

# Hackathon Instructions

26 October 2019

## Data files

- 1) songs\_train.csv (training data)
- 2) songs\_test\_without\_target.csv (test data)
- 3) songs\_test\_target\_predictions\_example.csv (format of the results file)

## Data description

Each row contains data related to a song, including its musical characteristics and information about its original performer (singer).

## What you need to do

**All your work must be done in Jupyter Notebook.**

- 1) Build a regression model that predicts the *song\_rating* attribute from the other song's attributes, using the data in file "songs\_train.csv" and **mean absolute error** (MAE) as a primary performance metric to evaluate your model. **You can only use the models imported below.**
- 2) Use your model to predict the *song\_rating* of the songs in file "songs\_test\_without\_target.csv" .
- 3) Submit the .ipynb file(s) containing the source code of ALL your experimental work.
- 4) Submit a file called "songs\_test\_target\_predictions.csv" that contains ONE column of the predictions of the test file. You can do this using the code below.

*Sample code:*

```
df = pd.Series(test_predictions)
df.name = 'song_rating'
df.to_csv('./songs_test_target_predictions.csv', index=False, float_format="%.3f", header=True)
```

(where test\_predictions is a 1-D array variable that contains your predicted values)

## Helpful imports

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor, GradientBoostingRegressor

from sklearn.metrics import mean_absolute_error
```

## Some Hints

- scoring parameter for cross-validation : **scoring='neg\_mean\_absolute\_error'**
- start with a very simple approach first: clean the data but do not do any feature engineering such as scaling and log normalization. You can try more complicated things if you have time.
- In order to save time in the beginning, do data preparation on all data (i.e. do not split data before data preparation step).
- if an attribute is too complicated or requires too much cleaning work (especially text attributes), then just remove it! You can deal with it later if you have time ...
- try different manual feature selection strategies.