

Entregable I - Multilabel Datasets

Isabella Hernández García,
z22hegai@uco.es
31 de enero de 2023

ÍNDICE

I. Introducción

II. Respuesta a los ejercicios

II-A.	Ejercicio 2	1
II-A1.	Información representada en los datasets	1
II-B.	Ejercicio 3	2
II-B1.	Script <i>car.py</i>	2
II-C.	Ejercicio 4	2
II-D.	Ejercicio 5	2
II-E.	Ejercicio 6	3
II-E1.	Ejemplo de código y descripción de cada función . .	3
II-E2.	Validación cruzada en el contexto de clasificación multi-etiqueta	4
II-F.	Ejercicio 7	4
II-G.	Ejercicio 8	5
II-G1.	Script	5
II-G2.	Resultados	5
II-H.	Ejercicio 9	5
II-H1.	Método que se comporta mejor globalmente	5
II-H2.	Tiempo de ejecución de cada método	5
II-H3.	Diferencias entre los valores de cada métrica	6

Referencias

I. INTRODUCCIÓN

El objetivo de la práctica es familiarizarse con el tema de clasificación multietiqueta utilizando la biblioteca *Scikit-multilearn*[4]

En el *Link a Github* se encuentran implementadas las soluciones a cada uno de los ejercicios.

II. RESPUESTA A LOS EJERCICIOS

En esta sección se da respuesta a cada una de las preguntas que se encontraban explícitas en la orden.

II-A. Ejercicio 2

Familiarízate con los datasets presentes en Scikit-multilearn y con los que trabajaremos durante las próximas sesiones. ¿Qué información representan?

Concretamente, dispone de los siguientes 17 datasets:

```
{'scene', 'Corel5k', 'bibtex',  
'enron', 'rcv1subset5', 'tmc2007_500',  
'rcv1subset3', 'rcv1subset1', 'delicious',  
'rcv1subset4', 'genbase', 'birds',  
'emotions', 'rcv1subset2', 'mediamill',  
'medical', 'yeast'}
```

II-A1. Información representada en los datasets:

- **Scene:** Un conjunto de datos de imágenes con 2407 imágenes etiquetadas hasta en 6 clases: playa, puesta de sol, hojas de otoño, campo, montaña y urbana. Cada imagen se describe con 294 características numéricas visuales correspondientes a momentos de color espacial en el espacio LUV.
- **Corel5k:** Es un *benchmark* popular para métodos de clasificación y anotación de imágenes. Se basa en 5000 imágenes Corel.
- **Bibtex** Este conjunto de datos se basa en los datos del desafío de descubrimiento ECML/PKDD 2008. Contiene 7395 entradas bibtex del sistema de intercambio de publicaciones y marcadores sociales BibSonomy, anotadas con un subconjunto de las etiquetas asignadas por los usuarios de BibSonomy.
- **Enron** El conjunto de datos de Enron es un subconjunto del corpus de correo electrónico de Enron, etiquetado con un conjunto de categorías. Se basa en una colección de mensajes de correo electrónico que se clasificaron en 53 categorías temáticas, como estrategia de la empresa, humor y asesoramiento legal.
- **Reuters-RCV1** (rcv1subset5, rcv1subset3, rcv1subset1, rcv1subset4, rcv1subset2) Este conjunto de datos es un *benchmark* de referencia bien conocido para los métodos de clasificación de texto. Tiene 5 subconjuntos, cada uno con 6000 artículos asignados en uno o más de 101 temas.
- **TMC2007** Es un subconjunto del conjunto de datos del Sistema de Informes de Seguridad de la Aviación. Contiene 28596 informes de texto libre de seguridad aérea que la tripulación envía después de cada vuelo sobre eventos que tuvieron lugar durante el vuelo. El objetivo es etiquetar los documentos con respecto a los tipos de problemas que describen. El conjunto de datos tiene 49060 atributos discretos correspondientes a términos de la colección. Los informes de seguridad están provistos de 22 etiquetas, cada una de las cuales representa un tipo de problema que aparece durante un vuelo.
- **genbase** Es un conjunto de datos para la clasificación de la función de las proteínas. Cada instancia es una proteína

y cada etiqueta es una clase de proteína. Este conjunto de datos es pequeño comparativamente con el gran número de etiquetas.

- **birds** Es un conjunto de datos para predecir el conjunto de especies de aves que están presentes, dado un clip de audio de diez segundos.
- **emotions** Es un pequeño conjunto de datos para clasificar la música en emociones que evoca según el modelo de estado de ánimo de Tellegen-Watson-Clark: asombrado-sorprendido, feliz-complacido, relajante-tranquilo, tranquilo-quieto, triste-solitario y enojado-agresivo. Consta de 593 canciones con 6 clases.
- **Mediamill** Es un conjunto de datos multimedia para la indexación de vídeo genérico, que se extrajo del punto de referencia TRECVID 2005/2006. Este conjunto de datos contiene 85 horas de datos de noticias de difusión internacional categorizados en 100 etiquetas y cada instancia de vídeo se representa como un vector de características de 120 dimensiones de entidades numéricas.
- **Medical** El conjunto de datos se basa en los datos disponibles durante el Computational Medicine Centers 2007 Medical Natural Language Processing Challenge 10. Consiste en 978 informes clínicos de texto libre etiquetados con uno o más de los 45 códigos de enfermedad.
- **Yeast** Este conjunto de datos contiene expresiones de micromatrices y perfiles filogenéticos para 2417 genes de levadura. Cada gen está anotado con un subconjunto de 14 categorías funcionales (por ejemplo, metabolismo, energía, etc.) del nivel superior del catálogo funcional.

II-B. Ejercicio 3

II-B1. Script carpy:

```
from functools import reduce
from sklearn.datasets import load_dataset
from utils import get_description
```

```
def instances(X):
    """
    Number of instances (N)
    """
    return X.shape[0]
```

```
def attributes(X):
    """
    Number of attributes (F)
    """
    return X.shape[1]
```

```
def labels(y):
    """
    Number of labels (L)
    """
    return y.shape[1]
```

```
def dl(y):
    """
    Distinct Label Set (DL)
    """
    return len(set(list(map(lambda x: sum([e*2**i
    ↪ for i, e in enumerate(x)]), y.toarray()))))
```

```
def pdl(y):
    """
    Proportion of Distinct Label Set (PDL)
    """
    return dl(y) / (2**y.shape[1] - 1)

def lcard(y):
    """
    Label Cardinality (Lcard)
    """
    return reduce(lambda x, y: x+y, map(len,
    ↪ y.data)) / y.shape[0]

def lden(y):
    """
    Label Density LDen
    """
    return lcard(y) / y.shape[1]
```

```
XMETRICS = [instances, attributes]
YMETRICS = [labels,
            dl,
            pdl,
            lcard,
            lden]
```

```
def metrics(datasetName):
    X, y, _, _ = load_dataset(set_name=datasetName,
    ↪ variant="train")
    resultX = {get_description(metric): metric(X)
    ↪ for metric in XMETRICS}
    resultY = {get_description(metric): metric(y)
    ↪ for metric in YMETRICS}
    return (**resultX, **resultY)
```

II-C. Ejercicio 4

Busca en Internet dos datasets extra. Descárgalos localmente y realiza un script en Python para cargarlos y recalcular todas las medidas del paso anterior sobre los mismos. Debes escoger datasets variados, con diferente número de etiquetas, variables e instancias.

La orden no tenía preguntas explícitas para este ejercicio pero se pueden observar los resultados en *Github*

II-D. Ejercicio 5

Familiarízate con la documentación *scikit-multilearn* en: <http://scikit.ml/> Prueba los métodos ML disponibles pertenecientes a las dos categorías (transformación y adaptación) que hemos visto en teoría.

La orden no tenía preguntas explícitas para este ejercicio pero se pueden observar los resultados en *Github*

II-E. Ejercicio 6

```
display(pd.DataFrame(hamming_scores))
```

II-E1. Ejemplo de código y descripción de cada función :

Este código es un ejemplo de validación cruzada que utiliza un clasificador basado en LabelPowerset con Random Forest de scikit-multilearn.

La función *hamming_loss* se utiliza como métrica para evaluar el modelo. La clase *KFold* divide los datos en 5 partes para la validación cruzada. La función *cross_validate* combina la división de datos y la evaluación del modelo en una sola llamada y devuelve tanto las puntuaciones de rendimiento como los tiempos de entrenamiento y prueba. Por otro lado, *cross_val_score* solo devuelve las puntuaciones del modelo en cada fold sin tiempos.

```
from skmultilearn.dataset import load_dataset

X_train, y_train, _, _ =
    ↳ load_dataset(set_name="scene", variant="un")
X_test, y_test, _, _ =
    ↳ load_dataset(set_name="scene", variant="test")
scene:train - exists, not redownloading
scene:test - exists, not redownloading
from skmultilearn.problem_transform import
    ↳ LabelPowerset
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import hamming_loss
from sklearn.metrics import make_scorer
from sklearn.model_selection import cross_validate,
    ↳ KFold, cross_val_score
import pandas as pd
from sklearn.metrics import classification_report
classifier = LabelPowerset(
    classifier =
        ↳ RandomForestClassifier(n_estimators=100),
    require_dense = [False, True]
)

# Es una función que crea un objeto que Recibe una
    ↳ métrica y evalúa el clasificador con ella. Puede
    ↳ ser utilizado en la evaluación de un modelo con
    ↳ cross_validate() o cross_val_score() usando el
    ↳ parámetro scoring.
hamming_scorer = make_scorer(hamming_loss)

# Es una clase que divide un conjunto de datos en
    ↳ partes para la validación cruzada.
kFold = KFold(n_splits=5)

# Kfold puede ser utilizado para obtener la división
    ↳ del dataset de la forma kFold.split(X_train) o
    ↳ puede ser utilizado por cross_validate o por
    ↳ cross_val_score usando el parámetro cv.

# cross_validate combina la división de datos y la
    ↳ evaluación del modelo en una sola llamada.
    ↳ Devuelve tanto las puntuaciones de rendimiento
    ↳ como los tiempos de entrenamiento y prueba
hamming_scores_with_times =
    ↳ cross_validate(classifier, X_train, y_train,
    ↳ scoring=hamming_scorer, cv=kFold)

display(pd_
    ↳ .DataFrame(hamming_scores_with_times).transpose())

# Funciona igual a cross_validate, pero devuelve
    ↳ solamente la puntuación del modelo en cada fold.
    ↳ No devuelve los tiempos
hamming_scores = cross_val_score(classifier,
    ↳ X_train, y_train, scoring=hamming_scorer,
    ↳ cv=kFold)
```

	0	1	2	3	4
fit_time	0.902332	0.945678	0.926189	0.879335	0.851875
score_time	0.016609	0.016332	0.016637	0.016828	0.017307
test_score	0.314815	0.157713	0.130165	0.280303	0.307851

Figura 1. Salida de la función cross validate.

	0
0	0.310700
1	0.161846
2	0.127410
3	0.278926
4	0.309229

Figura 2. Salida de la función cross val scores. Solo contiene los resultados de la métrica en cada fold.

II-E2. Validación cruzada en el contexto de clasificación multi-etiqueta: La validación cruzada en el contexto de clasificación multi-etiqueta funcionará de manera similar que en clasificación de etiquetas únicas. Consiste en dividir los datos de entrenamiento en k partes o "folds", donde k es un número determinado previamente, el modelo se entrena k veces, cada vez utilizando un fold diferente como datos de prueba y los $k-1$ folds restantes como datos de entrenamiento.

Quizás la única diferencia sería las métricas que se utilizan, que deben estar orientadas directamente a combinar los resultados de cada etiqueta (ver ejercicio 7).

Otro aspecto a tener en cuenta sería como equilibrar la cantidad de clases en cada *fold*, que funcionaría de manera diferente.

II-F. Ejercicio 7

- **Accuracy:** Es el número de predicciones correctas dividido por el número total de predicciones. En clasificación multietiqueta accuracy se calcula para el subconjunto, y calcula la fracción de veces que todas las etiquetas predichas para una muestra coinciden exactamente con las etiquetas reales. [3]
- **HammingLoss:** Es una métrica de evaluación basada en instancias que evalúa cuántas veces un par instancia-etiqueta se clasifica mal. Hamming Loss penaliza las etiquetas individuales que fueron clasificadas incorrectamente. No tiene en cuenta la relación entre etiquetas correctas e incorrectas. [3]
- **Precision:** Se refiere a la fracción de resultados verdaderos positivos de todos los resultados positivos. Mide la estabilidad de la medida frente a las repeticiones. [3] De todas las predicciones positivas, ¿cuántas son realmente positivas?
- **Recall:** Calcula la fracción de resultados verdaderos positivos que fueron correctamente identificados. [3] De todos los casos positivos reales, ¿cuántos son positivos predichos?
- **F1Score:** *F1Score* es la media armónica entre la precisión y el recall. Su objetivo sería maximizar tanto la precisión como la recuperación. [3]

En una tarea de clasificación multiclase y multietiqueta, se pueden aplicar las métricas de precisión, recall y F-measures a cada etiqueta de forma independiente. Hay varias maneras de combinar los resultados a través de las etiquetas, especificadas por el argumento de promedio en las funciones *average_precision_score* (sólo multietiqueta), *f1_score*, *fbeta_score*, *precision_recall_fscore_support*, *precision_score* y *recall_score*.

II-G. Ejercicio 8

Escribe un script en Python (*cl-cv.py*) seleccionando al menos 3 métodos (que NO pertenezcan todos a la misma categoría) para evaluarlos con 5 de los datasets que recopilaste anteriormente y calcula las métricas resultantes mediante validación cruzada. El script mostrará el resultado de las métricas anteriores

II-G1. Script:

```
from sklearn.metrics import classification_report,
↳ hamming_loss, make_scorer
from sklearn.model_selection import KFold,
↳ cross_val_predict, cross_validate
from sklearn.datasets import load_dataset
from sklearn import get_description
N_SPLITS = 3

def eval_report(classifierFactory, dataset_name):
    X_train, y_train, feature_names, label_names =
    ↳ load_dataset(
        set_name=dataset_name, variant="train")
    X_test, y_test, feature_names, label_names =
    ↳ load_dataset(
        set_name=dataset_name, variant="test")
    target_names = [label_name[0] for label_name in
    ↳ label_names]

    classifier = classifierFactory()

    kFold = KFold(n_splits=N_SPLITS)

    scoring = {**{item: item for item in
    ['accuracy',
    'precision_macro', 'precision_micro',
    'recall_macro',
    'recall_micro',
    'f1_macro',
    'f1_micro']}, **{'hamming':
    ↳ make_scorer(hamming_loss)}}

    scores = cross_validate(classifier, X_train,
    ↳ y_train, scoring=scoring,
        cv=kFold,
        ↳ return_train_score=True,
        ↳ return_estimator=True)

    # Making the classification_report using the
    ↳ test set
    # And returning both cross validation scores and
    ↳ report
    # for comparison
    estimator = scores['estimator'][N_SPLITS - 1]
    y_pred = estimator.predict(X_test)

    report = classification_report(y_test, y_pred,
    ↳ target_names=target_names)

    return scores, report
```

II-G2. Resultados: En esta sección se describen los resultados obtenidos al aplicar cada uno de los clasificadores a cada dataset

Datasets seleccionados

- scene
- enron
- medical
- genbase
- emotions

Clasificadores seleccionados

- (Transformation) Label Powerset with Random Forest
- (Adaptation) Binary Relevance kNN classifier (BRkNNa) [2]
- (Adaptation) Multilabel Twin Support Vector Machine (MLTSVM) [1]

## Classification report				
	precision	recall	f1-score	support
Beach	0.43	0.89	0.58	200
Sunset	0.87	0.92	0.90	199
Fallfoliage	0.81	0.89	0.85	200
Field	0.83	0.78	0.81	237
Mountain	0.54	0.28	0.37	256
Urban	0.00	0.00	0.00	207
micro avg	0.67	0.61	0.64	1299
macro avg	0.58	0.63	0.58	1299
weighted avg	0.58	0.61	0.58	1299
samples avg	0.67	0.63	0.64	1299

Figura 3. Métricas para el dataset Scene con el método Label Powerset with Random Forest

	precision	recall	f1-score	support
Beach	0.49	0.81	0.61	200
Sunset	0.99	0.60	0.75	199
Fallfoliage	0.68	0.82	0.74	200
Field	0.70	0.84	0.77	237
Mountain	0.59	0.33	0.43	256
Urban	0.00	0.00	0.00	207
micro avg	0.65	0.56	0.60	1299
macro avg	0.58	0.57	0.55	1299
weighted avg	0.58	0.56	0.55	1299
samples avg	0.61	0.57	0.58	1299

Figura 4. Métricas para el dataset Scene con el método BRkNNa

	precision	recall	f1-score	support
Beach	0.51	0.42	0.46	200
Sunset	0.94	0.38	0.54	199
Fallfoliage	0.42	0.32	0.36	200
Field	0.74	0.26	0.39	237
Mountain	0.40	0.22	0.28	256
Urban	0.21	0.07	0.10	207
micro avg	0.52	0.27	0.36	1299
macro avg	0.54	0.28	0.36	1299
weighted avg	0.53	0.27	0.35	1299
samples avg	0.24	0.28	0.25	1299

Figura 5. Métricas para el dataset Scene con el método MLTSVM

II-H. Ejercicio 9

II-H1. Método que se comporta mejor globalmente: El método **Label Powerset con Random Forest** parece obtener mejor rendimiento en todos los datasets. El valor de los promedios de F1Score es muy superior cuando se utiliza este método comparado con los demás. Obtuvo resultados especialmente buenos con el dataset genbase. 12

II-H2. Tiempo de ejecución de cada método: Para calcular el tiempo de ejecución de cada método se utilizó el resultado de la función *cross_validate*, esta retorna el tiempo de entrenamiento y de tests en cada fold. Si acumulamos el

	precision	recall	f1-score	support
A.A8	0.00	0.00	0.00	8
C.C9	0.20	0.05	0.07	22
B.B12	0.00	0.00	0.00	3
C.C11	0.33	0.08	0.13	12
C.C5	0.00	0.00	0.00	31
C.C7	0.33	0.17	0.22	24
B.B2	0.76	0.89	0.82	313
B.B3	0.00	0.00	0.00	12
D.D16	1.00	0.33	0.50	3
A.A7	0.05	0.11	0.06	9
D.D1	1.00	0.33	0.50	3
A.A4	0.44	0.89	0.59	188
C.C2	0.25	0.10	0.14	42
A.A3	0.56	0.32	0.41	68
A.A1	0.83	0.37	0.51	282
D.D9	0.00	0.00	0.00	13
D.D19	0.00	0.00	0.00	3
B.B8	0.33	0.11	0.17	9
D.D12	0.20	0.07	0.11	14
D.D6	0.00	0.00	0.00	7
C.C8	0.17	0.05	0.08	40
A.A6	0.39	0.22	0.28	65
B.B9	0.42	0.33	0.37	15
A.A5	0.38	0.23	0.29	26
C.C10	0.11	0.05	0.07	20
B.B1	0.58	0.76	0.66	226
D.D5	0.00	0.00	0.00	5
B.B11	0.00	0.00	0.00	3
D.D2	0.00	0.00	0.00	6
B.B4	0.75	0.75	0.75	32
D.D15	0.00	0.00	0.00	1
C.C4	0.11	0.05	0.07	20
D.D8	0.00	0.00	0.00	4
B.B6	0.25	0.12	0.17	8
D.D3	0.00	0.00	0.00	7
D.D13	0.00	0.00	0.00	3
D.D7	0.00	0.00	0.00	3
C.C12	0.00	0.00	0.00	12
B.B7	0.33	0.20	0.25	5
C.C6	0.62	0.38	0.47	86
B.B5	0.00	0.00	0.00	4
D.D11	0.00	0.00	0.00	8
A.A2	0.50	0.24	0.32	17
C.C3	0.67	0.08	0.15	24
D.D10	0.37	0.15	0.21	47
D.D18	0.00	0.00	0.00	1
B.B13	0.55	0.35	0.42	113
D.D17	0.00	0.00	0.00	1
B.B10	0.33	0.17	0.22	6
C.C1	0.38	0.21	0.27	63
D.D4	0.00	0.00	0.00	5
C.C13	0.00	0.00	0.00	3
D.D14	0.00	0.00	0.00	2
micro avg	0.56	0.47	0.51	1947
macro avg	0.25	0.15	0.17	1947
weighted avg	0.53	0.47	0.46	1947
samples avg	0.56	0.51	0.52	1947

Figura 6. Métricas para el dataset enron con el método Label Powerset with Random Forest

tiempo de cada uno de los métodos en train y test obtenemos los siguientes resultados:

Total fit time

- Label Powerset with Random Forest: 11.121889352798462
- Binary Relevance kNN classifier (BRkNNa): 0.034305572509765625
- Multilabel Twin Support Vector Machine (MLTSVM): 678.8933806419373

Podemos observar que el tiempo de ejecución de MLTSVM es mucho mayor.

II-H3. Diferencias entre los valores de cada métrica: Tomemos como ejemplo la figura 18 que representa el resultado de las métricas en el entrenamiento y prueba del modelo. Podemos ver que $test_{accuracy}$ muestra resultados muy satisfactorios, mientras que $test_{hamming}$ es muy bajo. Esto puede estar suceder porque el modelo puede estar fallando en predecir todas las etiquetas correctas para una instancia dada.

REFERENCIAS

- [1] Wei-Jie Chen, Yuan-Hai Shao, Chun-Na Li, and Nai-Yang Deng. Mltsvm: a novel twin support vector machine to multi-label learning. *Pattern Recognition*, 52:61–74, 2016.
- [2] Ioannis Vlahavas Eleftherios Spyromitros, Grigorios Tsoumakas. An empirical study of lazy multilabel classification algorithms. In *Proc. 5th Hellenic Conference on Artificial Intelligence (SETN 2008)*, 2008.

A.A8	0.00	0.00	0.00	8
C.C9	0.50	0.05	0.08	22
B.B12	0.00	0.00	0.00	3
C.C11	0.00	0.00	0.00	12
C.C5	0.00	0.00	0.00	31
C.C7	0.80	0.17	0.28	24
B.B2	0.76	0.39	0.52	313
B.B3	0.00	0.00	0.00	12
D.D16	0.00	0.00	0.00	3
A.A7	0.00	0.00	0.00	9
D.D1	0.00	0.00	0.00	3
A.A4	0.47	0.61	0.53	188
C.C2	0.43	0.07	0.12	42
A.A3	0.55	0.38	0.45	68
A.A1	0.75	0.33	0.46	282
D.D9	0.00	0.00	0.00	13
D.D19	0.00	0.00	0.00	3
B.B8	0.00	0.00	0.00	9
D.D12	0.25	0.07	0.11	14
D.D6	0.00	0.00	0.00	7
C.C8	0.50	0.03	0.05	40
A.A6	0.24	0.12	0.16	65
B.B9	0.44	0.47	0.45	15
A.A5	0.50	0.12	0.19	26
C.C10	0.14	0.05	0.07	20
B.B1	0.73	0.25	0.38	226
D.D5	0.00	0.00	0.00	5
B.B11	0.00	0.00	0.00	3
D.D2	0.00	0.00	0.00	6
B.B4	0.85	0.69	0.76	32
D.D15	0.00	0.00	0.00	1
C.C4	0.33	0.05	0.09	20
D.D8	0.00	0.00	0.00	4
B.B6	0.00	0.00	0.00	8
D.D3	0.00	0.00	0.00	7
D.D13	0.00	0.00	0.00	3
D.D7	0.00	0.00	0.00	3
C.C12	0.00	0.00	0.00	12
B.B7	0.00	0.00	0.00	5
C.C6	0.74	0.33	0.45	86
B.B5	0.00	0.00	0.00	4
D.D11	0.00	0.00	0.00	8
A.A2	1.00	0.18	0.30	17
C.C3	0.50	0.04	0.08	24
D.D10	0.36	0.11	0.16	47
D.D18	0.00	0.00	0.00	1
B.B13	0.38	0.54	0.44	113
D.D17	0.00	0.00	0.00	1
B.B10	0.00	0.00	0.00	6
C.C1	0.43	0.10	0.16	63
D.D4	0.00	0.00	0.00	5
C.C13	0.00	0.00	0.00	3
D.D14	0.00	0.00	0.00	2
micro avg	0.56	0.29	0.38	1947
macro avg	0.22	0.10	0.12	1947
weighted avg	0.55	0.29	0.35	1947
samples avg	0.48	0.34	0.37	1947

Figura 7. Métricas para el dataset enron con el método BRkNNa

- [3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [4] P.ński@ and T. Kajdanowicz. A scikit-based Python environment for performing multi-label classification. *ArXiv e-prints*, February 2017.

	precision	recall	f1-score	support
A.A8	0.00	0.00	0.00	8
C.C9	0.00	0.00	0.00	22
B.B12	0.00	0.00	0.00	3
C.C11	1.00	0.08	0.15	12
C.C5	0.00	0.00	0.00	31
C.C7	0.00	0.00	0.00	24
B.B2	0.77	0.21	0.33	313
B.B3	0.00	0.00	0.00	12
D.D16	0.00	0.00	0.00	3
A.A7	0.00	0.00	0.00	9
D.D1	1.00	0.33	0.50	3
A.A4	0.88	0.20	0.33	188
C.C2	0.00	0.00	0.00	42
A.A3	0.60	0.18	0.27	68
A.A1	0.78	0.15	0.26	282
D.D9	0.00	0.00	0.00	13
D.D19	0.00	0.00	0.00	3
B.B8	0.00	0.00	0.00	9
D.D12	0.00	0.00	0.00	14
D.D6	0.00	0.00	0.00	7
C.C8	0.05	0.05	0.05	40
A.A6	0.60	0.09	0.16	65
B.B9	0.00	0.00	0.00	15
A.A5	0.50	0.04	0.07	26
C.C10	0.50	0.10	0.17	20
B.B1	0.57	0.14	0.22	226
D.D5	0.00	0.00	0.00	5
B.B11	0.00	0.00	0.00	3
D.D2	0.00	0.00	0.00	6
B.B4	0.83	0.16	0.26	32
D.D15	0.00	0.00	0.00	1
C.C4	0.25	0.05	0.08	20
D.D8	0.00	0.00	0.00	4
B.B6	1.00	0.12	0.22	8
D.D3	0.00	0.00	0.00	7
D.D13	0.00	0.00	0.00	3
D.D7	0.00	0.00	0.00	3
C.C12	0.00	0.00	0.00	12
B.B7	0.00	0.00	0.00	5
C.C6	0.70	0.16	0.26	86
B.B5	0.00	0.00	0.00	4
D.D11	0.00	0.00	0.00	8
A.A2	1.00	0.06	0.11	17
C.C3	0.14	0.04	0.06	24
D.D10	0.50	0.04	0.08	47
D.D18	0.00	0.00	0.00	1
B.B13	0.61	0.12	0.21	113
D.D17	0.00	0.00	0.00	1
B.B10	0.00	0.00	0.00	6
C.C1	0.52	0.19	0.28	63
D.D4	0.00	0.00	0.00	5
C.C13	0.00	0.00	0.00	3
D.D14	0.00	0.00	0.00	2
micro avg	0.59	0.13	0.21	1947
macro avg	0.24	0.05	0.08	1947
weighted avg	0.58	0.13	0.21	1947
samples avg	0.31	0.13	0.17	1947

Figura 8. Métricas para el dataset enron con el método MLTSVM

	precision	recall	f1-score	support
Class-0-593.70	0.89	0.82	0.85	77
Class-1-079.99	1.00	0.33	0.50	6
Class-2-786.09	0.00	0.00	0.00	1
Class-3-759.89	0.25	1.00	0.40	1
Class-4-753.0	0.83	0.88	0.86	168
Class-5-786.2	0.00	0.00	0.00	0
Class-6-V72.5	0.00	0.00	0.00	0
Class-7-511.9	0.00	0.00	0.00	1
Class-8-596.8	0.00	0.00	0.00	1
Class-9-599.0	0.59	0.96	0.73	74
Class-10-510.0	0.00	0.00	0.00	11
Class-11-593.5	0.00	0.00	0.00	6
Class-12-V13.09	0.00	0.00	0.00	5
Class-13-791.0	0.00	0.00	0.00	1
Class-14-789.00	1.00	0.50	0.67	4
Class-15-593.1	0.00	0.00	0.00	2
Class-16-462	0.00	0.00	0.00	0
Class-17-592.0	1.00	0.40	0.57	5
Class-18-786.59	0.00	0.00	0.00	1
Class-19-785.6	0.00	0.00	0.00	3
Class-20-V67.09	0.00	0.00	0.00	0
Class-21-795.5	1.00	0.90	0.95	10
Class-22-789.00	0.00	0.00	0.00	4
Class-23-786.50	0.88	0.91	0.89	23
Class-24-596.54	0.85	1.00	0.92	35
Class-25-787.03	0.00	0.00	0.00	1
Class-26-V42.0	0.00	0.00	0.00	0
Class-27-786.05	0.00	0.00	0.00	2
Class-28-753.21	0.00	0.00	0.00	3
Class-29-783.0	0.00	0.00	0.00	1
Class-30-277.00	1.00	0.33	0.50	12
Class-31-780.6	0.53	0.93	0.68	45
Class-32-486	0.81	0.67	0.74	83
Class-33-788.41	0.00	0.00	0.00	1
Class-34-V13.02	0.20	0.08	0.11	13
Class-35-493.90	0.33	0.06	0.10	17
Class-36-788.30	0.57	0.13	0.22	30
Class-37-753.3	0.00	0.00	0.00	12
Class-38-593.89	0.92	0.48	0.63	25
Class-39-758.6	0.33	0.10	0.15	10
Class-40-741.90	0.00	0.00	0.00	0
Class-41-591	0.79	0.73	0.76	51
Class-42-599.7	0.00	0.00	0.00	1
Class-43-279.12	0.82	0.61	0.70	23
Class-44-786.07	0.89	0.55	0.68	31
micro avg	0.76	0.68	0.71	800
macro avg	0.34	0.28	0.28	800
weighted avg	0.71	0.68	0.67	800
samples avg	0.76	0.71	0.72	800

Figura 9. Métricas para el dataset medical con el método Label Powerset with Random Forest

	precision	recall	f1-score	support
Class-0-593_70	1.00	0.10	0.19	77
Class-1-079_99	0.00	0.00	0.00	6
Class-2-786_09	0.00	0.00	0.00	1
Class-3-759_89	0.00	0.00	0.00	1
Class-4-753_0	0.79	0.83	0.81	168
Class-5-786_2	0.00	0.00	0.00	0
Class-6-V72_5	0.00	0.00	0.00	0
Class-7-511_9	0.00	0.00	0.00	1
Class-8-596_8	0.00	0.00	0.00	1
Class-9-599_0	0.69	0.80	0.74	74
Class-10-518_0	0.00	0.00	0.00	11
Class-11-593_5	0.00	0.00	0.00	6
Class-12-V13_09	0.00	0.00	0.00	5
Class-13-791_0	0.00	0.00	0.00	1
Class-14-789_00	0.00	0.00	0.00	4
Class-15-593_1	0.00	0.00	0.00	2
Class-16-462	0.00	0.00	0.00	0
Class-17-592_0	0.00	0.00	0.00	5
Class-18-786_59	0.00	0.00	0.00	1
Class-19-785_6	0.00	0.00	0.00	3
Class-20-V67_09	0.00	0.00	0.00	0
Class-21-795_5	0.00	0.00	0.00	10
Class-22-789_09	0.00	0.00	0.00	4
Class-23-786_50	1.00	0.35	0.52	23
Class-24-596_54	0.92	0.31	0.47	35
Class-25-787_03	0.00	0.00	0.00	1
Class-26-V42_0	0.00	0.00	0.00	0
Class-27-786_05	0.00	0.00	0.00	2
Class-28-753_21	0.00	0.00	0.00	3
Class-29-783_0	0.00	0.00	0.00	1
Class-30-277_00	0.00	0.00	0.00	12
Class-31-780_6	0.40	0.00	0.60	45
Class-32-486	0.65	0.41	0.50	83
Class-33-788_41	0.00	0.00	0.00	1
Class-34-V13_02	0.00	0.00	0.00	13
Class-35-493_90	0.00	0.00	0.00	17
Class-36-788_30	0.00	0.00	0.00	30
Class-37-753_3	0.00	0.00	0.00	12
Class-38-593_89	0.80	0.16	0.27	25
Class-39-758_6	0.00	0.00	0.00	10
Class-40-741_90	0.00	0.00	0.00	0
Class-41-591	0.80	0.24	0.36	51
Class-42-599_7	0.00	0.00	0.00	1
Class-43-279_12	1.00	0.13	0.23	23
Class-44-786_07	1.00	0.03	0.06	31
micro avg	0.71	0.40	0.51	800
macro avg	0.20	0.09	0.11	800
weighted avg	0.63	0.40	0.42	800
samples avg	0.45	0.41	0.42	800

Figura 10. Métricas para el dataset medical con el método BRkNna

	precision	recall	f1-score	support
Class-0-593_70	0.00	0.00	0.00	77
Class-1-079_99	0.00	0.00	0.00	6
Class-2-786_09	0.00	0.00	0.00	1
Class-3-759_89	0.00	0.00	0.00	1
Class-4-753_0	0.00	0.00	0.00	168
Class-5-786_2	0.00	0.00	0.00	0
Class-6-V72_5	0.00	0.00	0.00	0
Class-7-511_9	0.00	0.00	0.00	1
Class-8-596_8	0.00	0.00	0.00	1
Class-9-599_0	1.00	0.08	0.15	74
Class-10-518_0	0.00	0.00	0.00	11
Class-11-593_5	0.00	0.00	0.00	6
Class-12-V13_09	0.00	0.00	0.00	5
Class-13-791_0	0.00	0.00	0.00	1
Class-14-789_00	0.00	0.00	0.00	4
Class-15-593_1	0.00	0.00	0.00	2
Class-16-462	0.00	0.00	0.00	0
Class-17-592_0	0.00	0.00	0.00	5
Class-18-786_59	0.00	0.00	0.00	1
Class-19-785_6	0.00	0.00	0.00	3
Class-20-V67_09	0.00	0.00	0.00	0
Class-21-795_5	0.00	0.00	0.00	10
Class-22-789_09	0.00	0.00	0.00	4
Class-23-786_50	0.00	0.00	0.00	23
Class-24-596_54	0.00	0.00	0.00	35
Class-25-787_03	0.00	0.00	0.00	1
Class-26-V42_0	0.00	0.00	0.00	0
Class-27-786_05	0.00	0.00	0.00	2
Class-28-753_21	0.00	0.00	0.00	3
Class-29-783_0	0.00	0.00	0.00	1
Class-30-277_00	0.00	0.00	0.00	12
Class-31-780_6	0.00	0.00	0.00	45
Class-32-486	1.00	0.01	0.02	83
Class-33-788_41	0.00	0.00	0.00	1
Class-34-V13_02	0.50	0.08	0.13	13
Class-35-493_90	0.00	0.00	0.00	17
Class-36-788_30	0.50	0.03	0.06	30
Class-37-753_3	0.00	0.00	0.00	12
Class-38-593_89	0.00	0.00	0.00	25
Class-39-758_6	0.00	0.00	0.00	10
Class-40-741_90	0.00	0.00	0.00	0
Class-41-591	1.00	0.06	0.11	51
Class-42-599_7	0.00	0.00	0.00	1
Class-43-279_12	0.00	0.00	0.00	23
Class-44-786_07	1.00	0.06	0.12	31
micro avg	0.88	0.02	0.03	800
macro avg	0.11	0.01	0.01	800
weighted avg	0.33	0.02	0.03	800
samples avg	0.02	0.02	0.02	800

Figura 11. Métricas para el dataset medical con el método MLTSVM

	precision	recall	f1-score	support
PDOC00154	1.00	1.00	1.00	24
PDOC00343	1.00	1.00	1.00	21
PDOC00271	1.00	1.00	1.00	18
PDOC00064	1.00	1.00	1.00	16
PDOC00791	1.00	1.00	1.00	54
PDOC00380	1.00	1.00	1.00	6
PDOC50007	1.00	1.00	1.00	7
PDOC00224	1.00	1.00	1.00	15
PDOC00100	1.00	1.00	1.00	1
PDOC00670	1.00	1.00	1.00	24
PDOC50002	1.00	0.91	0.95	11
PDOC50106	1.00	0.90	0.95	10
PDOC00561	1.00	1.00	1.00	10
PDOC50017	1.00	1.00	1.00	3
PDOC50003	1.00	0.75	0.86	4
PDOC50006	1.00	1.00	1.00	1
PDOC50156	1.00	1.00	1.00	2
PDOC00662	1.00	1.00	1.00	10
PDOC00018	1.00	1.00	1.00	3
PDOC50001	1.00	1.00	1.00	1
PDOC00014	0.00	0.00	0.00	1
PDOC00750	0.00	0.00	0.00	0
PDOC50106	0.00	0.00	0.00	0
PDOC50199	0.00	0.00	0.00	1
PDOC00660	0.00	0.00	0.00	1
PDOC00653	0.00	0.00	0.00	0
PDOC00030	0.00	0.00	0.00	0
micro avg	1.00	0.97	0.99	245
macro avg	0.74	0.72	0.73	245
weighted avg	0.98	0.97	0.98	245
samples avg	1.00	0.99	0.99	245

Figura 12. Métricas para el dataset genbase con el método Label Powerset with Random Forest

	precision	recall	f1-score	support
PDOC00154	0.42	0.21	0.28	24
PDOC00343	0.42	0.24	0.30	21
PDOC00271	0.75	0.11	0.15	18
PDOC00064	1.00	0.12	0.22	16
PDOC00791	0.50	0.39	0.44	54
PDOC00380	1.00	0.33	0.50	6
PDOC50007	0.67	0.29	0.40	7
PDOC00224	0.67	0.13	0.22	15
PDOC00100	0.00	0.00	0.00	1
PDOC00670	0.21	0.12	0.16	24
PDOC50002	0.00	0.00	0.00	11
PDOC50106	0.00	0.00	0.00	10
PDOC00561	0.40	0.20	0.27	10
PDOC50017	0.00	0.00	0.00	3
PDOC50003	0.00	0.00	0.00	4
PDOC50006	0.00	0.00	0.00	1
PDOC50156	0.00	0.00	0.00	2
PDOC00662	0.29	0.20	0.24	10
PDOC00018	0.00	0.00	0.00	3
PDOC50001	0.00	0.00	0.00	1
PDOC00014	0.00	0.00	0.00	1
PDOC00750	0.00	0.00	0.00	0
PDOC50196	0.00	0.00	0.00	1
PDOC50199	0.00	0.00	0.00	1
PDOC00660	0.00	0.00	0.00	1
PDOC00653	0.00	0.00	0.00	0
PDOC00030	0.00	0.00	0.00	0
micro avg	0.38	0.20	0.26	245
macro avg	0.22	0.09	0.12	245
weighted avg	0.40	0.20	0.25	245
samples avg	0.23	0.23	0.23	245

Figura 13. Métricas para el dataset genbase con el método BRkNna

	precision	recall	f1-score	support
PDOC00154	1.00	0.04	0.08	24
PDOC00343	1.00	0.33	0.50	21
PDOC00271	0.00	0.00	0.00	18
PDOC00064	1.00	0.50	0.67	16
PDOC00791	0.00	0.00	0.00	54
PDOC00380	1.00	0.17	0.29	6
PDOC50007	1.00	0.57	0.73	7
PDOC00224	0.00	0.00	0.00	15
PDOC00100	0.00	0.00	0.00	1
PDOC00670	1.00	0.33	0.50	24
PDOC50002	1.00	0.18	0.31	11
PDOC50106	1.00	0.20	0.33	10
PDOC00561	1.00	0.50	0.67	10
PDOC50017	1.00	1.00	1.00	3
PDOC50003	1.00	0.25	0.40	4
PDOC50006	0.00	0.00	0.00	1
PDOC50156	0.00	0.00	0.00	2
PDOC00662	1.00	0.10	0.18	10
PDOC00018	1.00	0.33	0.50	3
PDOC50001	0.03	1.00	0.05	1
PDOC00014	0.00	0.00	0.00	1
PDOC00750	0.00	0.00	0.00	0
PDOC50196	0.00	0.00	0.00	1
PDOC50199	0.00	0.00	0.00	1
PDOC00660	0.00	0.00	0.00	1
PDOC00653	0.00	0.00	0.00	0
PDOC00030	0.00	0.00	0.00	0
micro avg	0.39	0.18	0.25	245
macro avg	0.48	0.20	0.21	245
weighted avg	0.61	0.18	0.26	245
samples avg	0.15	0.16	0.14	245

Figura 14. Métricas para el dataset genbase con el método MLTSVM

	precision	recall	f1-score	support
amazed-surprised	0.59	0.56	0.57	54
happy-pleased	0.48	0.53	0.50	59
relaxing-calm	0.68	0.84	0.75	96
quiet-still	0.67	0.85	0.75	59
sad-lonely	0.77	0.66	0.71	73
angry-aggressive	0.70	0.69	0.70	58
micro avg	0.65	0.70	0.68	399
macro avg	0.65	0.69	0.66	399
weighted avg	0.66	0.70	0.67	399
samples avg	0.64	0.70	0.64	399

Figura 15. Métricas para el dataset emotions con el método Label Powerset with Random Forest

	precision	recall	f1-score	support
amazed-surprised	0.42	0.24	0.31	54
happy-pleased	0.25	0.22	0.23	59
relaxing-calm	0.72	0.80	0.76	96
quiet-still	0.53	0.32	0.40	59
sad-lonely	0.40	0.29	0.33	73
angry-aggressive	0.56	0.43	0.49	58
micro avg	0.52	0.42	0.46	399
macro avg	0.48	0.38	0.42	399
weighted avg	0.50	0.42	0.45	399
samples avg	0.50	0.43	0.43	399

Figura 16. Métricas para el dataset emotions con el método BRkNNA

	precision	recall	f1-score	support
amazed-surprised	0.58	0.20	0.30	54
happy-pleased	0.47	0.14	0.21	59
relaxing-calm	0.69	0.23	0.34	96
quiet-still	0.75	0.51	0.61	59
sad-lonely	0.60	0.34	0.43	73
angry-aggressive	0.75	0.21	0.32	58
micro avg	0.65	0.27	0.38	399
macro avg	0.64	0.27	0.37	399
weighted avg	0.64	0.27	0.37	399
samples avg	0.33	0.26	0.27	399

Figura 17. Métricas para el dataset emotions con el método MLTSVM

fit_time	2
score_time	0.42048
estimator	LabelPowerset(classifier=RandomForestClassifie...
test_accuracy	0.961039
train_accuracy	1.0
test_precision_macro	0.740741
train_precision_macro	0.851852
test_precision_micro	0.994810
train_precision_micro	1.0
test_recall_macro	0.722222
train_recall_macro	0.851852
test_recall_micro	0.974619
train_recall_micro	1.0
test_f1_macro	0.728395
train_f1_macro	0.851852
test_f1_micro	0.984615
train_f1_micro	1.0
test_hamming	0.001443
train_hamming	0.0

Figura 18. Cross validation training scores para Label Powerset (genbase)