Predicciones JESC 2020 con base en características del audio y series de tiempo

1. J. Oswaldo Gomez M
Universidad Cuauhtémoc
osw.gom@gmail.com

Abstract

Se analizaron las canciones que competirán en el *Junior Eurovision Song Contest 2020* utilizando las APIs de Spotify en donde obtuvimos ritmo, tono, timbre, duración entre otros, para entrenar múltiples modelos de aprendizaje automático y luego combinarlos en un modelo de *Soft Voting* para obtener mejores resultados. A su vez, utilizamos un modelo aditivo generalizado para predecir los puntajes del JESC 2020 y combinamos ambos resultados para predecir que el ganador estará entre España, Bielorrusia, Holanda y Polonia (en este orden).

Introducción

En general, una manera de pronosticar un variable es encontrar otra variable relacionada que ocurra un par de intervalos de tiempo antes, entre más grande sea la correlación y entre más grande sea el tiempo de ventaja, mejor será la estrategia.

Una estrategia es hacer extrapolaciones con base en que las tendencias actuales continúen y en implementar estimados adaptativos a estas tendencias. Por otro lado, lo que intentamos predecir es qué canción ganará *Junior Eurovision Song Contest 2020*, por lo que entrenamos un modelo de clasificación de aprendizaje automático supervisado (*Supervised Machine Learning*), con base en el puntaje obtenido de canciones de concursos pasados y los APIs de Spotify *Search* y *Tracks*, el primero para encontrar la canción dentro de la base de datos de Spotify y la última para enriquecer nuestro conjunto de entrenamiento con análisis de audio (ritmo, tono, y timbre) y características de audio (que tan bailable es, volumen, energía, golpes por minuto, etc). Al combinar la tendencia de los últimos años con un análisis de la canción misma, podemos hacer un pronóstico más robusto al plantear la teoría que debe haber algún patrón, poco evidente para el ser humano, que hace que los jueces califiquen una canción como buena o mala.

Desarrollo

Para generar nuestro conjunto de datos de entrenamiento, se minó la página de Wikipedia para descargar los puntajes desde 2003 hasta 2019 del concurso JESC 2020. Con esto generamos una tabla en formato CSV por cada año, en donde obtuvimos: artista, canción, país, puntos, entre otras. Utilizamos artista y canción para llamar al API de Spotify Search con el que obtuvimos el ID de la canción en la base de datos de Spotify. Con esto, logramos finalmente obtener datos de análisis y características del audio llamando al Spotify Tracks API. Una vez construído nuestro conjunto de entrenamiento, entrenamos 15 modelos diferentes de aprendizaje autónomo

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.6100	0.6111	0.6000	0.7167	0.6176	0.2166	0.2488	0.9940
knn	K Neighbors Classifier	0.6133	0.6361	0.5667	0.7000	0.6100	0.2308	0.2447	0.2170
lr	Logistic Regression	0.6500	0.6444	0.7000	0.6667	0.6714	0.2828	0.3027	0.0130
rf	Random Forest Classifier	0.5900	0.5694	0.6000	0.6583	0.5957	0.1857	0.2175	0.7280
ridge	Ridge Classifier	0.6300	0.0000	0.6667	0.6500	0.6448	0.2507	0.2693	0.0090
lda	Linear Discriminant Analysis	0.6300	0.6444	0.6667	0.6500	0.6448	0.2507	0.2693	0.0100
et	Extra Trees Classifier	0.5167	0.5722	0.5667	0.6000	0.5481	0.0217	0.0447	0.6730
nb	Naive Bayes	0.5933	0.7056	0.4667	0.5833	0.4900	0.2088	0.2170	0.0100
dt	Decision Tree Classifier	0.5433	0.5333	0.5500	0.5833	0.5414	0.0783	0.1005	0.0100
gbc	Gradient Boosting Classifier	0.5000	0.5500	0.6000	0.5567	0.5395	0.0117	0.0339	0.0550
lightgbm	Light Gradient Boosting Machine	0.5400	0.6306	0.7000	0.5533	0.6017	0.0474	0.0500	0.0160
ada	Ada Boost Classifier	0.5167	0.6000	0.6000	0.5483	0.5405	0.0182	0.0204	0.0750
svm	SVM - Linear Kernel	0.5533	0.0000	0.5667	0.5400	0.5371	0.1179	0.1167	0.0100
qda	Quadratic Discriminant Analysis	0.5967	0.6667	0.4667	0.5333	0.4800	0.2029	0.2168	0.0100
xgboost	Extreme Gradient Boosting	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0050

Figura 1 Entrenamiento de 15 modelos de aprendizaje autónomo

Como se puede observar, los mejores tres fueron: *CatBoost Classifier*, *K Neigbhors* y *Logistic Regression*. Posteriormente, se afinaron los hiperparámetros y se mezclaron los tres modelos utilizando la funcion `blend_models` de Pycaret, que entrena un modelo de *Soft Voting*. Este modelo tiene dos métodos: *hard* y *soft*. El primero es por mayoría de votos (aquella predicción con mayor cantidad de votos gana) y en el segundo, las predicciones se

hacen al sumar las probabilidades de predicción de los modelos individuales, y eligiendo la predicción que tiene la mayor suma.

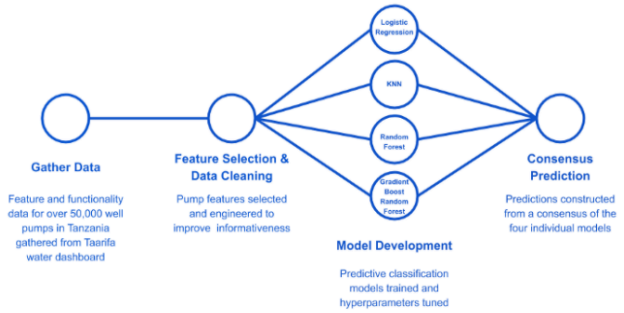


Figura 2 Entrenamiento de modelo de consenso via Soft Voting (Wyatt Sharber 2020)

Para la serie de tiempo, decidimos utilzar un método para pronosticar series de tiempo similar al modelo aditivo generalizado (Hastie & Tibshirani 1987).

$$y(\mathrm{t}) = g(\mathrm{t}) + s(\mathrm{t}) + h(t) + \epsilon_t$$

Con g(t) es la función de la tendencia que modela cambios no periódicos en el valor de la serie de tiempo, s(t) representa estacionalidad y h(t) modela los efectos de las vacaciones que pueden occurrir en momentos irregulares a lo largo de uno o mas días.

Resultados

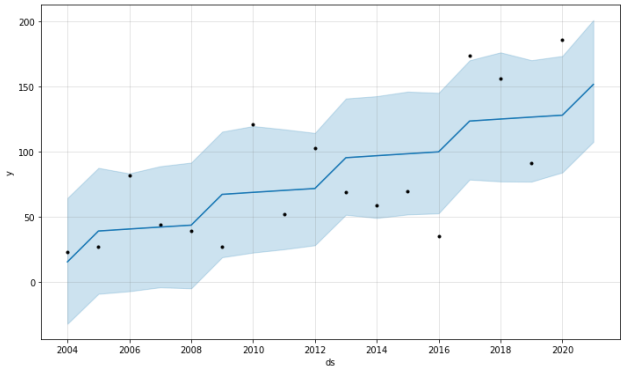
Proponemos una fórmula que aumente (disminuya) el puntaje en caso de tener una alta probabilidad asociada a la clasificación, pero que no afecte la serie de tiempo en caso de que esta sea cero.

$$\text{Puntos} = \text{pronostico\_serie}\left(1 \ +/- \ \frac{\text{prob\_predicción}}{2}\right)$$

Con lo anterior, arribamos a las siguientes predicciones:

	Pais	Serie de Tiempo	Categoría	Score	Prediction
0	España	258	Buena	0.7742	357.87180
1	Bielorrusia	183	Buena	0.5387	232.29105
2	Polonia	157	Mala	0.5529	200.40265
3	Holanda	151	Buena	0.7665	208.87075
4	Rusia	149	Mala	0.5233	187.98585

Es decir, estamos pronosticando que gane España, seguido de Bielorrusia y Holanda. A continuación veremos la serie de tiempo de Holanda, ya que había demasiados puntos de datos faltantes en España y Bielorrusia. El resto de las gráficas, el código y aprendizaje automático, minar Spotify y series de tiempo, se encuentra en este url (https://github.com/papagala/DataScienceMastersDegree/tree/master/time_series/notebooks).



Conclusión

Los resultados obtenidos por medio de series de tiempo y mediante un modelo de aprendizaje automático, son bastante similares como para desestimar alguna correlación y causalidad, es decir, parece que los paises que en años anteriores han obtenido un alto puntaje, tienen alta probabilidad de seguir llevando buenos artistas que se ve reflejado en una confiabilidad alta del modelo para etiquetarla como "buena". De manera cualitativa, me parece que los resultados del modelo de aprendizaje autónomo son bastante buenos, ya que las canciones que eligió como buenas, tienen un buen ritmo, son bailables y denotan postividad (elementos analizados por el modelo). Por otro lado, las que calificó como malas, tienen un ritmo más lento y no son muy bailables. Será sin duda interesante analizar estos resultados con base en la realidad, para ver posibles maneras de mejorar este modelo en miras hacia el JESC 2021.

References

[1] Search for an Item | Spotify for Developers. (2020). Retrieved 10 November 2020, from https://developer.spotify.com/documentation/web-api/reference/search/search/

[2] Soft Voting Classifier as a Consensus Method for Machine Learning Classification. (2020). Retrieved 10 November 2020, from https://medium.com/@wvsharber/soft-voting-classifier-as-a-consensus-method-for-machine-learning-classification-24ebd4d49943

[3] Taylor, S., & Letham, B. (2017). Forecasting at scale. doi: 10.7287/peerj.preprints.3190v2

[4] Hastie, T. & Tibshirani, R. (1987), 'Generalized additive models: some applications', Jour- nal of the American Statistical Association 82(398), 371–386.