



Instituto Tecnológico y de Estudios Superiores de Monterrey

TC4030.10 Inteligencia artificial y aprendizaje automático

Profesor titular: Dr. Luis Eduardo Falcón Morales

Profesor tutor: Jorge Luis Ramos Zavaleta

Nombre de la actividad: Actividad 10. Práctica: Bayes ingenuo y vecino más cercano

Nombre de los alumnos y matrícula:

Roberto Romero Vielma - A00822314

José Javier Granados Hernández - A00566717

Eduardo Gabriel Arévalo Aguilar - A01793897

Ernesto Enriquez Rubio - A01228409

Fecha: 14/Noviembre/2022

Ejercicio 1

El objetivo del artículo publicado es ofrecer claridad y una base teórica sobre 18 métricas que se utilizan continuamente en la evaluación de modelos clasificadores de aprendizaje de máquina. Estos métricos son basados en umbrales y puntaje cualitativo, métricos basados en aprendizaje probabilístico y métricos que ponderan lo bien que se clasifican las muestras.

Los resultados muestran que las métricas miden cosas diferentes y que la decisión en base a uno puede cambiar si usamos otro como referencia. De las métricas evaluadas hay cinco que dependen de la probabilidad de que una muestra sea predicha a pertenecer a un conjunto, también sujetas a un umbral de dicha probabilidad; estas métricas son “accuracy”, “kappa statistic”, “f-measure”, “macro average arithmetic” y “macro average geometric”. El resto de las métricas en el artículo no son dependientes de este término.

Por otro lado, las métricas que no son dependientes del umbral de probabilidad pueden dividirse en dos subconjuntos. Aquellas que se enfocan en “ranking” o separabilidad y aquellas enfocadas en probabilidades. Para evaluar estas métricas se realizaron una serie de experimentos usando a Witten y Frank (2005). Se usaron 6 algoritmos de aprendizaje de máquina. Los experimentos se realizaron dentro de 30 conjuntos de datos pequeños y medianos. Los problemas de clasificación eran 15 binarios y el resto de multiclase.

Los resultados muestran que hay métricos que proveen la misma información; es decir, que se pueden usar indistintamente uno de otro basados en la correlación que hay entre ellos. También podría decirse que gracias a los dendrogramas provistos podríamos seleccionar un conjunto de métricas mínimas para evaluar distintos modelos y compararlos entre sí. Para aprovechar al máximo esta herramienta, los modelos a comparar deberían clasificarse por balance de clases, número de clases; binaria o multiclase y tamaño del set de datos.

También se prueban las métricas al sesgar las probabilidades de clasificación, de manera que sea más difícil clasificar. Las métricas se comportan de manera diferente, ahora si estos datos se cambian con un método de swap de métodos consecutivos, se modifican también las métricas, pero la diferencia entre métricas es más marcada en el primer método que en este último. Se experimenta también con el set de datos al remover gradualmente las clases del modelo. De modo que se aprecia la variabilidad de cada una de las métricas en función al número de clases removidas. Las conclusiones más relevantes en este ejercicio es que la métrica de “Accuracy” es la más útil para medir el rendimiento de un modelo de clasificación de una o múltiples clases. Le siguen de cerca las métricas de MSE, MAE, MPR y LogL.

Ejercicio 2

- **Descripción del problema y el objetivo que se busca**

El problema que se tiene en el Breast Cancer Wisconsin (Diagnostic) Data Set, consiste en que se tiene información respecto a pruebas de biopsia con aspiración de aguja fina, los datos de cada estudio se convirtieron en columnas con información de los núcleos celulares del paciente, además de incluir el diagnóstico del paciente, representado con una M (Maligno) o B (Benigno), por lo que el problema que se tiene en este dataset, es saber si el tumor encontrado dentro del seno es maligno o benigno, por lo que el objetivo es generar un modelo de clasificación, que pueda predecir si el tumor cae dentro de la categoría M o B, a través del uso de las columnas que contienen la información de los núcleos celulares.

- **Justificación de si el problema requiere del uso de técnicas de balanceo de clases**

Dado que la clase B presenta un 63% de los casos, mientras que la clase M cuenta con el 37% restante, queda en evidencia que las clases no requieren de una técnica de balanceo de clases, dado que no se cuenta con la proporción 70-30 para considerar que las clases están desbalanceadas, la cantidad de casos en ambas clases es lo suficiente grande como para poder predecir de manera correcta si una persona tiene cáncer o no.

- **Análisis de las implicaciones que conllevan los FN y FP**

Un falso positivo implica que una persona que no tiene cáncer sea erróneamente diagnosticada con cáncer, por otro lado un falso negativo significa que una persona con cáncer ha sido erróneamente diagnosticada como libre de cáncer, cualquiera de los 2 resultados implicaría el mismo riesgo para un paciente, dado que al no tratarse una enfermedad tan grave como lo es el cáncer implicaría que el paciente empiece a desarrollar la enfermedad en más partes de su cuerpo, al grado de que en el algún momento, el paciente ya no podrá recurrir a tratamientos para combatir el cáncer, mientras que los pacientes que no tengan cáncer y sean expuestos a quimioterapias pueden desarrollar a largo plazo cáncer, además de que afectaría su salud mental con enfermedades crónicas como la depresión, lo que podría llevar al paciente a desarrollar otras enfermedades por su sistema inmunológico debilitado, a raíz de creer que tiene cáncer y de los tratamientos contra el cáncer.

- **Justificación de la métrica a utilizar**

Lo que hicimos en las métricas fue obtener el número mínimo de métricas necesarias para evaluar el modelo, en base a las métricas utilizadas en el paper de C. Ferri, 2009. Utilizamos los dendrogramas para agrupar los métricos que tenían correlaciones mayores al 90%. De ésta manera teníamos grupos representando dendrogramas y subgrupos representando un conjunto de métricos con correlación entre ellos de encima del 90%.

De cada grupo (dendrograma) seleccionamos el que tenga el menor número de subconjuntos. Seleccionamos de ellos un métrico que represente dicho subconjunto; posteriormente, comparamos con el resto de grupos (dendrogramas) por los subconjuntos que necesiten un métrico para ser representados. De ésta manera nos aseguramos de que los métricos seleccionados sirvan para evaluar los agrupados de datos según sus características. Algoritmo codificado en Anexo A.

Características del agrupado de datos: Bi-clase, Clases balanceadas, Agrupado de datos pequeño y con ruido de probabilidad /traslape de datos.

Acrónimos de Métricas a utilizar: 'CALB', 'CALL', 'MPR', 'ACC', 'LOGL', 'MSE', 'AU1U' y 'SAUC'

Ejercicio 3

- **Descripción del problema y el objetivo que se busca**

El problema que presenta el South German Credit Data Set, es la selección de buenos candidatos para el otorgamiento de crédito, en base a factores como la edad, duración en los empleos, sexo, si cuenta con una hipoteca, entre otras características, el objetivo que se busca es desarrollar un modelo de clasificación que sea capaz de determinar si un cliente será bueno o malo para pagar el crédito y en base a ello poder tomar la decisión de si otorgarlo o negarlo.

- **Justificación de si el problema requiere del uso de técnicas de balanceo de clases**

Dado que la clase 1 representa el 70% y la clase 0 el 30% restante, el uso de técnicas de balanceo de clases serán requeridas, dado que la clase 0 no cuenta con suficientes muestras para poder predecir de forma correcta si un cliente cae en la categoría de bueno (1) o malo (0).

- **Análisis de las implicaciones que conllevan los FN y FP**

Las consecuencias de contar con una alta cantidad de falsos negativos, privará a la institución de otorgar a sus clientes con alta probabilidad de garantías de pago el otorgamiento de un crédito con sus respectivas ganancias, tanto monetarias como de imagen. Al mismo tiempo si los falsos positivos son altos las consecuencias son pérdidas financieras y crearían una imagen falsa ante sus potenciales clientes.

- **Justificación de la métrica a utilizar**

Lo que hicimos en las métricas fue obtener el número mínimo de métricas necesarias para evaluar el modelo, en base a las métricas utilizadas en el paper de C. Ferri, 2009. Utilizamos los dendrogramas para agrupar los métricos que tenían correlaciones mayores al 90%. De ésta manera teníamos grupos representando dendrogramas y subgrupos representando un conjunto de métricos con correlación entre ellos de encima del 90%.

De cada grupo (dendrograma) seleccionamos el que tenga el menor número de subconjuntos. Seleccionamos de ellos un métrico que represente dicho subconjunto; posteriormente, comparamos con el resto de grupos (dendrogramas) por los subconjuntos que necesiten un métrico para ser representados. De ésta manera nos aseguramos de que los métricos seleccionados sirvan para evaluar los agrupados de datos según sus características. Algoritmo codificado en Anexo B.

Características del agrupado de datos: Bi-clase, Clases desbalanceadas, Agrupado de datos pequeño y con ruido de probabilidad /traslape de datos.

Acrónimos de Métricas a utilizar: 'CALB', 'CALL', 'MPR', 'ACC', 'LOGL', 'MSE', 'AU1U', 'SAUC', 'PAUC', 'MAPR', 'MFM', 'MAVA' y 'MAVG'.

Ejercicio 4

- **Descripción del problema y el objetivo que se busca**

El problema presentado en el dataset llamado Telco customer churn: IBM dataset, es referente a la rotación de clientes de una empresa de telecomunicaciones ficticia, por lo que es imperativo conocer en base a ciertas características, si el cliente dejará de consumir nuestros servicios en base a factores como el método de facturación, la cantidad que paga por el servicio, entre otras, el objetivo que se busca es desarrollar un modelo que permita clasificar a clientes que piensan dejar el servicio durante un periodo (Q1, Q2, Q3, Q4) y en base a ello poder mandarle descuentos o promociones para que el cliente siga contratando el servicio y no pasarse a otro.

- **Justificación de si el problema requiere del uso de técnicas de balanceo de clases**

El problema requiere del uso de técnicas de balanceo de clases, dado que la clase positiva (clientes que dejaron el servicio) cuenta con el 73% de los casos, mientras que la clase negativa (clientes que siguen consumiendo el servicio) es del 26%, lo que indica que faltan más datos de la clase minoritaria para poder hacer mejores predicciones de dicha clase.

- **Análisis de las implicaciones que conllevan los FN y FP**

Los falsos negativos simbolizan clientes que dejaron el servicio, pero el modelo los clasificó como que no iban a dejar el servicio, por ende la compañía ha perdido clientes y requerirá más dinero para atraer a nuevos clientes, mientras que los falsos positivos indican que nuestro modelo ha clasificado mal a los clientes, que si van a continuar consumiendo el servicio y por lo tanto no requieren de ofertas y/o promociones en su suscripción, cualquiera de los 2 casos sería malo para la empresa, pero en particular los falsos negativos serán los más dañinos, dado que involucran la pérdida de clientes, y por ende se requiere invertir una mayor cantidad de dinero para conseguir nuevos clientes, cuando simplemente se pudo haber hecho una mejor oferta, a clientes actuales que pensaban cambiar de servicio.

- **Justificación de la métrica a utilizar**

Lo que hicimos en las métricas fue obtener el número mínimo de métricas necesarias para evaluar el modelo, en base a las métricas utilizadas en el paper de C. Ferri, 2009. Utilizamos los dendrogramas para agrupar los métricos que tenían correlaciones mayores al 90%. De ésta manera teníamos grupos representando dendrogramas y

subgrupos representando un conjunto de métricos con correlación entre ellos de encima del 90%.

De cada grupo (dendrograma) seleccionamos el que tenga el menor número de subconjuntos. Seleccionamos de ellos un métrico que represente dicho subconjunto; posteriormente, comparamos con el resto de grupos (dendrogramas) por los subconjuntos que necesiten un métrico para ser representados. De ésta manera nos aseguramos de que los métricos seleccionados sirvan para evaluar los agrupados de datos según sus características. Algoritmo codificado en Anexo C.

Características del agrupado de datos: Bi-clase, Clases desbalanceadas, Agrupado de datos grande y con ruido de probabilidad /traslape de datos.

Acrónimos de Métricas a utilizar: 'LOGL', 'AU1U', 'MPR', 'SAUC', 'ACC', 'CALB', 'MSE', 'CALL', 'MFM', 'MAVA' y 'MAVG'.

Ejercicio 5

- **Descripción del problema y el objetivo que se busca**

El problema con el agrupado de datos Qualitative Bankruptcy Dataset, contiene información de instituciones financieras. Con ésta información se pretende modelar la probabilidad de dichas instituciones de llegar a la bancarrota.

- **Justificación de si el problema requiere del uso de técnicas de balanceo de clases**

El problema requiere de balanceo de clases pues la proporción está sesgada favoreciendo el modelado para los casos que representan instituciones de no bancarrota. Por lo que es necesario usar técnicas de balanceo para refinar el modelo en casos de bancarrota.

- **Análisis de las implicaciones que conllevan los FN y FP**

Los falsos positivos en éste análisis, de llegar a ser usados en instituciones con una alta influencia; por ejemplo, en el mercado bursátil, costaría la desvalorización de las compañías en éste grupo. Por otro lado, los falsos positivos harían que las compañías en éste grupo estuvieran sobre valuadas. Esto costaría a los potenciales inversionistas una eventual caída en sus acciones.

- **Justificación de la métrica a utilizar**

Lo que hicimos en las métricas fue obtener el número mínimo de métricas necesarias para evaluar el modelo, en base a las métricas utilizadas en el paper de C. Ferri, 2009. Utilizamos los dendrogramas para agrupar los métricos que tenían correlaciones mayores al 90%. De ésta manera teníamos grupos representando dendrogramas y subgrupos representando un conjunto de métricos con correlación entre ellos de encima del 90%.

De cada grupo (dendrograma) seleccionamos el que tenga el menor número de subconjuntos. Seleccionamos de ellos un métrico que represente dicho subconjunto; posteriormente, comparamos con el resto de grupos (dendrogramas) por los subconjuntos que necesiten un métrico para ser representados. De ésta manera nos aseguramos de que los métricos seleccionados sirvan para evaluar los agrupados de datos según sus características. Algoritmo codificado en Anexo D.

Características del agrupado de datos: Bi-clase, Clases desbalanceadas, Agrupado de datos pequeño y con ruido de probabilidad /traslape de datos.

Acrónimos de Métricas a utilizar: 'CALB', 'CALL', 'MPR', 'ACC', 'LOGL', 'MSE', 'AU1U', 'SAUC', 'PAUC', 'MAPR', 'MFM', 'MAVA' y 'MAVG'.

Anexos

```
python.py

import numpy as np
# Sets de métricos por características:
# se agrupan entre ellos por correlación sobre el 90%
biClass = [
    ['CALB'],
    ['CALL'],
    ['MPR', 'MAE', 'SAUCC', 'PAUC', 'MAPR'],
    ['ACC', 'MAVG', 'MAVA', 'MFM', 'KAPS'],
    ['LOGL'],
    ['MSE'],
    ['AU1U', 'AUNU', 'AU1P', 'AUNP'],
]

multiclass = [
    ['MSE'],
    ['ACC', 'KAPS', 'MFM', 'MAVA', 'MAVG'],
    ['CALB'],
    ['MPR', 'MAE', 'SAUC', 'PAUC', 'MAPR'],
    ['CALL'],
    ['LOGL'],
    ['AU1U', 'AUNU', 'AU1P', 'AUNP'],
]

balancedClass = [
    ['LOGL'],
    ['AU1U', 'AUNU', 'AU1P', 'AUNP'],
    ['CALB'],
    ['SAUC', 'PAUC', 'MAPR', 'MPR', 'MAE'],
    ['CALL'],
    ['MSE', 'ACC', 'KAPS', 'MAVG', 'MFM', 'MAVA'],
]

imbalancedClass = [
    ['CALB'],
    ['MPR', 'MAE'],
    ['SAUC', 'PAUC', 'MAPR'],
    ['AU1U', 'AUNU', 'AU1P', 'AUNP'],
    ['MFM', 'MAVA', 'MAVG'],
    ['MSE'],
    ['ACC', 'KAPS'],
    ['LOGL'],
    ['CALL']
]

smallDataSet = biClass
largeDataSet = [
    ['LOGL'],
    ['AU1U', 'AUNU', 'AU1P', 'AUNP'],
    ['MPR', 'MAE'],
    ['SAUC', 'PAUC', 'MAPR'],
    ['ACC', 'KAPS', 'MAVG', 'MFM', 'MAVA'],
    ['CALB'],
    ['MSE'],
    ['CALL'],
]

probClass = [
    ['CALL'],
    ['MSE', 'ACC', 'KAPS', 'MAVG', 'MFM', 'MAVA'],
    ['SAUC']
]
```

```

python.py

# All metrics group
def flatOut(nestedList):
    listOut = []
    for x in nestedList:
        listOut += x
    return listOut

allMetrics = flatOut(biClass)
# Check if group has all the metrics
def checkAllMetrics(group):
    global allMetrics
    if len(flatOut(group)) < len(allMetrics):
        return False
    else:
        return True

# Procedure to Get the minimum number of metrics
# Pick a group with all the metrics
groups = [biClass, balancedClass, smallDataSet, probClass]
groupNames = ['biClass', 'balancedClass', 'smallDataSet', 'biClass', 'probClass']
groupsDict = {x:y for x,y in zip(groupNames, groups)}
def getGroupNameAllMetrics():
    global groups
    global groupNames
    listOut = []
    for gr, name in zip(groups, groupNames):
        if checkAllMetrics(gr):
            listOut.append(name)
    return listOut

names = getGroupNameAllMetrics()
print(names)
# From the groups with all the metrics, choose the one with less subgroups
# Return the group name with less subgroups
def returnSmallestSubgroups(groups):
    global groupsDict
    minLen = np.inf
    minName = 'NA'
    for name in groups:
        if len(groupsDict.get(name)) < minLen:
            minLen = len(groupsDict.get(name))
            minName = name
    return name
name = returnSmallestSubgroups(names)
print(name)
# Choose from each subgroup one metric
def minimumLiestOfMetrics(groupName):
    global groupsDict
    listOut = []
    for lst in groupsDict.get(groupName):
        listOut.append(lst[0])
    return listOut
mIom = minimumLiestOfMetrics(name)
print(mIom)

# If the metrics chosen are at least in one subgroup of a
#second group;
return the metrics
def groupDifference(mIom, groupName2):
    global groupsDict
    gp2subGroups = groupsDict.get(groupName2).copy()
    listOut = []
    for metric in mIom:
        for subgp in gp2subGroups:
            if metric in subgp:
                gp2subGroups.remove(subgp)
    return gp2subGroups

missingMetrics = flatOut(groupDifference(mIom, 'balancedClass'))
missingMetrics2 = flatOut(groupDifference(mIom, 'smallDataSet'))
print(missingMetrics, missingMetrics2)
# otherwise choose one metric from subgroups without representation

```

Anexo A

Biclase, balanceado, dataset pequeño y ruido de probabilidad (probabilidades traslapadas)

```
[55] groups = [biClass, balancedClass, smallDataSet, probClass]
      groupNames = ['biClass', 'balancedClass', 'smallDataSet', 'probClass']
      groupsDict = {x:y for x,y in zip(groupNames, groups)}
```

Grupos con todas las métricas:

```
[56] names = getGroupNameAllMetrics()
      names
```

```
['biClass', 'balancedClass', 'smallDataSet']
```

Subgrupo con todas las métricas pero menos subgrupos:

```
[57] name = returnSmallestSubgroups(names)
      print(name)
```

```
smallDataSet
```

Número Mínimo de métricas:

```
[58] mlom = minimumLiestOfMetrics(name)
      print(mlom)
```

```
['CALB', 'CALL', 'MPR', 'ACC', 'LOGL', 'MSE', 'AU1U']
```

Agregar métricas de subgrupos no tomados en cuenta con las métricas seleccionadas:

```
▶ missingMetrics = flatOut(groupDifference(mlom, 'balancedClass'))
  missingMetrics2 = flatOut(groupDifference(mlom, 'biClass'))
  missingMetrics3 = flatOut(groupDifference(mlom, 'probClass'))
  print(missingMetrics, missingMetrics2, missingMetrics3)
```

```
[] [] ['SAUC']
```

Anexo B

Biclase, desbalanceado, dataset pequeño y ruido de probabilidad (probabilidades traslapadas)

```
[48] groups = [biClass, imbalancedClass, smallDataSet, probClass]
      groupNames = ['biClass', 'imbalancedClass', 'smallDataSet', 'probClass']
      groupsDict = {x:y for x,y in zip(groupNames, groups)}
```

Grupos con todas las métricas:

```
[49] names = getGroupNameAllMetrics()
      names

['biClass', 'imbalancedClass', 'smallDataSet']
```

Subgrupo con todas las métricas pero menos subgrupos:

```
[50] name = returnSmallestSubgroups(names)
      print(name)

smallDataSet
```

Número Mínimo de métricas:

```
[51] mlom = minimumLiestOfMetrics(name)
      print(mlom)

['CALB', 'CALL', 'MPR', 'ACC', 'LOGL', 'MSE', 'AU1U']
```

Agregar métricas de subgrupos no tomados en cuenta con las métricas seleccionadas:

```
[54] missingMetrics = flatOut(groupDifference(mlom, 'imbalancedClass'))
      missingMetrics2 = flatOut(groupDifference(mlom, 'biClass'))
      missingMetrics3 = flatOut(groupDifference(mlom, 'probClass'))
      print(missingMetrics, missingMetrics2, missingMetrics3)

['SAUC', 'PAUC', 'MAPR', 'MFM', 'MAVA', 'MAVG'] [] ['SAUC']
```

Anexo C

Biclasa, desbalanceado, dataset grande y ruido de probabilidad (probabilidades traslapadas)

```
[61] groups = [biClass, imbalancedClass, largeDataSet, probClass]
      groupNames = ['biClass', 'imbalancedClass', 'largeDataSet', 'probClass']
      groupsDict = {x:y for x,y in zip(groupNames, groups)}
```

Grupos con todas las métricas:

```
[62] names = getGroupNameAllMetrics()
      names

['biClass', 'imbalancedClass', 'largeDataSet']
```

Subgrupo con todas las métricas pero menos subgrupos:

```
[63] name = returnSmallestSubgroups(names)
      print(name)

largeDataSet
```

Número Mínimo de métricas:

```
[64] mlom = minimumLiestOfMetrics(name)
      print(mlom)

['LOGL', 'AU1U', 'MPR', 'SAUC', 'ACC', 'CALB', 'MSE', 'CALL']
```

Agregar métricas de subgrupos no tomados en cuenta con las métricas seleccionadas:

```
[66] missingMetrics = flatOut(groupDifference(mlom, 'imbalancedClass'))
      missingMetrics2 = flatOut(groupDifference(mlom, 'biClass'))
      missingMetrics3 = flatOut(groupDifference(mlom, 'probClass'))
      print(missingMetrics, missingMetrics2, missingMetrics3)

['MFM', 'MAVA', 'MAVG'] [] []
```

Anexo D

Biclase, desbalanceado, dataset pequeño y ruido de probabilidad (probabilidades traslapadas)

```
[67] groups = [biClass, imbalancedClass, smallDataSet, probClass]
      groupNames = ['biClass', 'imbalancedClass', 'smallDataSet', 'probClass']
      groupsDict = {x:y for x,y in zip(groupNames, groups)}
```

Grupos con todas las métricas:

```
[68] names = getGroupNameAllMetrics()
      names

['biClass', 'imbalancedClass', 'smallDataSet']
```

Subgrupo con todas las métricas pero menos subgrupos:

```
[69] name = returnSmallestSubgroups(names)
      print(name)
```

```
smallDataSet
```

Número Mínimo de métricas:

```
[70] mlom = minimumLiestOfMetrics(name)
      print(mlom)

['CALB', 'CALL', 'MPR', 'ACC', 'LOGL', 'MSE', 'AU1U']
```

Agregar métricas de subgrupos no tomados en cuenta con las métricas seleccionadas:

```
▶ missingMetrics = flatOut(groupDifference(mlom, 'imbalancedClass'))
  missingMetrics2 = flatOut(groupDifference(mlom, 'biClass'))
  missingMetrics3 = flatOut(groupDifference(mlom, 'probClass'))
  print(missingMetrics, missingMetrics2, missingMetrics3)
```

```
['SAUC', 'PAUC', 'MAPR', 'MFM', 'MAVA', 'MAVG'] [] ['SAUC']
```


Tabla de acrónimos

ACC	Accuracy
AU1P	AUC, variante c(c-1) P
AU1U	AUC, variante c(c-1) U
AUC	Area under the ROC Curve
AUNP	AUC, variante P
AUNU	AUC, variante U
CALB	Calibration by Bins
CALL	Calibration Loss
KAPS	Kappa Statistic
LOGL	Logaritmic Loss
MAE	Mean Absolute Error
MAPR	MActo Average Mean Probability Rate
MAVA	Macro Average Arithmetic
MAVG	Macro Average Geometric
MFM	Mean F Measure
MPR	Mean Probability Rate
MSE	Mean Squared Error
PAUC	Probabilistic AUC
SAUCC	Scored AUC

Referencias bibliográficas

- Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Elsevier
- Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern recognition letters*, 30(1), 27-38.