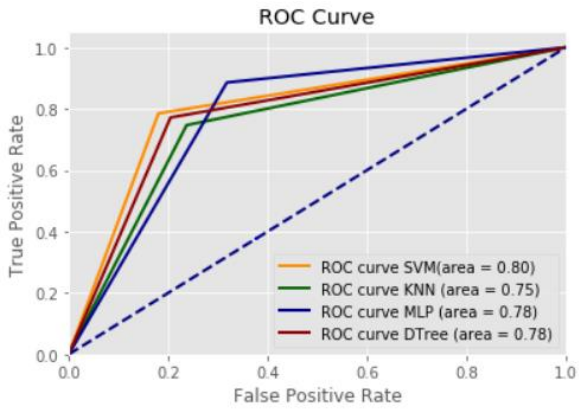
VII. Técnica de Clasificación # 1.4 (Arboles de Decisión)

Datos original 10.578 instancias y 17 atributos. Distribuidos de la siguiente forma {train= 7.404 clase (0) 3721 , clase (1) 3.683}, {test=3174 clase(0) 1.606 clase (1) 1.568}

```
############## Accuracy DT##################
0.7835538752362949

############## Matriz de Confusion DT ##################
[[1277  329]
 [ 358 1210]]

############## Metricas DT ##################
              precision    recall  f1-score   support

           0       0.78      0.80      0.79      1606
           1       0.79      0.77      0.78      1568

   micro avg       0.78      0.78      0.78      3174
   macro avg       0.78      0.78      0.78      3174
weighted avg       0.78      0.78      0.78      3174
```



Para finalizar este experimento se incluye una gráfica con las curvas ROC de las cuatro técnicas probadas para poder visualizar cuál de estas técnicas obtuvo el mejor resultado.



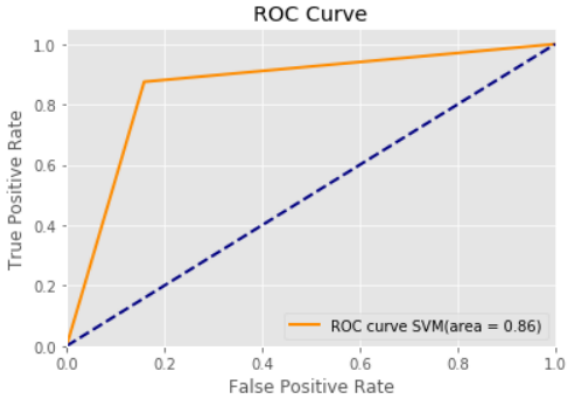
VIII. Experimento 2

El experimento 2 conjunto de datos "bank-full-adcion" con 41.189 instancias y 21 atributos, los cuatro atributos adicionales corresponde a características socio económico de los clientes. Se probarán las mismas técnicas que primer experimento.

IX. Técnica de Clasificación # 2.1 (SVM)

Conjunto de datos 6.496 instancias 21 atributos
train = (clase (1) : 3249, clase (0) : 3247),
test = (clase (0): 1393, clase (1): 1391)

```
############## Accuracy Linear SVM FINAL ##################
0.858117816091954
[[1172  221]
 [ 174 1217]]
              precision    recall  f1-score   support

           0       0.87      0.84      0.86      1393
           1       0.85      0.87      0.86      1391

   micro avg       0.86      0.86      0.86      2784
   macro avg       0.86      0.86      0.86      2784
weighted avg       0.86      0.86      0.86      2784
```

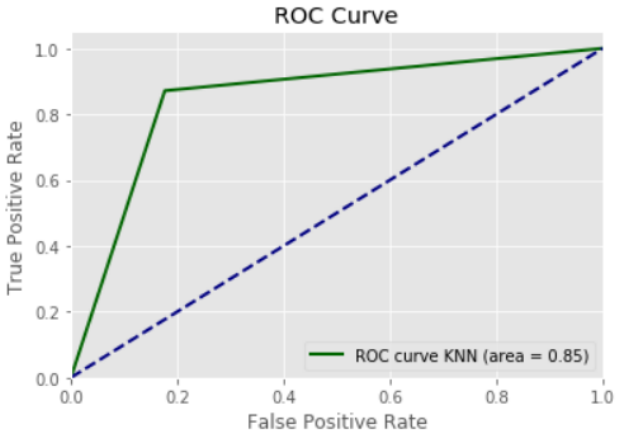


Se puede observar que las metricas en general, con la adición de los atributos socio económicos de los cliente mejora la capacidad de generalizacion de la técnica (SVM)

X. Técnica de Clasificación # 2.2 (K-NN)

Conjunto de datos 6.496 instancias 21 atributos
train = (clase (1) : 3249, clase (0) : 3247),
test = (clase (0): 1393, clase (1): 1391)

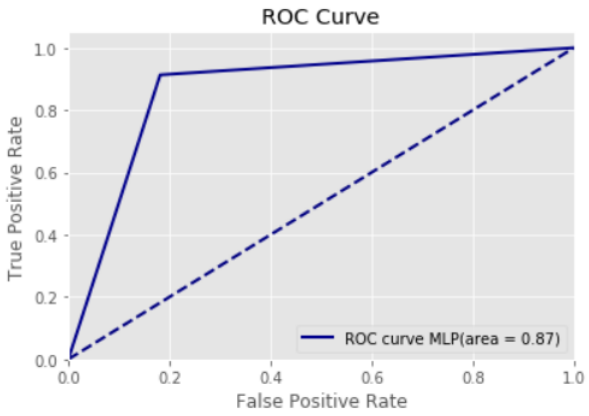
```
############## Accuracy KNN ##################
0.8473419540229885
[[1147  246]
 [ 179 1212]]
              precision    recall  f1-score   support

           0       0.87      0.82      0.84      1393
           1       0.83      0.87      0.85      1391

   micro avg       0.85      0.85      0.85      2784
   macro avg       0.85      0.85      0.85      2784
weighted avg       0.85      0.85      0.85      2784
```



XI. Técnica de Clasificación # 2.3 (MLP)

Conjunto de datos 6.496 instancias 21 atributos
train = (clase (1) : 3249, clase (0) : 3247),
test = (clase (0): 1393, clase (1): 1391)

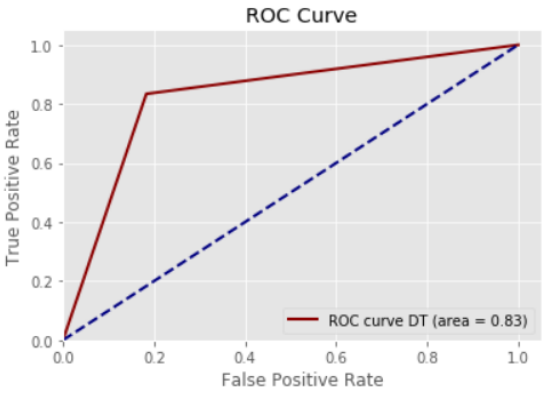
```
############## Accuracy MLP ##################
0.8656609195402298
[[1140  253]
 [ 121 1270]]
              precision    recall  f1-score   support

           0       0.90      0.82      0.86      1393
           1       0.83      0.91      0.87      1391

   micro avg       0.87      0.87      0.87      2784
   macro avg       0.87      0.87      0.87      2784
weighted avg       0.87      0.87      0.87      2784
```



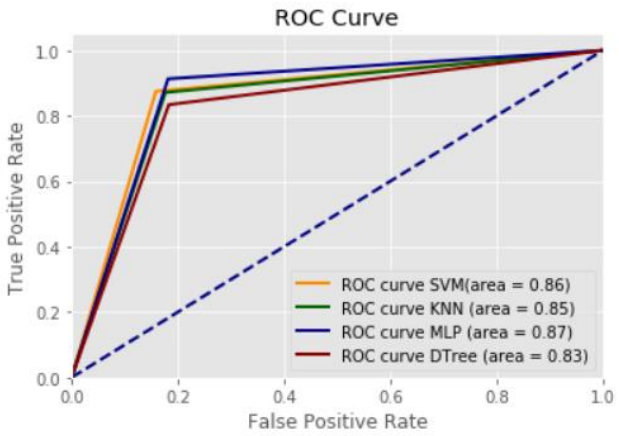
XII. TÉCNICA DE CLASIFICACIÓN # 2.4 (ARBOLES DE DECISIÓN)

Conjunto de datos 6.496 instancias 21 atributos
train = (clase (1) : 3249, clase (0) : 3247),
test = (clase (0): 1393, clase (1): 1391)

```
############## Accuracy DT##################
0.8254310344827587

############## Matriz de Confusion DT ##################
[[1138  255]
 [ 231 1160]]

############## Metricas DT ##################
              precision    recall  f1-score   support

           0       0.83      0.82      0.82      1393
           1       0.82      0.83      0.83      1391

   micro avg       0.83      0.83      0.83      2784
   macro avg       0.83      0.83      0.83      2784
weighted avg       0.83      0.83      0.83      2784
```



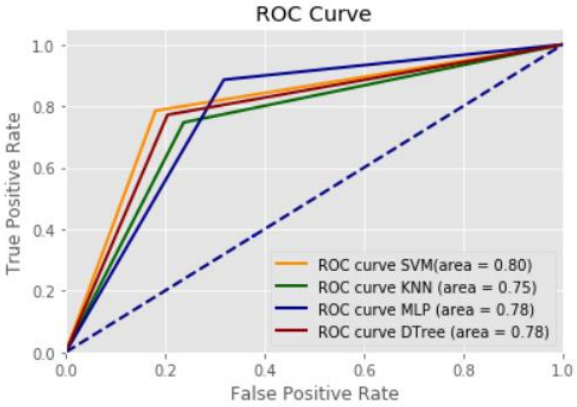
Para comparar cuál de los cuatro clasificadores empleados obtuvo el mejor desempeño, se empleara la curva ROC con las métricas de todos los clasificadores al tiempo.



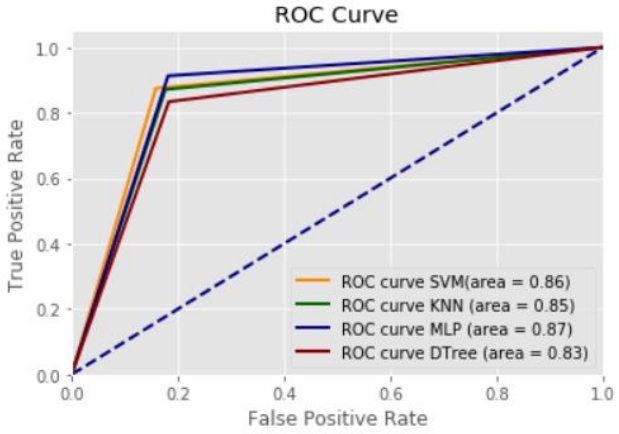
Como se puede observar en la gráfica la mejor generalización la realiza el MLP, seguido por SVM y KNN que tienen metricas similares, el clasificador que menor capacidad de generalización demostro fue DT.

XIII. COMPARACIÓN ENTRE EXPERIMENTOS

Resultados de curva ROC para el experimento 1

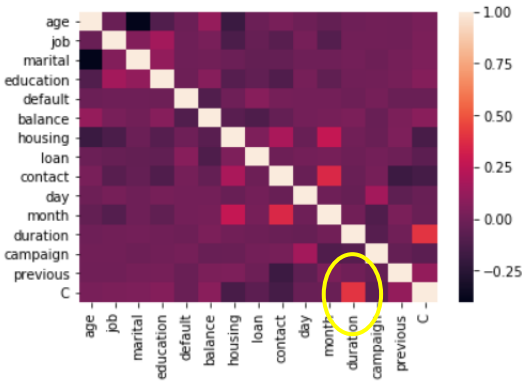


Resultados de curva ROC para el experimento 2

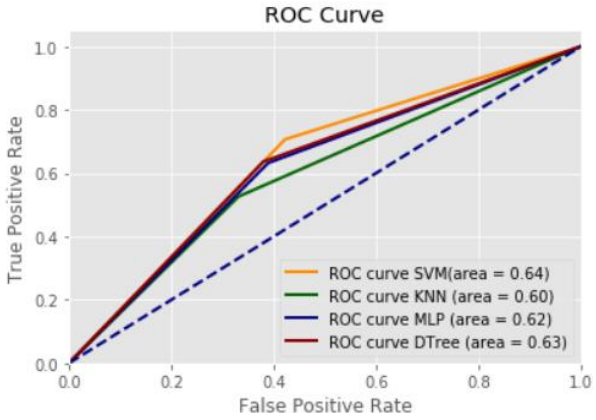


El desempeño de los clasificadores en el segundo experimento son superiores al primero, los cuatro atributos adicionales en el conjunto de datos del experimiento 2 (socio económicos) agregados a los registros permitieron obtener mejores resultados.

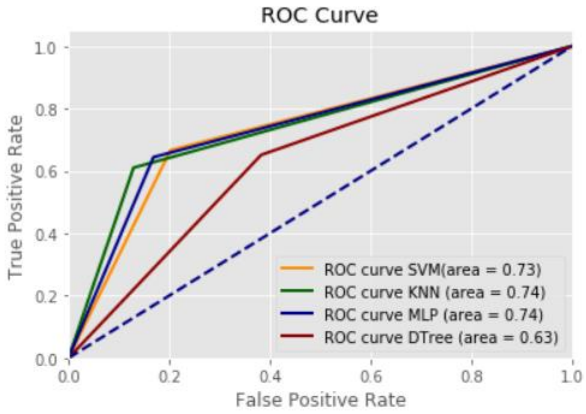
Se realizará una iteración mas para comparar el resultado de quitar el atributo “duration” que esta altamente relacionado con la clase objetivo, como se puede observar en la matriz de correlación de abajo.



Curva ROC experimento 1, luego de quitar el atributo “duraction”, 16 atributos en total, el redimiento disminuye en todos los clasificadores como se observa en la gráfica siguiente.



Curva ROC experimento 2, quitando el atributo “duraction”, en el conjunto con los datos socio economicos, 20 atributos en total.



Como se puede observar en las dos gráficas, el atributo “duraction” afecta la capacidad de generalización del primer experimento, mientras que para el segundo experimento la capacidad de generalización de los cuatro clasificadores no es efectado tan drasticamente como en el experimento 1, parece ser que los atributos socio económicos tiene una importancia igual o superior al atributo duration. El atributo duration ha sido ampliamente documentado en la literatura demostrando su importancia en los estudios de marketing, pese a esto los atributos socio económicos parecen tener igual o mayor importancia.

XIV. CONCLUSIONES

A pesar que la métrica de acurracy para los diferentes técnicas de clasificación fue alta en general, no es posible evaluar un modelo con tan solo esta métrica, es necesario verificar otras métricas como la precisión, recall, f1-core, puesto que el objetivo es mejorar la capacidad de generalización de las técnicas.

La implementación del balanceo de clases con under-sampling fue factor definitivo en la obtención de buenos resultados, esto permitió mejorar la capacidad de generalización de las técnicas. Cuando se trabaja con clases altamente desbalanceados se cae en el error de entrena un clasificador experto en predecir la clase mayoritaria.

Por ultimo la utilizacion de la curva ROC para interpretar la verdadera la capacidad de generalizacion de un modelo es un requerimiento necesario y utili a la hora de evaluar varias tecnicas como se observo en la ultima iteracion realizada.