

Music recommendation

Raquel Leandra Pérez Arnal i Adrián Sánchez Albanell

Contents

1	Descripción del trabajo	3
1.1	Introducción	3
1.2	Conjunto de datos disponible	3
1.3	Notas sobre el lenguaje de programación escogido	4
2	Trabajo Relacionado	4
3	Data exploration and Preprocessing	4
3.1	Tratamiento de valores perdidos	5
3.2	Tratamiento de outliers	6
3.3	Tratamiento de valores incorrectos	6
3.4	Codificación de variables categóricas	6
3.5	Selección de features	6
3.6	Creación de nuevas variables	6
3.7	Estandarización	6
3.8	Transformación de variables	6
4	Resampling Protocol	6
5	Resultados de los métodos lineales	6
6	Resultados de los métodos no lineales	6
7	Descripción y justificación del modelo escogido	6
8	Conclusiones	6

1 Descripción del trabajo

1.1 Introducción

Este trabajo consiste en elegir un problema de regresión o clasificación y generar un modelo para resolverlo. Para ello usaremos algunos de los métodos, lineales y no lineales, vistos en clase durante el curso de Aprendizaje Autónomo.

Hemos elegido un problema de [kaggle competitions](#) sobre recomendación de música llamado [WSDM - KKBox's Music Recommendation Challenge](#). Este consiste en un dataset, proporcionado por [KKBOX](#) - servicio de streaming de música asiático - con información sobre diferentes canciones, usuarios y como ha sido el acceso de los usuarios a dichas canciones.

El objetivo es predecir si un usuario que ha escuchado una canción lo volverá a hacer en un periodo de tiempo determinado, por lo tanto se trata de un problema de clasificación binaria: si el usuario volverá a escuchar o no una canción que ya ha oído anteriormente.

1.2 Conjunto de datos disponible

Kaggle nos ha proporcionado los datos en seis ficheros CSV, de los cuales usaremos cuatro para la práctica. Los dos restantes son un conjunto de datos de muestra sobre como enviar los datos para el concurso y los datos de test para el concurso (que no nos sirven ya que vienen sin la variable target).

train.csv

Contiene la información de las reproducciones de canciones por parte del usuario. Tiene las siguientes variables:

msno	identificador del usuario.
song_id	identificador de la canción.
source_system_tab	nombre de la pestaña donde se selecciono el evento. Ejemplos: <i>my library</i> , <i>search</i> , etc.
source_screen_name	nombre de la pantalla que ve el usuario.
source_type	des de donde se ha reproducido la canción. Ejemplos: <i>album</i> , <i>online-playlist</i> , <i>song</i> , etc.
target	variable de target. Si el usuario ha escuchado la canción más de una vez en un intervalo de un mes target es 1, si no es 0.

members.csv

Contiene información de los usuarios. Tiene las siguientes variables:

msno	identificador del usuario.
city	identificador de ciudad.
bd	edad del usuario. Contiene valores outlier.
gender	genero del usuario. Puede ser <i>female</i> o <i>male</i> .
registered_via	identificador del método de registro de usuario.
registration_init_time	día del registro de usuario, en formato <i>%Y%m%d</i> .
expiration_date	día de expiración del registro de usuario, en formato <i>%Y%m%d</i> .

songs.csv

Tiene tamaño: (2,286,220 , 7)

Contiene información de las canciones. Tiene las siguientes variables:

song_id	identificador de la canción.
song_length	duración de la canción en milisegundos.
genre_ids	género musical de la canción. Hay canciones con más de un genero, donde el carácter hace de separador.
artist_name	nombre del artista.
composer	nombre del compositor o compositores. Si hay más de uno el carácter hace de separador.
lyricist	nombre del escritor o escritores de la canción. Si hay más de uno el carácter hace de separador.
language	identificador del lenguaje de la canción.

song_extra_info.csv

Contiene información extra de las canciones. Tiene las siguientes variables:

song_id	identificador de la canción.
song_name	nombre de la canción.
isrc	International Standard Recording Code . En teoría se puede usar como identificador de la canción, pero hay codigos ISRC sin verificar. Contiene información de la canción aunque puede ser erronea o confusa como el country code, que no se refiere a la canción si no a la agencia que proporciona el codigo ISRC.

1.3 Notas sobre el lenguaje de programación escogido

2 Trabajo Relacionado

3 Data exploration and Preprocessing

Los datos iniciales son:

- train.csv
- test.csv
- songs.csv
- members.csv
- song_extra_info.csv

La idea inicial es mejorar train utilizando songs, members y song_extra_info y convertirlo en un único train y test.

Después pasar todas las variables categóricasa numéricas y aplicarle un MCA.

Debería quedar:

- clean_train.csv
- clean_test.csv

3.1 Tratamiento de valores perdidos

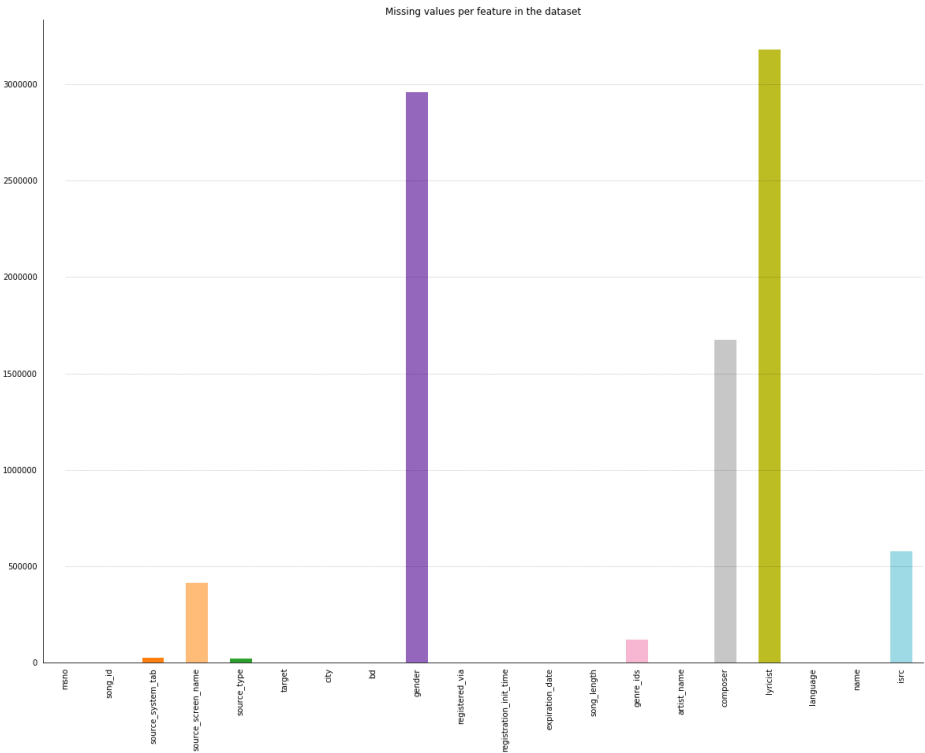


Figure 1: En esta gráfica podemos ver la cantidad de valores perdidos que tiene el conjunto de datos final

Table 1: My caption

Nombre	Porcentage
msno	0
song_id	0
source_system_tab	0.3368
source_screen_name	5.6226
source_type	0.2919
target	0
city	0
bd	0
gender	40.142
registered_via	0
registration_init_time	0
expiration_date	0
song_length	0.0015
genre_ids	1.605
artist_name	0.0015
composer	22.7139
lyricist	43.0882
language	0.0020
name	0.0197
isrc	7.8327

Viendo esta información descartamos *lyricist* y *composer*

En el resto de variables eliminamos las muestras con missings salvo en *gender*.

Tratamiento de los valores perdidos en *gender*.

Los imputamos utilizando la clasificación que nos daría un knn.

3.2 Tratamiento de outliers

3.3 Tratamiento de valores incorrectos

3.4 Codificación de variables categóricas

3.5 Selección de features

Table 2: Nos quedaremos con estas variables para el análisis

Nombre	Percentage
source_system_tab	0.3368
source_screen_name	5.6226
source_type	0.2919
target	0
city	0
bd	0
gender	40.142
registered_via	0
registration_init_time	0
expiration_date	0
song_length	0.0015
genre_ids	1.605

3.6 Creación de nuevas variables

3.7 Estandarización

3.8 Transformación de variables

4 Resampling Protocol

5 Resultados de los métodos lineales

logistic regression, multinomial regression (single-layer MLP), LDA, QDA, RDA, **Naive Bayes**, **nearest-neighbours**, **linear SVM**, quadratic SVM

6 Resultados de los métodos no lineales

one-hidden-layer MLP, the RBFNN, the SVM with RBF kernel, a Random Forest

7 Descripción y justificación del modelo escogido

8 Conclusiones