Análisis y aprendizaje automático

Universidad Europea

LAUREATE INTERNATIONAL UNIVERSITIES

Máster en Big Data Analytics

Clasificación usando orange

in [PhD. Esteban-García-Cuesta](#)

Profesor

in [Christian Sucuzhanay Arévalo](#)

Alumno

Clasificación usando

Introducción.

Este documento refleja paso a paso la realización de la práctica sobre clasificación, usando diferente modelos en el programa Orange

El objetivo es observar el funcionanmiento del programa además de generar un documento indicando las pruebas que se han realizado y determinar el mejor modelo obtenido, interpretando los resultados.

Los datasets que analizare son:

1. **zoo.tab**
2. **emotions.tab**

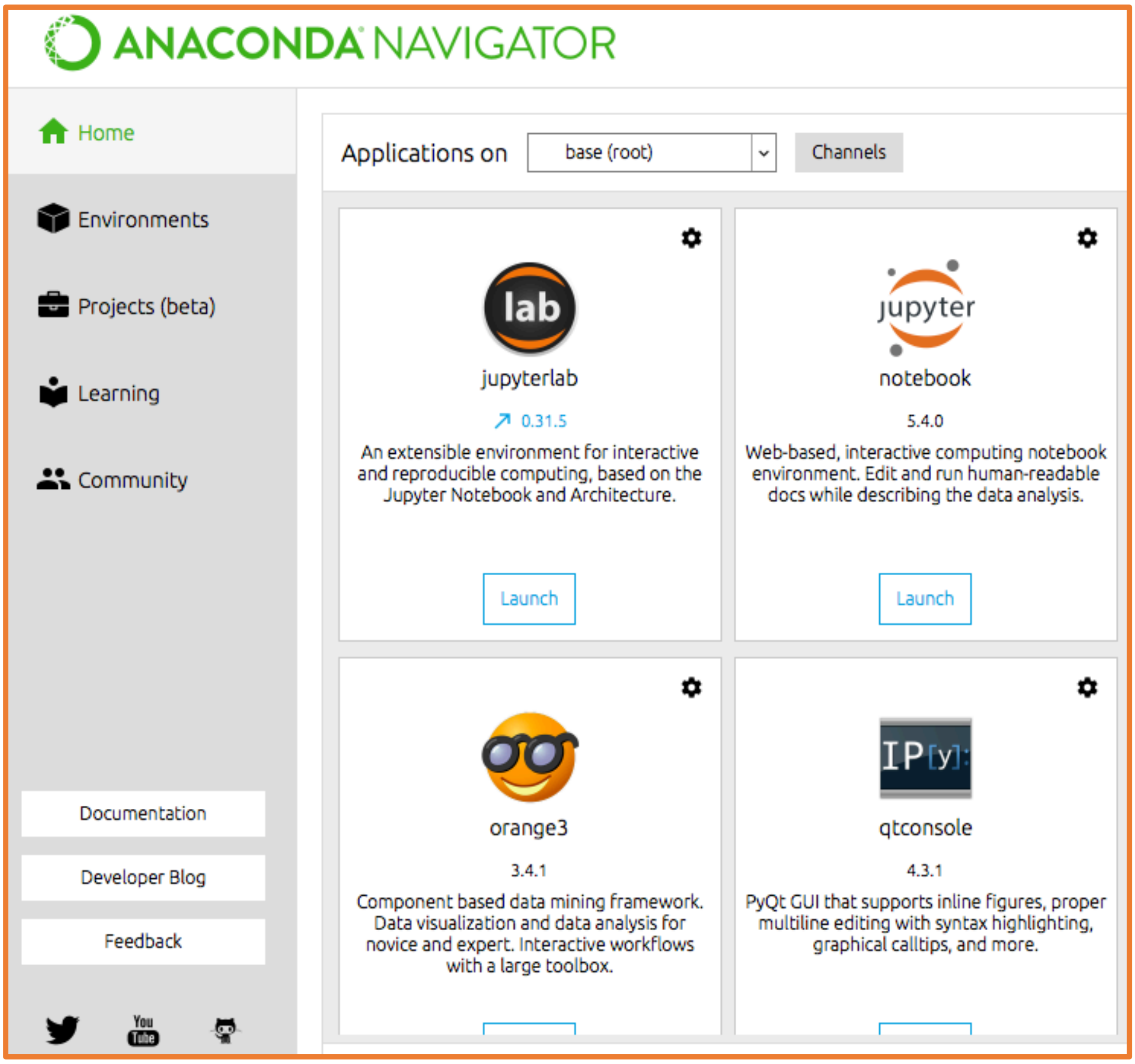
Las herramientas que he utilizado son:

1. Anaconda.- framework con varias herramientas para análisis.
2. Orange.- interfaz grafica incluida en la distribución de Anaconda

Todos los ficheros y demás librería esta en mi repositorio de GitHub

Configuraciones e Instalaciones

Instalación de Anaconda.



An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

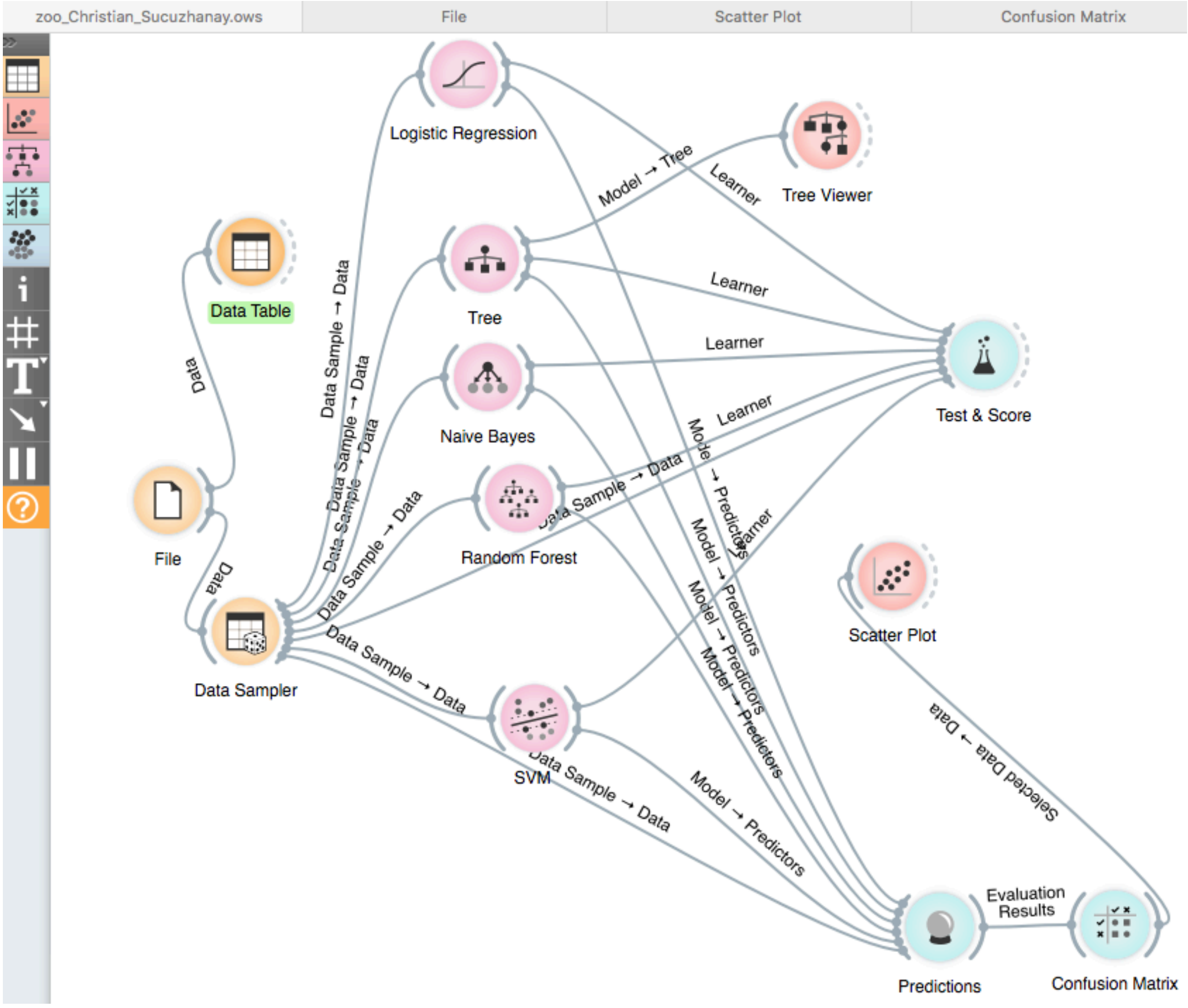
Seleccionamos el programa Orange y arrastramos los componentes necesarios al workspace.

Debemos tener clara la ruta de los datasets para que el programa pueda acceder a ellos y procesarlos.

ZOO.tab

Probando modelos de clasificación

Como se puede observar en la grafica he probado diferentes modelos



De la ejecución de los modelos, 5 MODELOS usando 70% para el training, 20% para el test y 10% para validación he obtenidos los siguientes resultados, como se pueden apreciar en los siguientes informes.

Evaluando los resultados

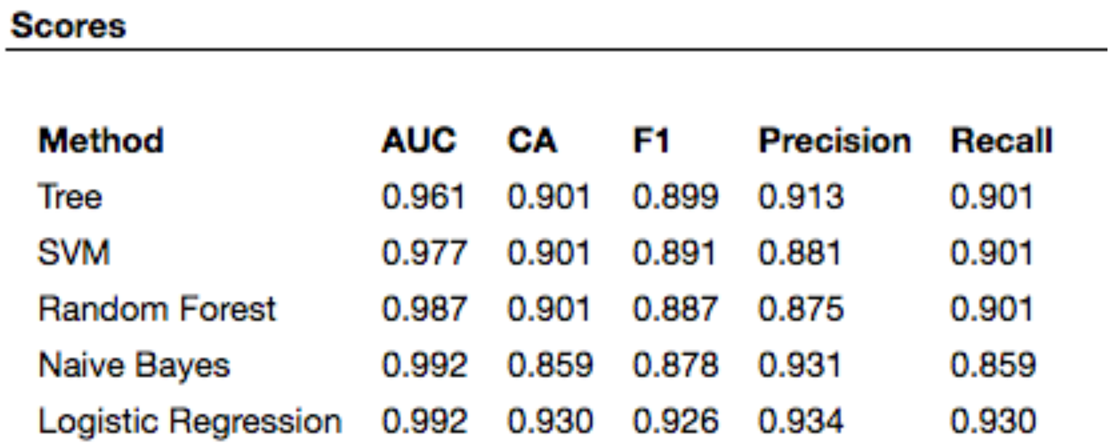
El mejor modelo

Como se puede observar en la tabla de SCORES claramente **LOGISTIC REGRESION** es el mejor de todos los modelos en todas las métricas empleadas.

El peor modelo

En todas las métricas **NAIVE BAYES** es el peor de todos, su propio nombre lo indica es un algoritmo probabilístico ingenuo.

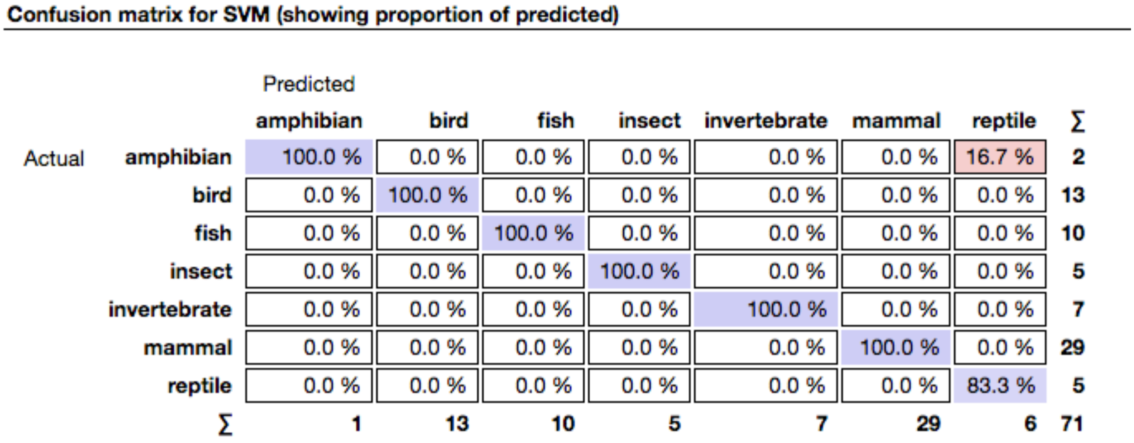
SCORES



Scores

Method	AUC	CA	F1	Precision	Recall
Tree	0.961	0.901	0.899	0.913	0.901
SVM	0.977	0.901	0.891	0.881	0.901
Random Forest	0.987	0.901	0.887	0.875	0.901
Naive Bayes	0.992	0.859	0.878	0.931	0.859
Logistic Regression	0.992	0.930	0.926	0.934	0.930

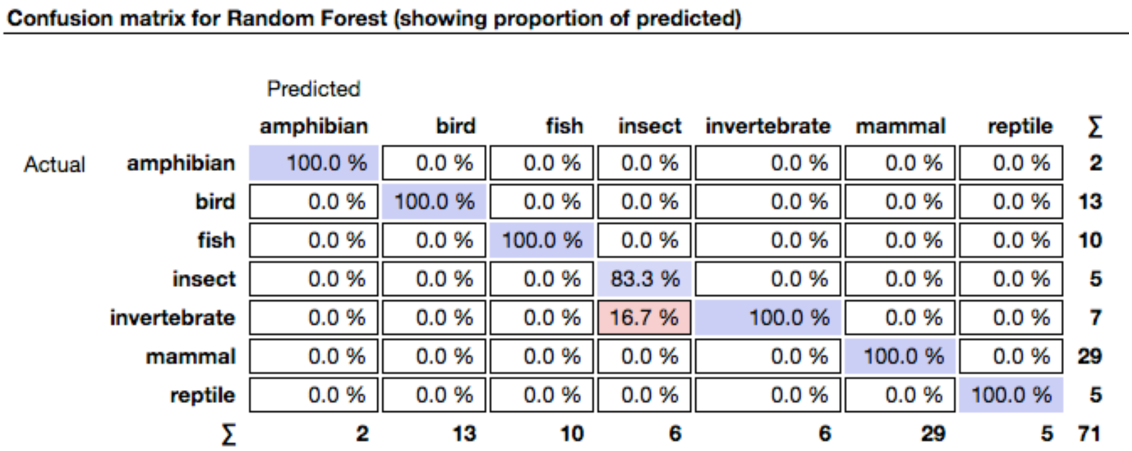
SVM



Confusion matrix for SVM (showing proportion of predicted)

		Predicted							
		amphibian	bird	fish	insect	invertebrate	mammal	reptile	∑
Actual	amphibian	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	16.7 %	2
	bird	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	13
	fish	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10
	insect	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	5
	invertebrate	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	7
	mammal	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	29
	reptile	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	83.3 %	5
	∑	1	13	10	5	7	29	6	71

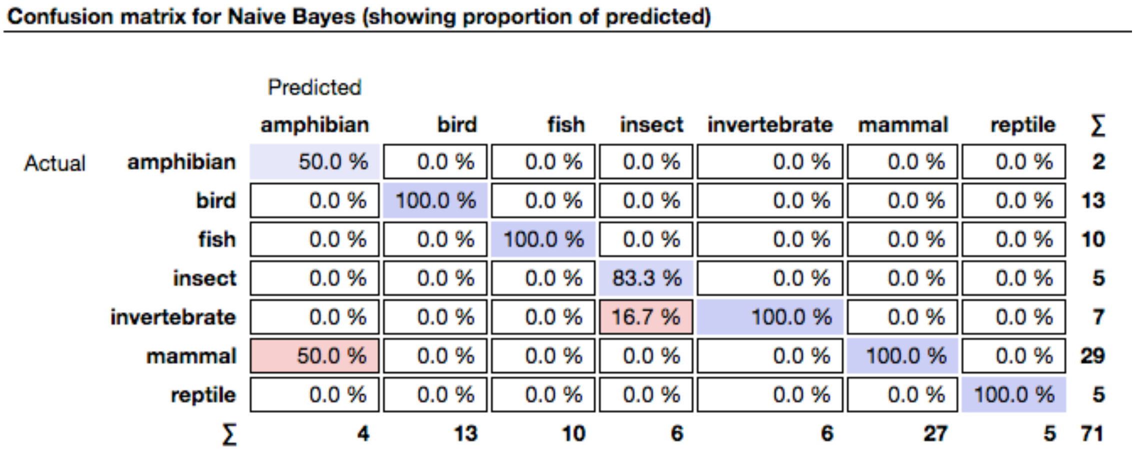
RANDOM FOREST



Confusion matrix for Random Forest (showing proportion of predicted)

		Predicted							
		amphibian	bird	fish	insect	invertebrate	mammal	reptile	∑
Actual	amphibian	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	2
	bird	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	13
	fish	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10
	insect	0.0 %	0.0 %	0.0 %	83.3 %	0.0 %	0.0 %	0.0 %	5
	invertebrate	0.0 %	0.0 %	0.0 %	16.7 %	100.0 %	0.0 %	0.0 %	7
	mammal	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	29
	reptile	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	5
	∑	2	13	10	6	6	29	5	71

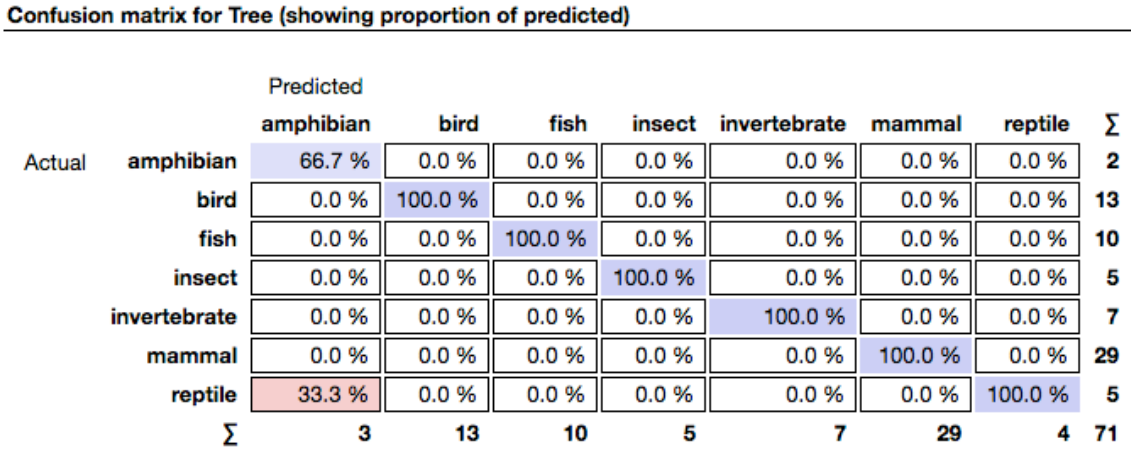
NAIVE BAYES



Confusion matrix for Naive Bayes (showing proportion of predicted)

		Predicted							
		amphibian	bird	fish	insect	invertebrate	mammal	reptile	∑
Actual	amphibian	50.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	2
	bird	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	13
	fish	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10
	insect	0.0 %	0.0 %	0.0 %	83.3 %	0.0 %	0.0 %	0.0 %	5
	invertebrate	0.0 %	0.0 %	0.0 %	16.7 %	100.0 %	0.0 %	0.0 %	7
	mammal	50.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	29
	reptile	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	5
	∑	4	13	10	6	6	27	5	71

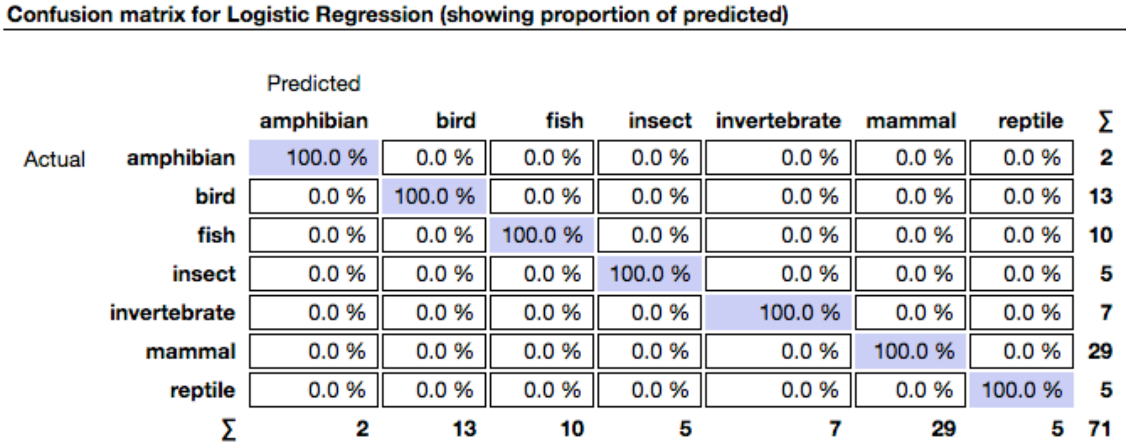
DECISION TREE



Confusion matrix for Tree (showing proportion of predicted)

		Predicted							
		amphibian	bird	fish	insect	invertebrate	mammal	reptile	∑
Actual	amphibian	66.7 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	2
	bird	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	13
	fish	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10
	insect	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	5
	invertebrate	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	7
	mammal	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	29
	reptile	33.3 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	5
	∑	3	13	10	5	7	29	4	71

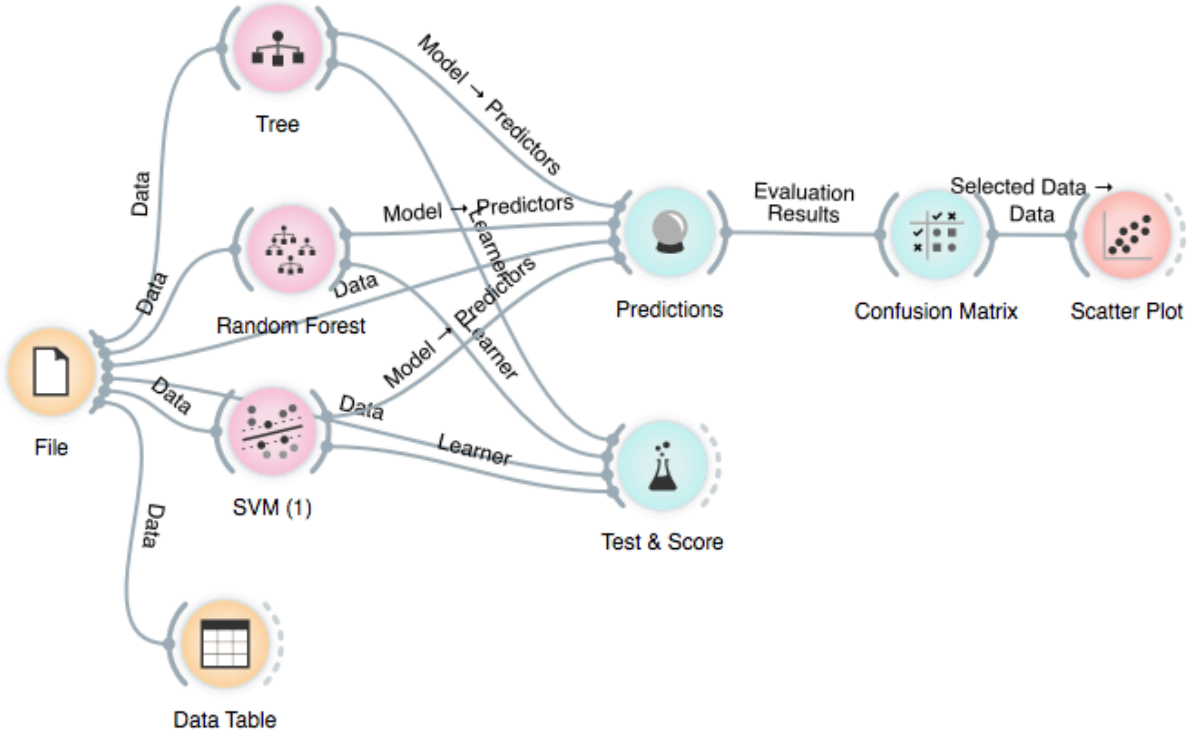
LOGISTIC REGRESION



Confusion matrix for Logistic Regression (showing proportion of predicted)

		Predicted							
		amphibian	bird	fish	insect	invertebrate	mammal	reptile	∑
Actual	amphibian	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	2
	bird	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	13
	fish	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10
	insect	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	5
	invertebrate	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	7
	mammal	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	29
	reptile	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	5
	∑	2	13	10	5	7	29	5	71

EMOTIONS.tab



Para analizar emotions.tab primero se ha realizado un proceso de transformación de los datos REEMPLAZANDO las variables amazed-surprised, happy-pleased, relaxing-calm, quiet-still, sad lonely y angry aggresive por letras. He utilizado 3 modelos de clasificación:

1. Random Forest.
2. Decision Tree.
3. SVM.

Evaluando los resultados

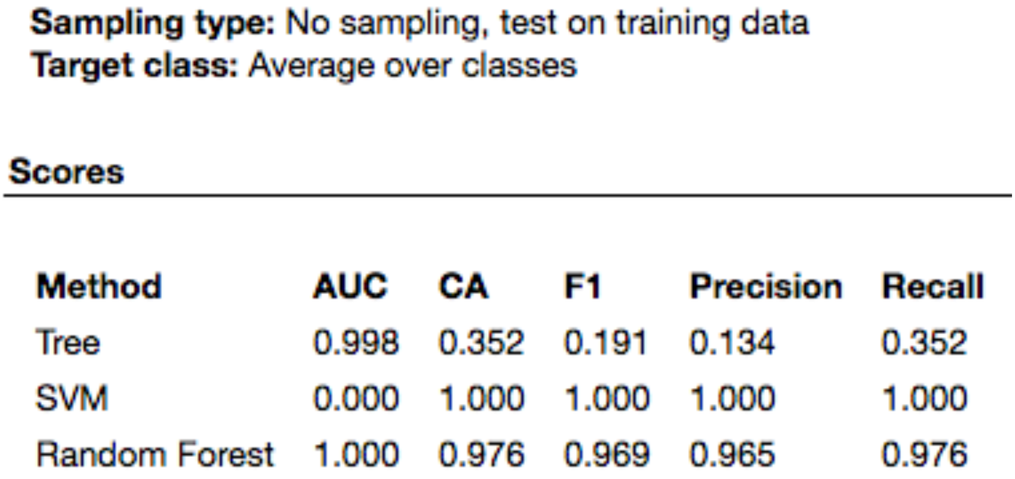
El mejor modelo

Como se puede observar en la tabla de SCORES claramente **SVM** es el mejor de todos los modelos en todas las métricas empleadas,

El peor modelo

En todas las métricas **DECISIÓN TREE** es el peor de todos los modelos, en mi opinión es que no se maneja bien con problemas MULTI-LABEL.

SCORES

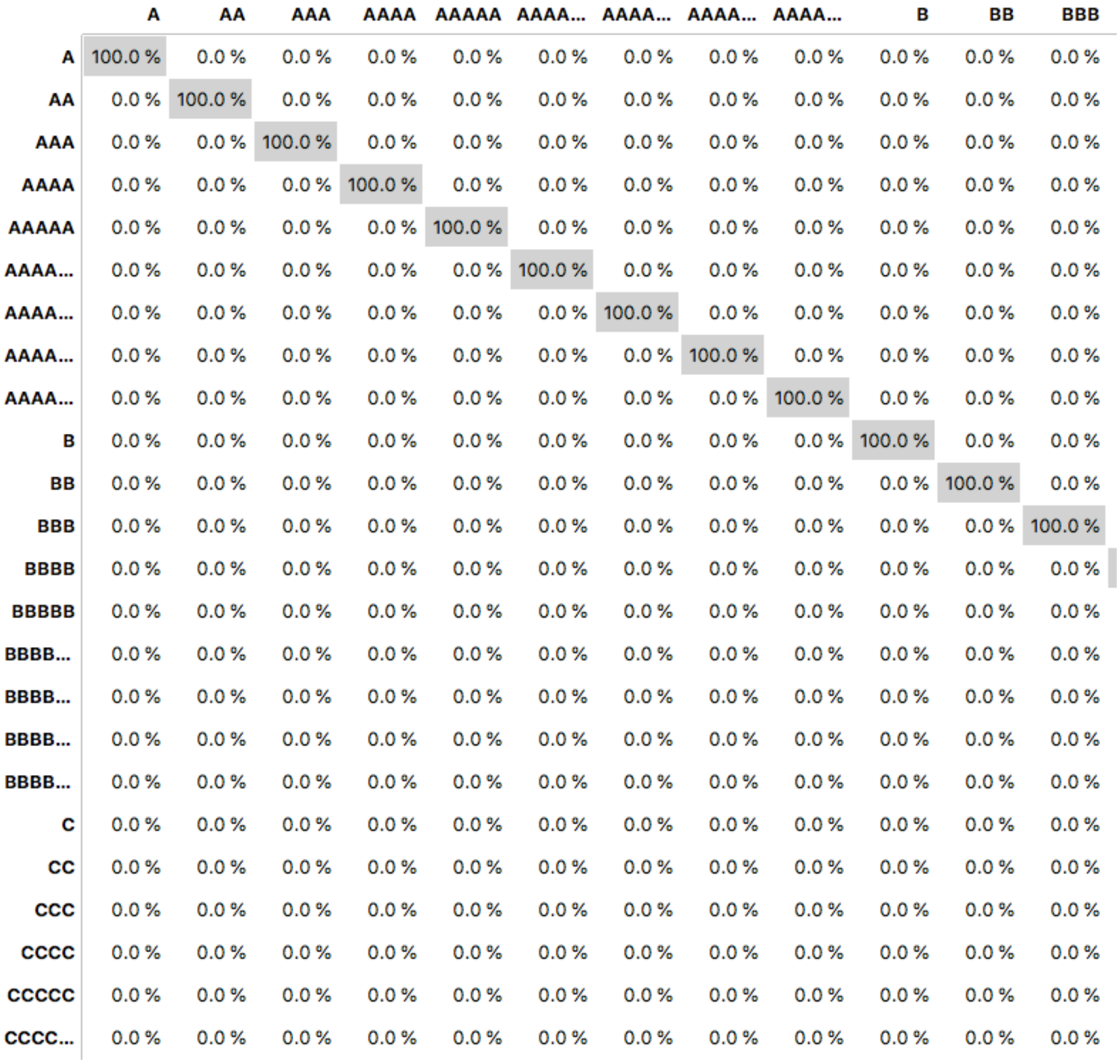


Sampling type: No sampling, test on training data
Target class: Average over classes

Scores

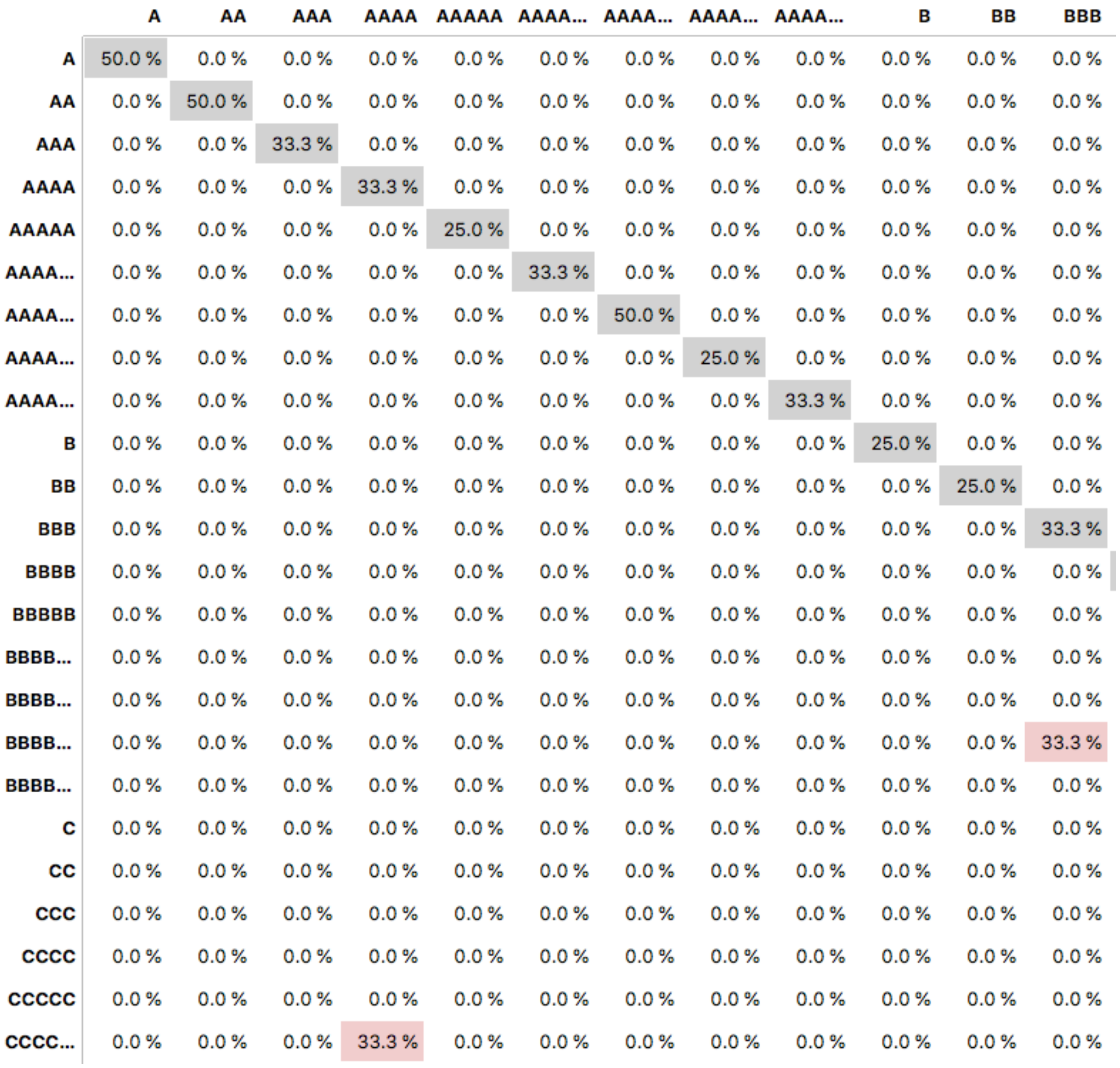
Method	AUC	CA	F1	Precision	Recall
Tree	0.998	0.352	0.191	0.134	0.352
SVM	0.000	1.000	1.000	1.000	1.000
Random Forest	1.000	0.976	0.969	0.965	0.976

SVM



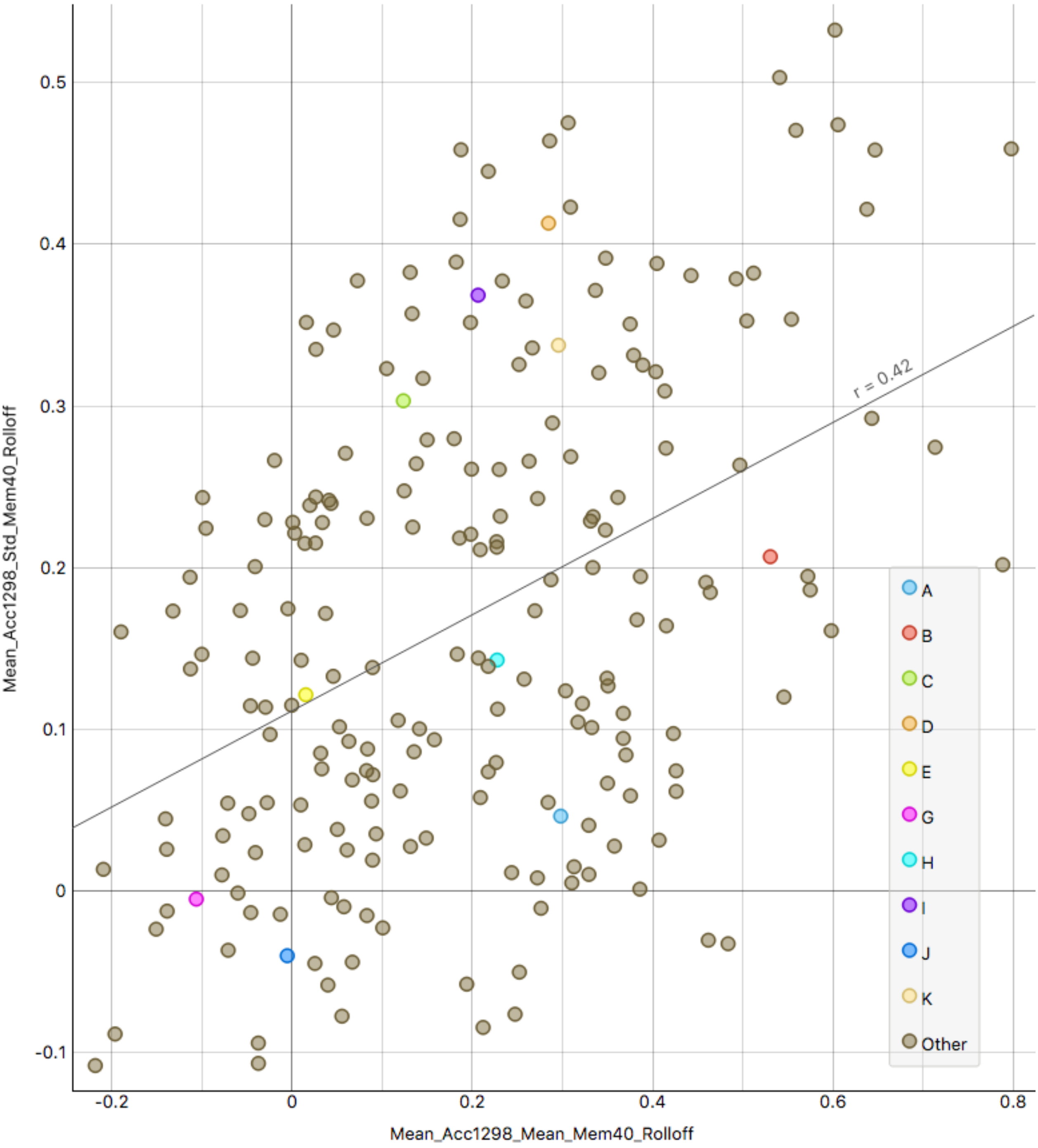
	A	AA	AAA	AAAA	AAAAA	AAAA...	AAAA...	AAAA...	AAAA...	B	BB	BBB
A	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AA	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAA	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAAA	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %
B	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %
BB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %
BBB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %
BBBB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBBB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
C	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCCC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCCCC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCCC...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %

DECISIÓN TREE



	A	AA	AAA	AAAA	AAAAA	AAAA...	AAAA...	AAAA...	AAAA...	B	BB	BBB
A	50.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AA	0.0 %	50.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAA	0.0 %	0.0 %	33.3 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA	0.0 %	0.0 %	0.0 %	33.3 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAAA	0.0 %	0.0 %	0.0 %	0.0 %	25.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	33.3 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	50.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	25.0 %	0.0 %	0.0 %	0.0 %	0.0 %
AAAA...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	33.3 %	0.0 %	0.0 %	0.0 %
B	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	25.0 %	0.0 %	0.0 %
BB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	25.0 %	0.0 %
BBB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	33.3 %
BBBB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBBB	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	33.3 %
BBBB...	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
C	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCCC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCCCC	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
CCCC...	0.0 %	0.0 %	0.0 %	33.3 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %

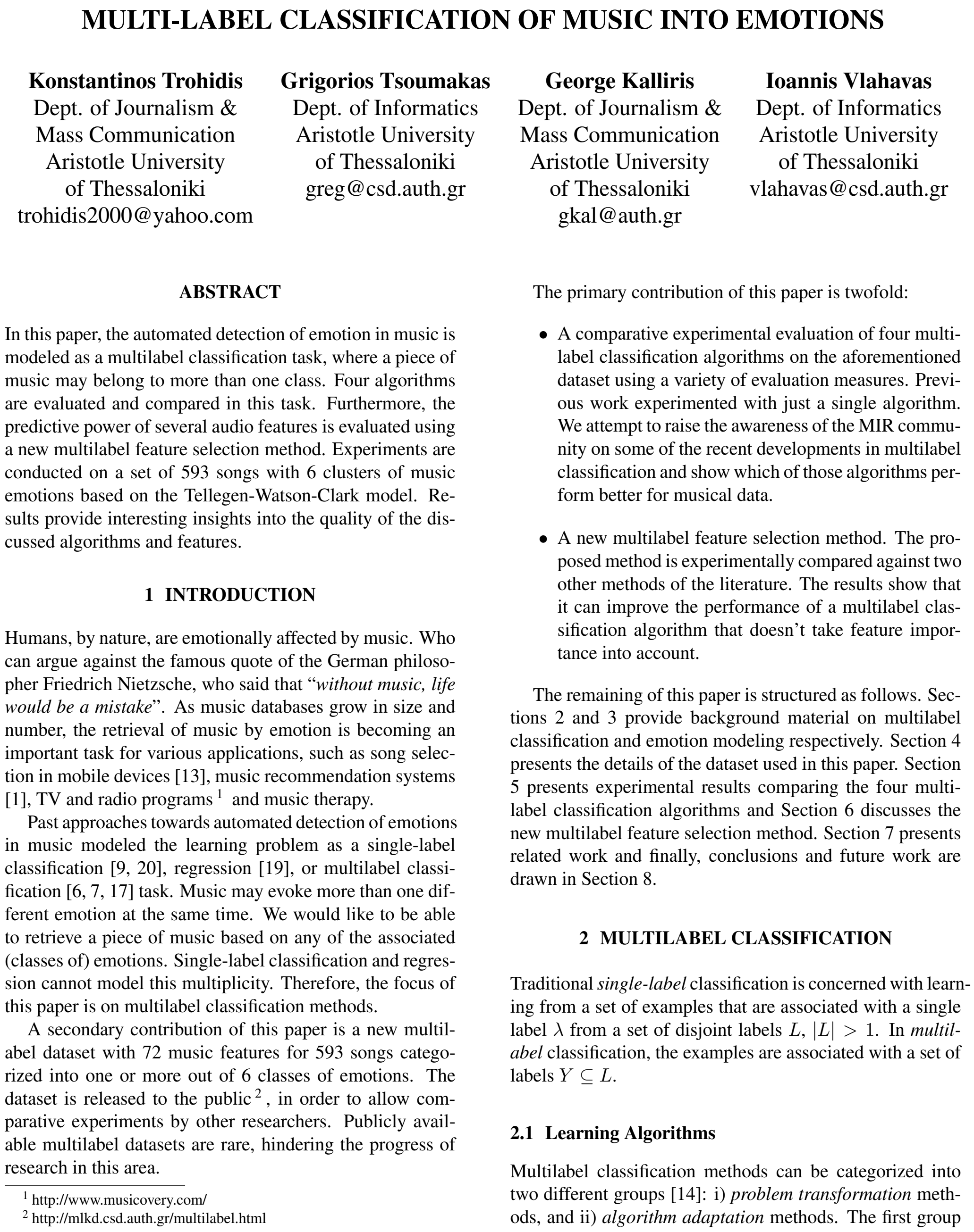
PLOT DE EMOTIONS



Conclusiones de emotions.

Con los modelos utilizados no logro obtener una clasificación coherente, además para obtener resultados obttimos debería primero normalizar o darle pesos, es un problema de MULTILABEL

PAPER ABOUT EMOTIONS



MULTI-LABEL CLASSIFICATION OF MUSIC INTO EMOTIONS

Konstantinos Trohidis
Dept. of Journalism &
Mass Communication
Aristotle University
of Thessaloniki
trohidis2000@yahoo.com

Grigorios Tsoumakas
Dept. of Informatics
Aristotle University
of Thessaloniki
greg@csd.auth.gr

George Kalliris
Dept. of Journalism &
Mass Communication
Aristotle University
of Thessaloniki
gkal@auth.gr

Ioannis Vlahavas
Dept. of Informatics
Aristotle University
of Thessaloniki
vlahavas@csd.auth.gr

ABSTRACT

In this paper, the automated detection of emotion in music is modeled as a multilabel classification task, where a piece of music may belong to more than one class. Four algorithms are evaluated and compared in this task. Furthermore, the predictive power of several audio features is evaluated using a new multilabel feature selection method. Experiments are conducted on a set of 593 songs with 6 clusters of music emotions based on the Tellegen-Watson-Clark model. Results provide interesting insights into the quality of the discussed algorithms and features.

1 INTRODUCTION

Humans, by nature, are emotionally affected by music. Who can argue against the famous quote of the German philosopher Friedrich Nietzsche, who said that "*without music, life would be a mistake*". As music databases grow in size and number, the retrieval of music by emotion is becoming an important task for various applications, such as song selection in mobile devices [13], music recommendation systems [1], TV and radio programs [1] and music therapy.

Past approaches towards automated detection of emotions in music modeled the learning problem as a single-label classification [9, 20], regression [19], or multilabel classification [6, 7, 17] task. Music may evoke more than one different emotion at the same time. We would like to be able to retrieve a piece of music based on any of the associated (classes of) emotions. Single-label classification and regression cannot model this multiplicity. Therefore, the focus of this paper is on multilabel classification methods.

A secondary contribution of this paper is a new multilabel dataset with 72 music features for 593 songs categorized into one or more out of 6 classes of emotions. The dataset is released to the public [2], in order to allow comparative experiments by other researchers. Publicly available multilabel datasets are rare, hindering the progress of research in this area.

The primary contribution of this paper is twofold:

- A comparative experimental evaluation of four multilabel classification algorithms on the aforementioned dataset using a variety of evaluation measures. Previous work experimented with just a single algorithm. We attempt to raise the awareness of the MIR community on some of the recent developments in multilabel classification and show which of those algorithms perform better for musical data.
- A new multilabel feature selection method. The proposed method is experimentally compared against two other methods of the literature. The results show that it can improve the performance of a multilabel classification algorithm that doesn't take feature importance into account.

The remaining of this paper is structured as follows. Sections 2 and 3 provide background material on multilabel classification and emotion modeling respectively. Section 4 presents the details of the dataset used in this paper. Section 5 presents experimental results comparing the four multilabel classification algorithms and Section 6 discusses the new multilabel feature selection method. Section 7 presents related work and finally, conclusions and future work are drawn in Section 8.

2 MULTILABEL CLASSIFICATION

Traditional *single-label* classification is concerned with learning from a set of examples that are associated with a single label λ from a set of disjoint labels L, $|L| > 1$. In *multilabel* classification, the examples are associated with a set of labels $Y \subseteq L$.

2.1 Learning Algorithms

Multilabel classification methods can be categorized into two different groups [14]: i) *problem transformation* methods, and ii) *algorithm adaptation* methods. The first group

[1] http://www.musicovery.com/
[2] http://mlkd.csd.auth.gr/multilabel.html

contains methods that are algorithm independent. They transform the multilabel classification task into one or more single-label classification, regression or ranking tasks. The second group contains methods that extend specific learning algorithms in order to handle multilabel data directly.

2.2 Evaluation Measures

Multilabel classification requires different evaluation measures than traditional single-label classification. A taxonomy of multilabel classification evaluation measures is given in [15], which considers two main categories: *example-based* and *label-based measures*. A third category of measures, which is not directly related to multilabel classification, but is often used in the literature, is ranking-based measures, which are nicely presented in [21] among other publications.

3 MUSIC AND EMOTION

Hevner [4] was the first to study the relation between music and emotion. She discovered 8 clusters of adjective sets describing music emotion and created an emotion cycle of these categories. Hevner's adjectives were refined and regrouped into ten groups by Farnsworth [2].

Figure 1 shows another emotion model, called Thayer's model of mood [12], which consists of 2 axes. The horizontal axis described the amount of stress and the vertical axis the amount of energy.

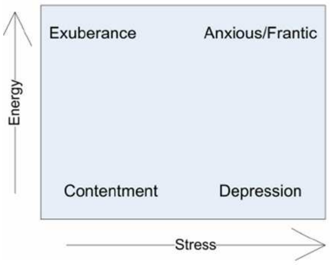


Figure 1. Thayer's model of mood

The model depicted in Figure 2 extends Thayer's model with a second system of axes, which is rotated by 45 degrees compared to the original axes [11]. The new axes describe (un)pleasantness versus (dis)engagement.

4 DATASET

The dataset used for this work consists of 100 songs from each of the following 7 different genres: Classical, Reggae, Rock, Pop, Hip-Hop, Techno and Jazz. The collection was created from 233 musical albums choosing three songs from each album. From each song a period of 30 seconds after the initial 30 seconds was extracted. The resulting sound clips were stored and converted into wave files of 22050 Hz sampling rate, 16-bit per sample and mono. The following subsections present the features that were extracted from each wave file and the emotion labeling process.

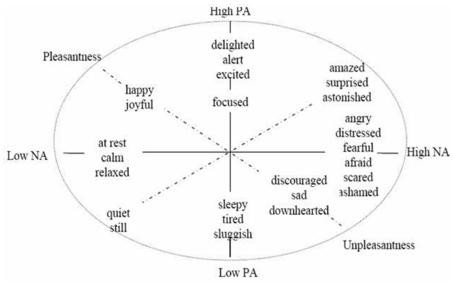


Figure 2. The Tellegen-Watson-Clark model of mood (figure reproduced from [18])

4.1 Feature Extraction

For the feature extraction process, the Marsyas tool [16] was used. The extracted features fall into two categories: rhythmic and timbre.

4.1.1 Rhythmic Features

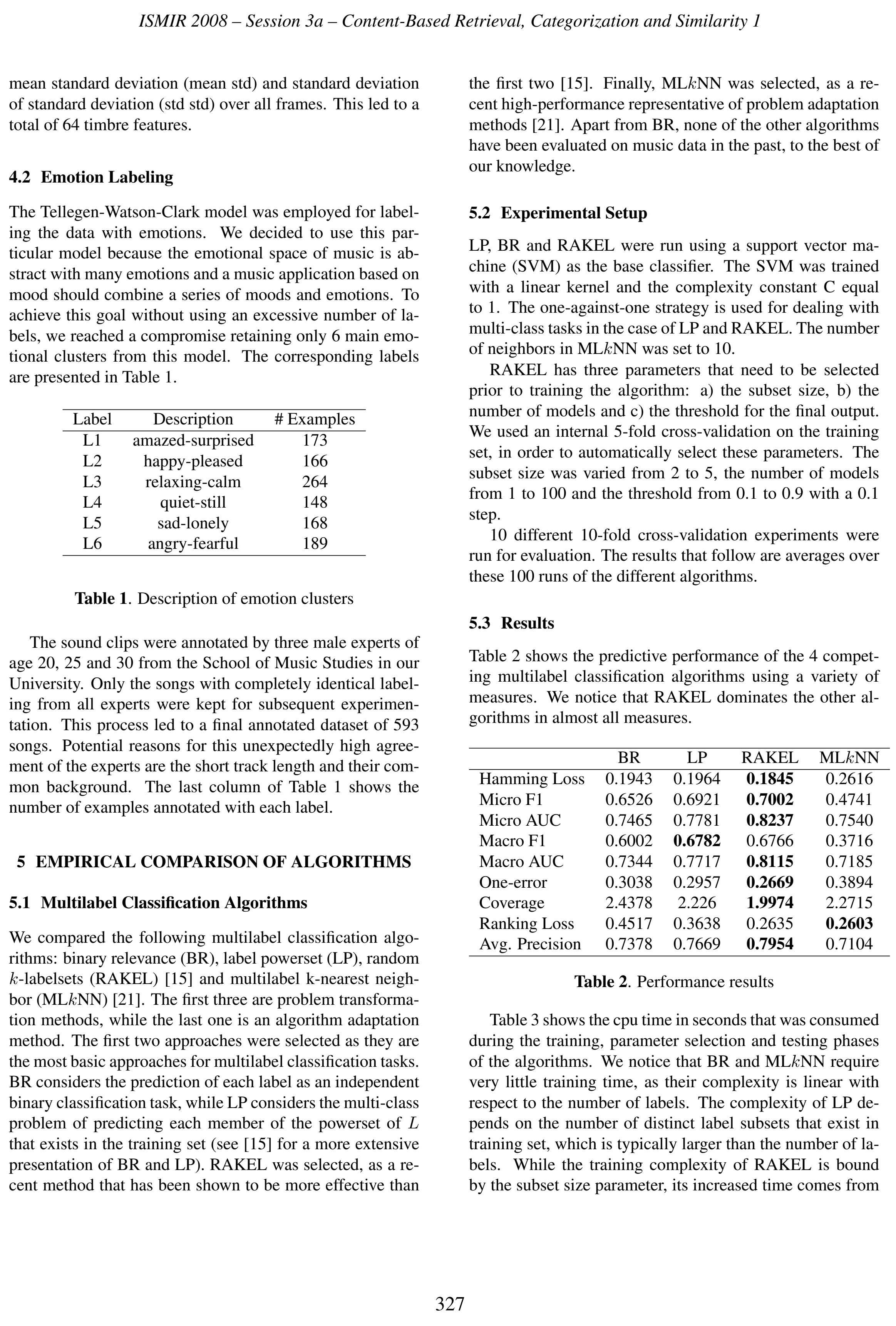
The rhythmic features were derived by extracting periodic changes from a beat histogram. An algorithm that identifies peaks using autocorrelation was implemented. We selected the two highest peaks and computed their amplitudes, their BMPs (beats per minute) and the high-to-low ratio of their BPMs. In addition, 3 features were calculated by summing the histogram bins between 40-90, 90-140 and 140-250 BPMs respectively. The whole process led to a total of 8 rhythmic features.

4.1.2 Timbre Features

Mel Frequency Cepstral Coefficients (MFCCs) are used for speech recognition and music modeling [8]. To derive MFCCs features, the signal was divided into frames and the amplitude spectrum was calculated for each frame. Next, its logarithm was taken and converted to Mel scale. Finally, the discrete cosine transform was implemented. We selected the first 13 MFCCs.

Another set of 3 features that relate to timbre textures were extracted from the Short-Term Fourier Transform (FFT): Spectral centroid, spectral rolloff and spectral flux.

For each of the 16 aforementioned features (13 MFCCs, 3 FFT) we calculated the mean, standard deviation (std),



mean standard deviation (mean std) and standard deviation of standard deviation (std std) over all frames. This led to a total of 64 timbre features.

4.2 Emotion Labeling

The Tellegen-Watson-Clark model was employed for labeling the data with emotions. We decided to use this particular model because the emotional space of music is abstract with many emotions and a music application based on mood should combine a series of moods and emotions. To achieve this goal without using an excessive number of labels, we reached a compromise retaining only 6 main emotional clusters from this model. The corresponding labels are presented in Table 1.

Label	Description	# Examples
L1	amazed-surprised	173
L2	happy-pleased	166
L3	relaxing-calm	264
L4	quiet-still	148
L5	sad-lonely	168
L6	angry-fearful	189

Table 1. Description of emotion clusters

The sound clips were annotated by three male experts of age 20, 25 and 30 from the School of Music Studies in our University. Only the songs with completely identical labeling from all experts were kept for subsequent experimentation. This process led to a final annotated dataset of 593 songs. Potential reasons for this unexpectedly high agreement of the experts are the short track length and their common background. The last column of Table 1 shows the number of examples annotated with each label.

5 EMPIRICAL COMPARISON OF ALGORITHMS

5.1 Multilabel Classification Algorithms

We compared the following multilabel classification algorithms: binary relevance (BR), label powerset (LP), random k-labelsets (RAKEL) [15] and multilabel k-nearest neighbor (MLkNN) [21]. The first three are problem transformation methods, while the last one is an algorithm adaptation method. The first two approaches were selected as they are the most basic approaches for multilabel classification tasks. BR considers the prediction of each label as an independent binary classification task, while LP considers the multi-class problem of predicting each member of the powerset of L that exists in the training set (see [15] for a more extensive presentation of BR and LP). RAKEL was selected, as a recent method that has been shown to be more effective than the first two [15]. Finally, MLkNN was selected, as a recent high-performance representative of problem adaptation methods [21]. Apart from BR, none of the other algorithms have been evaluated on music data in the past, to the best of our knowledge.

5.2 Experimental Setup

LP, BR and RAKEL were run using a support vector machine (SVM) as the base classifier. The SVM was trained with a linear kernel and the complexity constant C equal to 1. The one-against-one strategy is used for dealing with multi-class tasks in the case of LP and RAKEL. The number of neighbors in MLkNN was set to 10.

RAKEL has three parameters that need to be selected prior to training the algorithm: a) the subset size, b) the number of models and c) the threshold for the final output. We used an internal 5-fold cross-validation on the training set, in order to automatically select these parameters. The subset size was varied from 2 to 5, the number of models from 1 to 100 and the threshold from 0.1 to 0.9 with a 0.1 step.

10 different 10-fold cross-validation experiments were run for evaluation. The results that follow are averages over these 100 runs of the different algorithms.

5.3 Results

Table 2 shows the predictive performance of the 4 competing multilabel classification algorithms using a variety of measures. We notice that RAKEL dominates the other algorithms in almost all measures.

	BR	LP	RAKEL	MLkNN
Hamming Loss	0.1943	0.1964	**0.1845**	0.2616
Micro F1	0.6526	0.6921	**0.7002**	0.4741
Micro AUC	0.7465	0.7781	**0.8237**	0.7540
Macro F1	0.6002	**0.6782**	0.6766	0.3716
Macro AUC	0.7344	0.7717	**0.8115**	0.7185
One-error	0.3038	0.2957	**0.2669**	0.3894
Coverage	2.4378	2.226	**1.9974**	2.2715
Ranking Loss	0.4517	0.3638	0.2635	**0.2603**
Avg. Precision	0.7378	0.7669	**0.7954**	0.7104

Table 2. Performance results

Table 3 shows the cpu time in seconds that was consumed during the training, parameter selection and testing phases of the algorithms. We notice that BR and MLkNN require very little training time, as their complexity is linear with respect to the number of labels. The complexity of LP depends on the number of distinct label subsets that exist in training set, which is typically larger than the number of labels. While the training complexity of RAKEL is bound by the subset size parameter, its increased time comes from