Tecnológico de Monterrey

Módulo 2 Análisis y Reporte sobre el desempeño del modelo.

Inteligencia artificial avanzada para la ciencia de datos I

Alumno

Jackeline Conant Rubalcava A01280544

Profesores

Antonio Carlos Bento

Alfredo Esquivel Jaramillo

Mauricio González Soto

Julio Antonio Juárez Jiménez

Frumencio Olivas Álvarez

Jesús Adrián Rodríguez Rocha

Hugo Terashima Marín

07 de septiembre del 2024

Introducción a Base de Datos y Modelos de aprendizaje automático con árboles de decisión

Para esta actividad, utilicé una base de datos llamada “Dataset of Songs in Spotify”, en la cual se muestran varios aspectos sobre diferentes canciones y sus géneros. Por esta razón, se decidió crear modelos de predicción para determinar qué tipo de género se está escuchando.
La base de datos cuenta con 42305 registros y 22 columnas originalmente, donde se pueden ver:

Posteriormente se hizo una limpieza de los datos que serían apropiados para los modelos a utilizar y terminó con 12 columnas, de las cuales 11 serían las características y 1 sería la etiqueta.

Previo a implementar los valores de mi dataset, dividí mis características y etiquetas utilizando la función train_test_split. En esta división, el conjunto de entrenamiento representa el 80% de mi dataset, mientras que el conjunto de prueba representa el otro 20% restante. También se utilizó random_state para mezclar los datos y asegurarse de que no estuvieran en el orden original.

Ya que buscaba aprender más sobre los árboles de decisión para el aprendizaje automático, decidí crear 2 algoritmos para machine learning: Xgboost y Decision Tree.

Inicialmente, buscaba utilizar únicamente XGBoost, pero al ver la complejidad que conlleva, ya que utiliza muchos árboles de decisión, decidí también crear un

solo árbol de decisión y comparar los resultados de sesgo y varianza que obtendría con ambos modelos.

XGboost

Es un algoritmo de boosting basado en árboles de decisión. El boosting implica construir varios modelos de forma secuencial, donde cada modelo corrige los errores del anterior, buscando encontrar la solución más precisa, en este caso, la predicción que más se acerque a lo que deseamos.

Los parámetros que utilicé primeramente ya están algo tuneados, ya que ya estuve experimentando con ellos y ya los había utilizado en previos ejercicios.

Parámetros iniciales:

- n_estimators=200
- max_depth=12
- learning_rate=0.1
- objective='multi:softprob'

Posteriormente obtuvimos los siguientes resultados:

Misclassified samples: 2768

	precision	recall	f1-score	support
0	0.54	0.47	0.50	970
1	0.73	0.74	0.73	341
2	0.43	0.40	0.41	621
3	0.24	0.11	0.15	98
4	0.45	0.32	0.37	341
5	0.40	0.35	0.37	396
6	0.34	0.30	0.32	384
7	0.39	0.50	0.44	1192

8	0.96	0.98	0.97	599
9	0.91	0.94	0.93	619
10	0.95	0.93	0.94	598
11	0.89	0.91	0.90	568
12	0.88	0.87	0.88	590
13	0.84	0.89	0.86	562
14	0.87	0.87	0.87	582

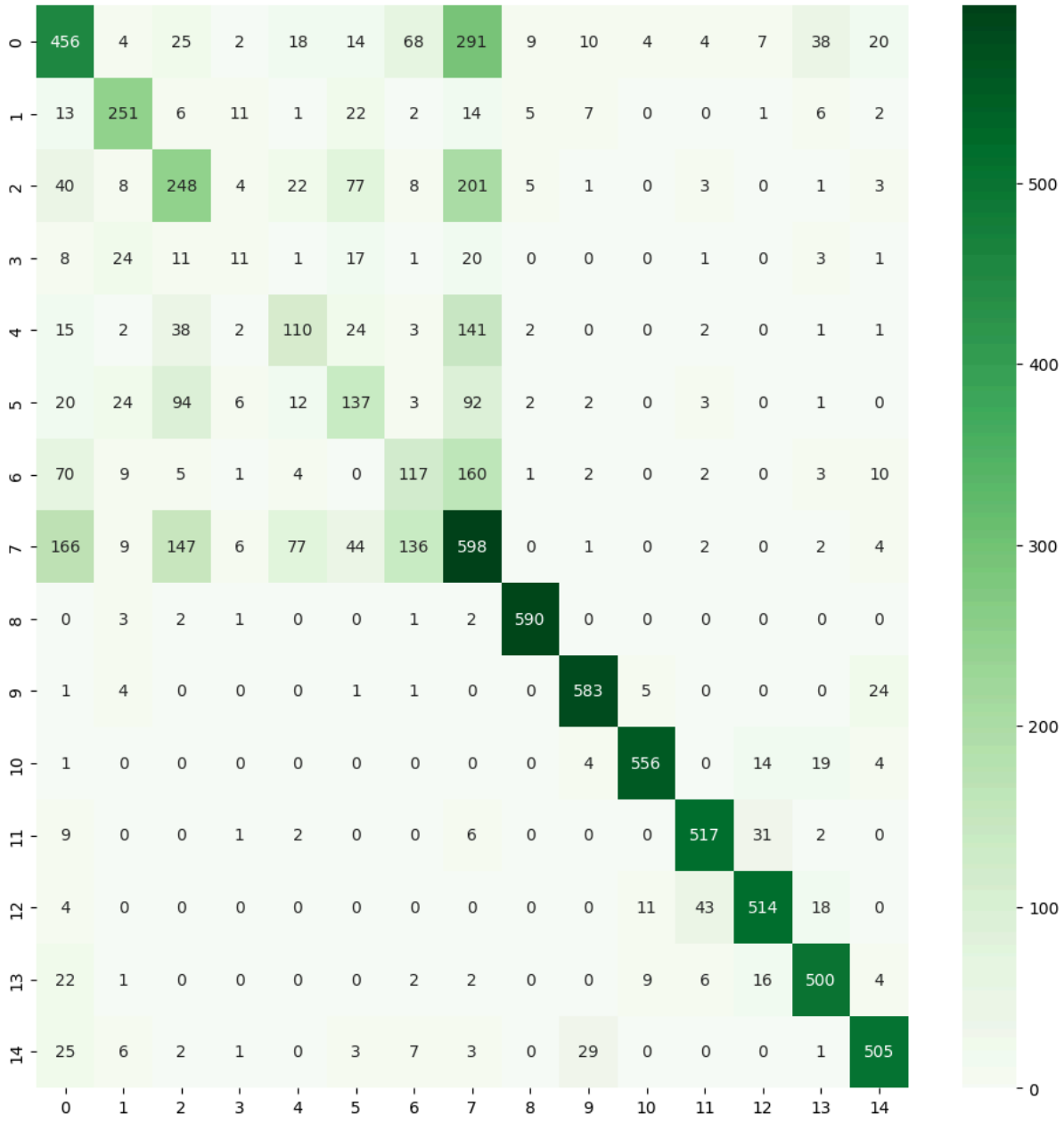
y obtuvimos una accuracy de: 0.6728519087578301

Después de esto para poder analizar más claramente utilicé cross validation, para poder analizar las habilidades de mi modelo y ver si había alguna mejora.

Cross-Validation Scores: [0.67501773 0.66154573 0.67099976 0.66320019 0.66769085 0.66099291
 0.65886525 0.66879433 0.65271868 0.67281324]

Mean Score: 0.6652638663294058
Standard Deviation Score: 0.00661841529228662

Finalmente hicimos una gráfica para ver la comparación:



Decision Tree

Es un algoritmo que utilizaremos para la clasificación de nuestros valores. Su propósito es predecir utilizando un árbol con nodos, donde cada nodo evalúa una característica hasta llegar a una predicción final.

Este algoritmo no lo había empleado anteriormente por lo que puse algunos valores de previos ejercicios y los ejecute para el proyecto.

Parámetros iniciales:

- max_depth = 14 máxima profundidad de búsqueda
- random_state=42

Posteriormente obtuvimos los siguientes resultados:

Misclassified samples: 3167

	precision	recall	f1-score	support
0	0.46	0.40	0.43	970
1	0.58	0.56	0.57	341
2	0.39	0.39	0.39	621
3	0.10	0.08	0.09	98
4	0.58	0.30	0.39	341
5	0.34	0.28	0.30	396
6	0.33	0.22	0.26	384
7	0.41	0.60	0.48	1192
8	0.96	0.94	0.95	599
9	0.84	0.87	0.85	619
10	0.92	0.87	0.90	598
11	0.84	0.84	0.84	568
12	0.81	0.80	0.81	590
13	0.77	0.80	0.79	562
14	0.80	0.76	0.78	582
accuracy			0.63	8461
macro avg	0.61	0.58	0.59	8461
weighted avg	0.63	0.63	0.62	8461

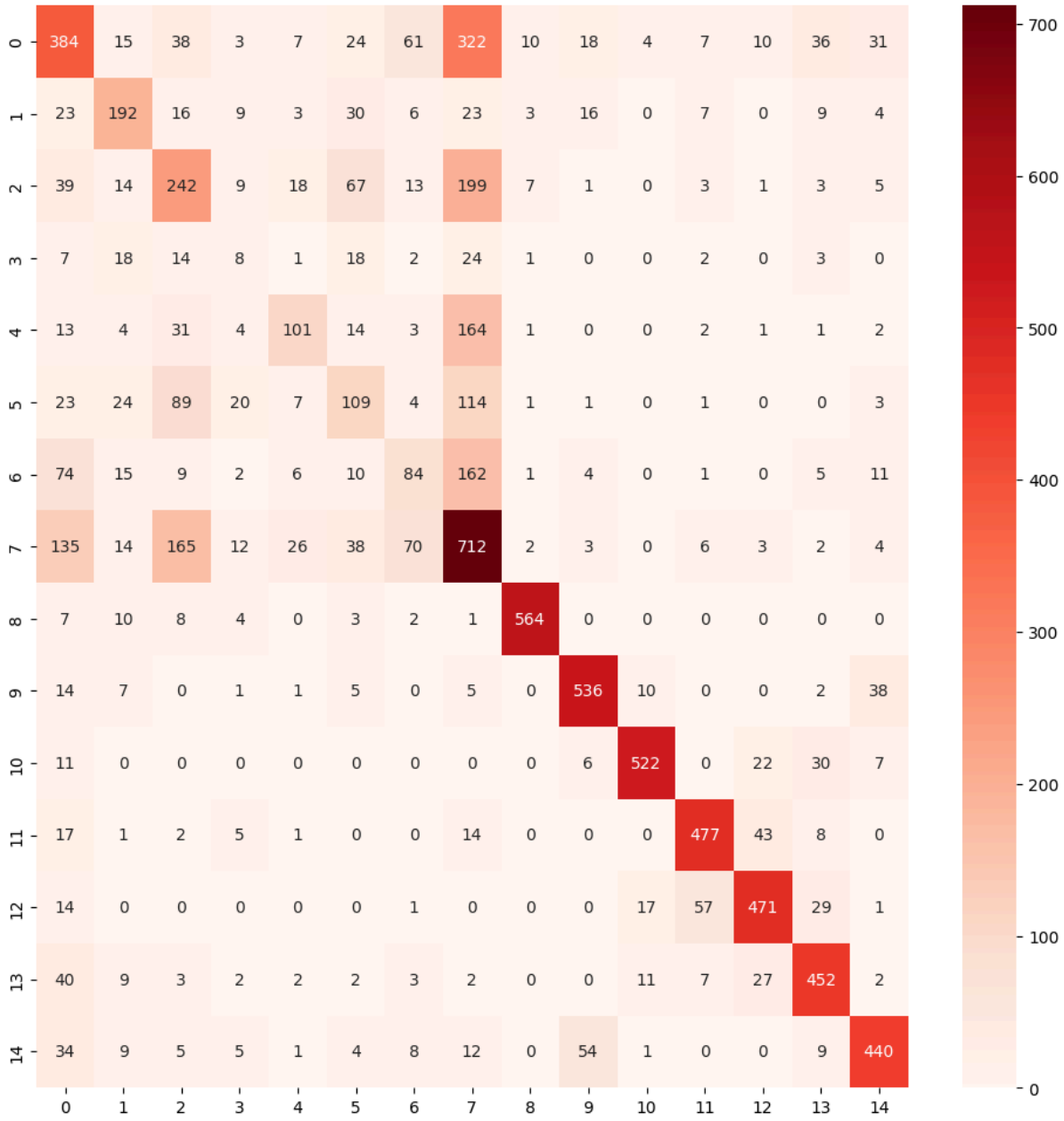
Tree Classifier Score: 0.6256943623685144

Después de esto para poder analizar más claramente utilicé cross validation, para poder analizar las habilidades de mi modelo y ver si había alguna mejora.

Cross-Validation Scores: [0.63932876 0.62372961 0.62798393 0.63034744 0.63223824 0.62174941
 0.63120567 0.62624113 0.6144208 0.62269504]

Mean Score: 0.6269940040665738
Standard Deviation Score: 0.006518491119475002

Finalmente hicimos una gráfica para ver la comparación:



Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto

XGboost

El diagnóstico nos mostró que el bias que tiene nuestro modelo es bajo.

Bias: 1.1286865557862478

Decision Tree

El diagnóstico nos mostró que el bias que tiene nuestro modelo es relativamente alto.

Bias: 2.4622489510963903

Diagnóstico y explicación el grado de varianza: bajo medio alto

XGboost

El diagnóstico nos mostró que la varianza que tiene nuestro modelo es relativamente alta.

Varianza: 3.070903055097839

Decision Tree

El diagnóstico nos mostró que la varianza que tiene nuestro modelo es alta.

Varianza: 3.5115424175282945

Diagnóstico y explicación el nivel de ajuste del modelo: underfitt fitt overfitt

XGboost

Este modelo muestra que tenemos un caso de **overfitting**, ya que utilizamos una gran cantidad y puede que eso haya cambiado el tamaño del ruido que tenemos en el modelo.

Decision Tree

Este modelo muestra que tenemos un caso de **underfitting**, ya que solo utilizamos un árbol de búsqueda, lo que era bastante razonable.

GridSearch

Para este modelo use en los dos modelos Grid Search lo que resultó con los valores que se pueden ver el código del proyecto.

Bibliografía:

- Interview Query. (n.d.). Regression datasets and projects. Retrieved September 7, 2024, from https://www.interviewquery.com/p/regression-datasets-and-projects
- Kaggle. (n.d.). Dataset of songs in Spotify. Retrieved September 7, 2024, from https://www.kaggle.com/datasets/
- Wohlwend, B. (2020, October 15). Decision tree, random forest, and XGBoost: An exploration into the heart of machine learning. Medium. Retrieved September 7, 2024, from https://medium.com/@brandonwohlwend/decision-tree-random-forest-and-xgboost-an-exploration-into-the-heart-of-machine-learning-78c29c8d7b3b
- XGBoost Development Team. (2018). XGBoost parameters — xgboost 0.90 documentation. Retrieved September 7, 2024, from https://xgboost.readthedocs.io/en/latest/parameter.html