

Clasificador de géneros musicales

Gabriel Álvarez 09-10029 - Francisco Martínez 09-10502

Abstracto - La clasificación de géneros musicales es una tarea realizada primordialmente por los humanos ya que está caracterizada por subjetividad, con los avances en el aprendizaje de máquinas y la inteligencia artificial en general se ha buscado que más de estas tareas que parecieran ser realizadas únicamente por humanos sean realizadas de manera determinística con las computadoras. Nuestro equipo a través de distintas técnicas ha generado un conjunto de datos de cuatro géneros distintos (metal, jazz, rap y electrónica) y ha conseguido un nivel de predicción del 97,3% respecto al género musical. Este documento discute las técnicas y tareas realizadas para llegar a este resultado.

I. INTRODUCCIÓN

Con el avance de la tecnología más tareas que se pensaban que solo podían ser realizadas por humanos son realizadas por computadoras, de estas tareas que aun hoy en día solo pueden ser realizadas por los humanos a causa de su subjetividad y nivel de abstracción, las cuales son cualidades difíciles de expresar en código y algoritmos de manera explícita, el aprendizaje de máquinas ofrece una solución permitiendo aprender de cada caso visto y de estos poder generalizar conceptos.

El problema que se desea resolver es el de poder clasificar correctamente los géneros musicales en una canción dada. Actualmente es posible automatizar esta tarea gracias a el aprendizaje de máquinas, ya que de lo contrario, al ser un tema subjetivo, no se podría hacer un algoritmo que predijera correctamente el género de una canción; el aprendizaje de máquinas facilita esta tarea y le brinda objetividad al tema.

Una de las dificultades de este problema es el tipo de datos que se debe utilizar como ejemplos para el conjunto de entrenamiento. Estos datos, al ser canciones, poseen la particularidad de ser continuos y variar en el tiempo, lo cual presenta una mayor complejidad a la hora de obtener los atributos de cada canción, este tipo de datos se denomina "time series data" [2]. Pero gracias a las herramientas que se poseen hoy en día, no se tuvo que realizar el análisis respectivo, ya que fue posible obtener toda esta metadata de las canciones a través del API de Echonest [3]. El cual te permite hacer "requests" a su servidor para obtener toda la información de las canciones que se le especifiquen. Cabe destacar que esta información la obtuvieron utilizando el método de MFCC.

En este trabajo se utilizó el software Orange [4] el cual posee una gran cantidad de herramientas para aprendizaje de máquinas, entre estas herramientas se puede seleccionar el algoritmo que se va a utilizar y en nuestro caso realizamos corridas con todos los algoritmos que el software posee, estos algoritmos son: "Random Forest", "SVM", "Naive Bayes", "Logistic Regression", "Nearest Neighbors" y "Classification Tree", en los siguientes puntos se presentarán los resultados obtenidos.

II. MARCO TEÓRICO

1. Algoritmos:

Random Forest: Un algoritmo de aprendizaje de máquinas que está basado en la técnica de "random decision forest", el cual corrige el hábito de "decision tree" de hacer "overfitting" sobre el conjunto de entrenamiento [5].

SVM: También denominado “Support vector machine” es un modelo de aprendizaje supervisado, el cual tiene algoritmos de aprendizaje para clasificación y análisis de regresión [6].

Naive Bayes: Perteneciente a la familia de clasificadores probabilísticos, los cuales están basados en el teorema de Bayes [7].

Logistic Regression: Es un tipo de análisis de regresión utilizado para predecir una variable categórica en función de variables independientes [8].

Nearest Neighbors: Es un método de clasificación supervisada que sirve para estimar una función de densidad, es un método de clasificación del tipo no-paramétrico [9].

Classification Tree: Es un modelo de predicción el cual utiliza reglas lógicas que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema [9].

2. Herramientas:

Echonest: El API que contiene toda la metadata de las canciones que fueron utilizadas como conjunto de entrenamiento y prueba [3].

Orange: Una aplicación la cual posee una gran variedad de herramientas para data mining. Los cuales incluyen diversos algoritmos, graficadores, entre otras [4]

Pyechonest: Una librería de python la cual fue utilizada para los requests de canciones a Echonest.

3. Conceptos generales:

MFCC: Utilizado para la representación del sonido en coeficientes, los cuales contienen los datos de las canciones [10]

HDF5: Es un formato de archivo el cual es utilizado para almacenar grandes cantidades de datos.

III. DISEÑO DE LA SOLUCIÓN

Como fue mencionado anteriormente, los pasos que se siguieron para obtener la solución fueron los siguientes.

Usar un script en python el cual usa la librería Pyechonest para hacer requests al servidor de Echonest y obtener la metadata de las canciones y generar un archivo de texto el cual contiene el formato adecuado

para la ser usado por la aplicación Orange.

Una vez obtenido el conjunto de entrenamiento, se procedió a utilizar Orange, y se probaron cada uno de los algoritmos que esta posee para comparar resultados.

Las políticas empleadas se centran principalmente en los géneros y los “features” que fueron escogidos para cada canción, ya que habían mucha información en la metadata que no fue considerada como importante. Ej: el nombre de las canciones. Además, se escogieron cuatro géneros “bastante diferentes” entre sí, los cuales son: metal, jazz, rap y electronic. Los “features” que fueron escogidos son los siguientes:

Artist: nombre del artista. (discreto)

Danceability: Un coeficiente que representa que tan bailable es una canción. Mientras más cercano a 1.0, más bailable. (continuo)

Duration: La duración de la canción. (continuo)

Energy: Medida porcentual de intensidad. (continuo)

Key: La clave de la canción. (discreto)

Liveness: Detecta la presencia de una audiencia en la canción. (continuo)

Loudness: Que tan fuerte suena la canción en promedio. (continuo)

Mode: El modo de la canción (discreto)

Speechiness: El porcentaje de palabras que son dichas en una canción. (continuo)

Tempo: Es la velocidad promedio de la canción medida en BPM (Beats per minute). (discreto)

Time-Signature: La métrica rítmica de la canción. (discreto)

Genre: Representa el género de la canción. El atributo usado para clasificar.

IV. IMPLEMENTACIÓN

El proyecto fue realizado con diferentes herramientas las cuales fueron explicadas en el marco teórico de este documento, inicialmente nuestro objetivo era utilizar el conjunto de datos “Million Song Dataset”, este conjunto posee una librería de funciones que te permite obtener la información técnica de cada canción, la cual tiene que ser leída por este método ya que el conjunto de datos está escrito en un formato de compresión llamado HDF5; para esto tuvimos que instalar multiples librerías, pero

el conjunto de datos no posee una característica sumamente primordial para nuestro proyecto el cual es el género, aunque el conjunto de datos tiene unos valores de identificación, los cuales puedes utilizar en conjunto de otros dos API (Musicbrainz - The Echonest), muchos de los seriales de identificación dados por el conjunto de datos se encontraban desactualizados en relación a estos dos API, por lo cual se nos complicó en gran medida la obtención del género para cada canción, además de esto había varios valores indefinidos en los datos para una proporción de las canciones.

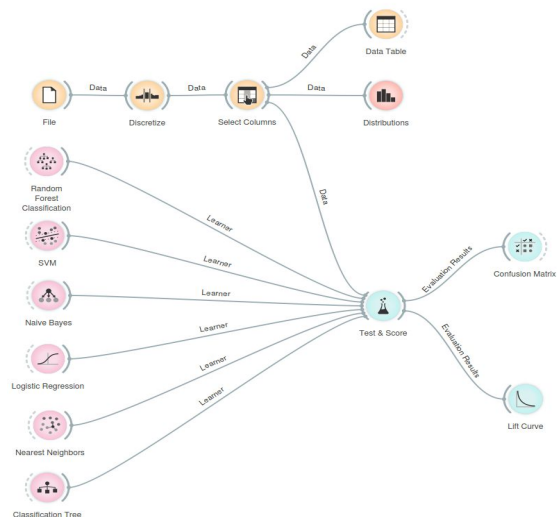
Finalmente para simplificar el proceso decidimos crear nuestro propio conjunto de datos, el cual tendría además de estar actualizado, el hecho de que las canciones estarían ya agrupadas por género, permitiéndonos escoger los géneros que quisiéramos clasificar.

La obtención de las canciones fue realizada únicamente con el API de Echonest utilizando una de sus librerías oficiales (Pyechnest) la cual está escrita en Python, con esta librería se realizó un programa donde pedimos listas de reproducción de canciones de cuatro géneros distintos (metal ,jazz, rap y electronic) , estos géneros fueron escogidos de manera tal que cada género entre si fuera lo más distinto posible de los demás para así mejorar el resultado de los algoritmos, cada canción en las listas de reproducción escogidas contienen ya de por sí una cantidad de metadatos, permitiendo el aprendizaje de la canción.

El set generado fue de 3600 canciones, donde hay 915 canciones aproximadamente para cada género, este conjunto de datos es balanceado por lo que no habrá algún tipo de inclinación en el algoritmo hacia algún género. Las características que posee una canción ya fueron explicadas antes, pero se recalca el hecho de que al ser muchas de estas valores continuos estos fueron transformados en valores discretos con una de las herramientas de Orange, la cual permite transformar los datos de continuos a discretos y viceversa, luego de esto, se dividió el conjunto de 3600 canciones en una proporción de 85% para casos de entrenamiento y de 15% para el conjunto de prueba (separados de manera aleatoria), estos conjuntos fueron utilizados por igual para cada algoritmo y cada algoritmo fue corrido 2 veces, siendo los resultados un promedio de estas

corridas. A continuación se muestra una imagen que ejemplifica gráficamente el proceso realizado para obtener los resultados :

Vista de Implementación

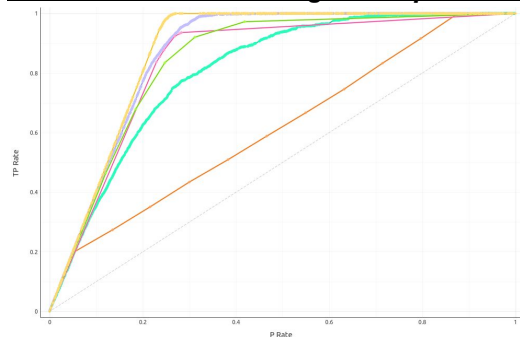


V. EXPERIMENTOS Y RESULTADOS

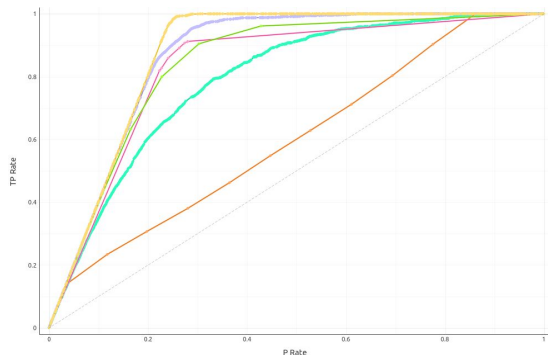
La experimentación consistió en realizar 2 corridas de conjuntos creados a partir del conjunto de las 3600 canciones, los conjuntos fueron creados aleatoriamente en una proporción de 85% y 15%. A continuación se muestran las gráficas que representan la evolución del aprendizaje (True Positives) de cada algoritmo con respecto a la población (Population) :

- Random Forest : Aquamarino
- SVM : Naranja oscuro
- Naive Bayes : Morado
- Classification Tree : Rosado
- k-Nearest Neighbors : Verde
- Logistic Regression : Amarillo

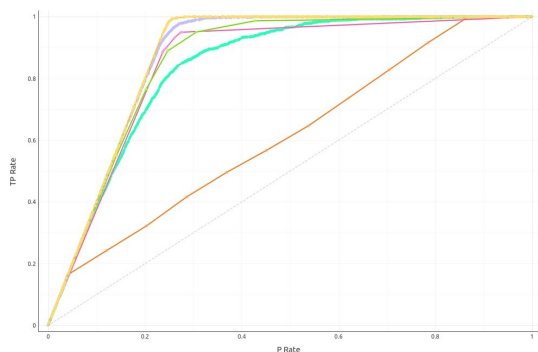
Clasificación de cada algoritmo para Jazz



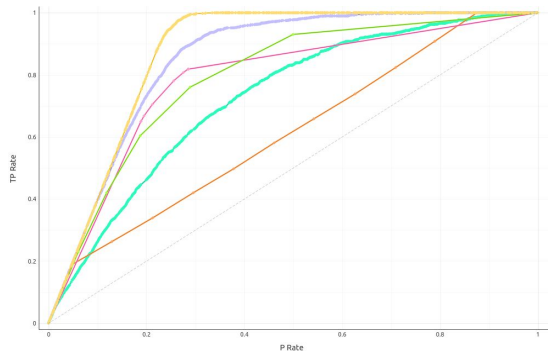
Clasificación de cada algoritmo para Metal



Clasificación de cada algoritmo para Rap



Clasificación de cada algoritmo para Electronic



A continuación se muestra una tabla con los resultados de cada algoritmo para facilitar su comparación :

- Método : Algoritmo utilizado
 - RF: Random Forest
 - SVM :Support Vector Machine
 - NB : Naive Bayes
 - CT : Classification Tree
 - kNN: k nearest neighbors

- LR : Logistic Regression
- CA : “Classification Accuracy”
- F1 : Promedio entre “precision” y “recall”.
- P : Proporción de predicciones correctas entre las predicciones
- Recall : Proporciones de predicciones correctas respecto a los valores reales.

Método	CA	F1	P	Recall
RF	0.678	0.676	0.675	0.678
SVM	0.722	0.730	0.783	0.722
NB	0.898	0.898	0.898	0.898
CT	0.860	0.860	0.861	0.817
kNN	0.817	0.816	0.816	0.817
LR	0.973	0.973	0.973	0.973

Finalmente se muestra una tabla que permite observar los resultados específicos del algoritmo “Logistic Regression”, el cual dio mejores resultados (97,3%), en esta tabla se puede observar cuántas de las veces que dijo que era electrónica (la columna en el extremo izquierdo) predijo correctamente como electrónica, y así con los otros tres géneros.

	Predicted				Σ
	electronic	jazz	metal	rap	
electronic	97.8 %	2.3 %	1.8 %	2.6 %	1361
jazz	0.8 %	96.6 %	0.4 %	0.1 %	1343
metal	0.9 %	0.9 %	97.7 %	0.1 %	1343
rap	0.5 %	0.3 %	0.1 %	97.2 %	1353
Σ	1299	1372	1349	1380	5400

VI. CONCLUSIÓN

Se puede concluir que los resultados fueron satisfactorios, el algoritmo de “Logic Regression” logró casi 100% de precisión, aunque en gran parte una de las decisiones que más influyó en estos resultados fue la escogencia de géneros que “no tienen mucho en común”. El artículo en el que nos inspiramos logró un 61% de

precisión con una mayor cantidad de géneros [1].

Se puede argumentar también que es posible realizar una clasificación de géneros mas completa, incluyendo mas generos, tambien es posible mejorar el algoritmo si se incluye un mayor número de características al realizar el aprendizaje.

REFERENCIAS

[1] Aaron Kravitz, Eliza Lupone, Ryan Diaz. "From Classical To Hip-Hop: Can Machines Learn Genres?". Disponible en Internet: <http://cs229.stanford.edu/proj2014/Aaron%20Kravitz,%20Eliza%20Lupone,%20Ryan%20Diaz,%20Can%20Machines%20Learn%20Genres.pdf>. Consultado el 5 de febrero de 2016.

[2] Wikipedia "Time series". Disponible en Internet: https://en.wikipedia.org/wiki/Time_series. Consultado el 5 de febrero de 2016.

[3] Echonest. Disponible en Internet: <http://the.echonest.com>. Consultado el 5 de febrero de 2016.

[4] Orange Data Mining. Disponible en Internet: <http://orange.biolab.si>. Consultado el 5 de febrero de 2016.

[5] Wikipedia "Random forest". Disponible en Internet: https://en.wikipedia.org/wiki/Random_forest. Consultado el 5 de febrero de 2016.

[6] Wikipedia "Support vector machine". Disponible en Internet: https://en.wikipedia.org/wiki/Support_vector_machine. Consultado el 5 de febrero de 2016.

[7] Wikipedia "Naive Bayes classifier". Disponible en Internet: https://en.wikipedia.org/wiki/Naive_Bayes_classifier. Consultado el 5 de febrero de 2016.

[8] Wikipedia "Logistic regression". Disponible en Internet: https://en.wikipedia.org/wiki/Logistic_regression. Consultado el 5 de febrero de 2016.

[9] Wikipedia "K-nearest neighbors algorithm". Disponible en Internet: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. Consultado el 5 de febrero de 2016.

[10] Wikipedia "Decision-tree". Disponible en Internet: https://en.wikipedia.org/wiki/Decision_tree. Consultado el 5 de febrero de 2016.

[11] Wikipedia "Mel-frequency cepstrum". Disponible en Internet: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum. Consultado el 5 de febrero de 2016.

