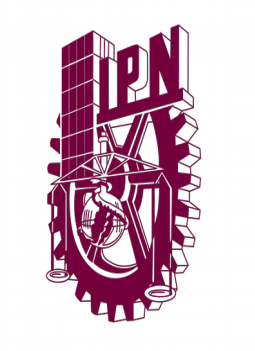


INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

DATA MINING

M. en C. Erika Hernández Rubio

Proyecto:
Árboles de decisión con machine learning en Jupyter y python

Lucio Silva Francisco Javier
Hernández González Luis Armando
Hernández Hernández Alejandro

3cm6

PLANTEAMIENTO DEL PROBLEMA

Trabajar con cantidades grandes de datos para extraer información es difícil, como lo es para clasificar dicha información o como para predecir.

Para eficientar esto, se pueden generar árboles de decisión, los cuales nos permiten, desde clasificar la información, como para predecir algún tipo de información que se agregue, poder clasificarla por su tipo, o conocer algún índice de probabilidad de algún evento.

Pero se enfrenta otro problema, ¿Cómo programarlo para eficientar aún más es trabajo?

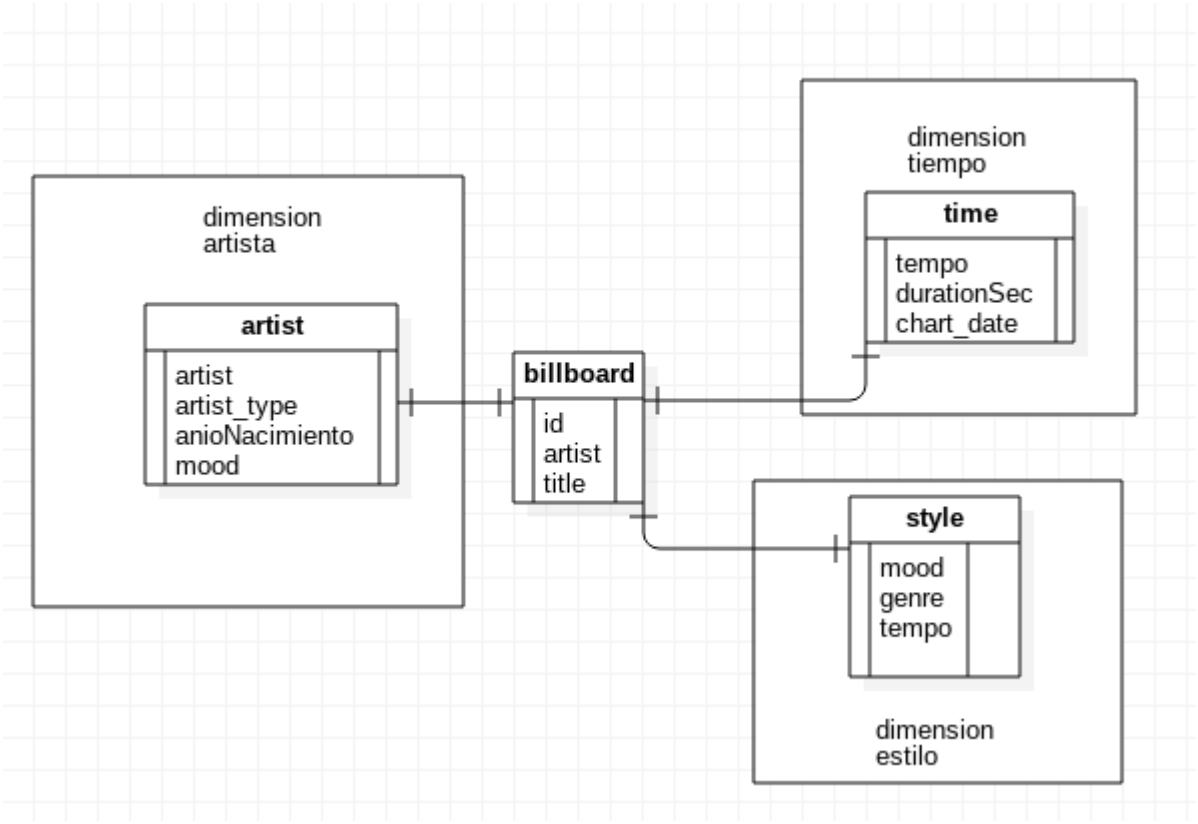
La solución: programar un algoritmo usando machine learning.

Esto es, generar un algoritmo que aprenda como implementar reglas a partir de los datos obtenidos, para así poder clasificar la información y realizar predicciones.

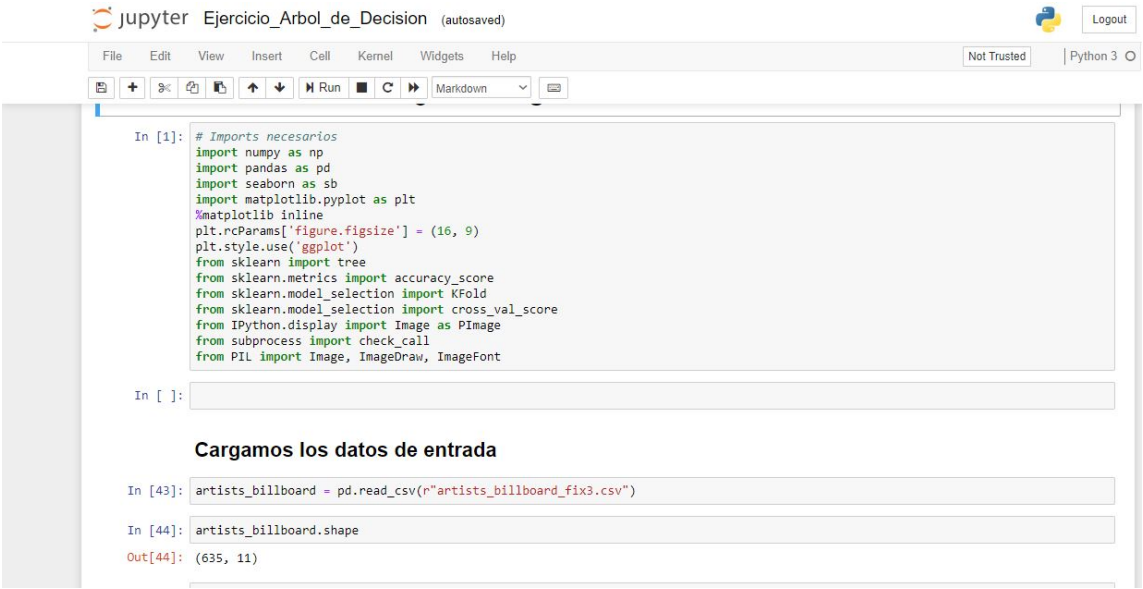
En este proyecto, se desarrollo el árbol de decisión con machine learning en python con Jupyter como front de python, usando como ejemplo un archivo csv con datos de artistas de la lista Billboard, para conocer los posibles primeros lugares.

DATAMART

Para el caso especifico del csv de billboard, quedaria:



capturas del algoritmo:



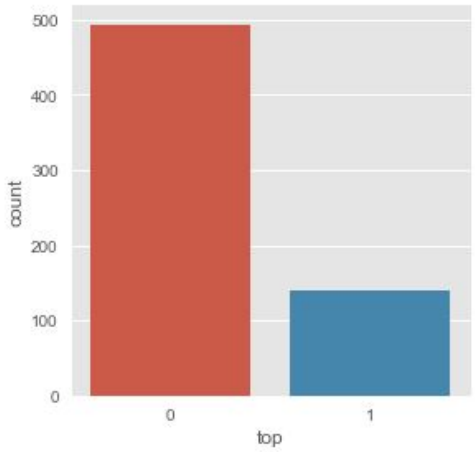
¿Cuántos alcanzaron el número 1?

In [46]:
```
artists_billboard.groupby('top').size()
```

Out[46]:
```
top
0    494
1    141
dtype: int64
```

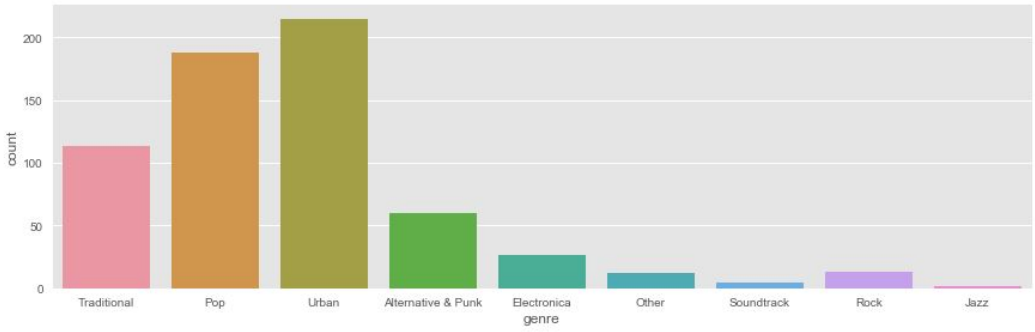
In [47]:
```
sb.factorplot('top',data=artists_billboard,kind="count")
```

Out[47]:
```
<seaborn.axisgrid.FacetGrid at 0x117c23490>
```



In [52]:
```
sb.factorplot('genre',data=artists_billboard,kind="count", aspect=3)
```

Out[52]:
```
<seaborn.axisgrid.FacetGrid at 0x1198d4e90>
```



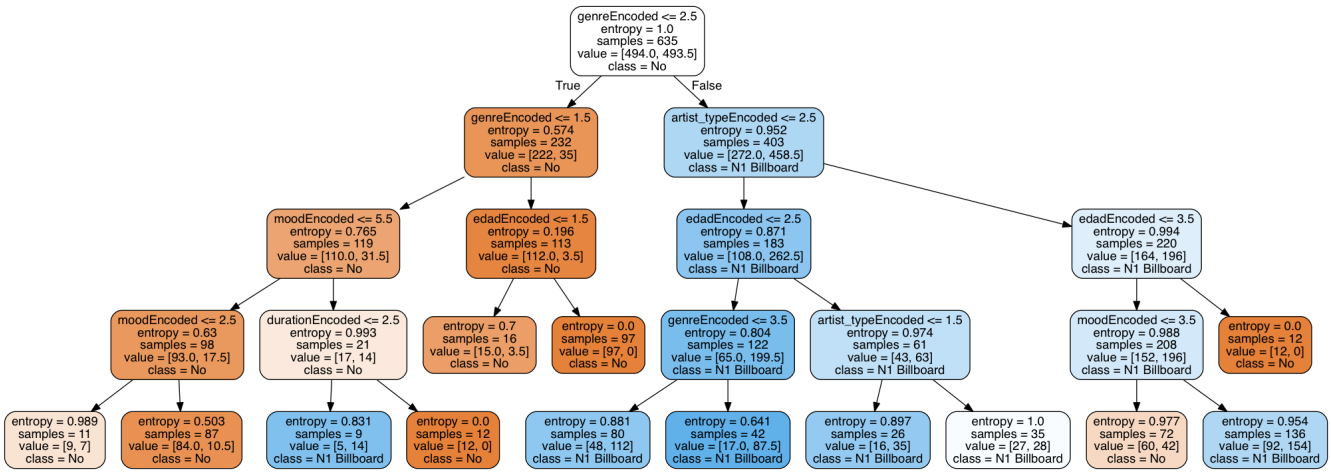
Predicción del árbol de decisión

In [82]:
```
#predecir artista CAMILA CABELLO featuring YOUNG THUG
# con su canción Havana llego a numero 1 Billboard US en 2017

x_test = pd.DataFrame(columns=('top','moodEncoded', 'tempoEncoded', 'genreEncoded','artist_typeEncoded','edadEncoded','durationEn
x_test.loc[0] = (1,5,2,4,1,0,3)
y_pred = decision_tree.predict(x_test.drop(['top'], axis = 1))
print("Prediccion: " + str(y_pred))
y_proba = decision_tree.predict_proba(x_test.drop(['top'], axis = 1))
print("Probabilidad de Acierto: " + str(round(y_proba[0][y_pred][0]* 100, 2))+"%")
```

```
Prediccion: [1]
Probabilidad de Acierto: 83.73%
```

In [83]:
```
#predecir artista Imagine Dragons
# con su canción Believer llego al puesto 42 Billboard US en 2017

x_test = pd.DataFrame(columns=('top','moodEncoded', 'tempoEncoded', 'genreEncoded','artist_typeEncoded','edadEncoded','durationEn
x_test.loc[0] = (0,4,2,1,3,2,3)
y_pred = decision_tree.predict(x_test.drop(['top'], axis = 1))
print("Prediccion: " + str(y_pred))
y_proba = decision_tree.predict_proba(x_test.drop(['top'], axis = 1))
print("Probabilidad de Acierto: " + str(round(y_proba[0][y_pred][0]* 100, 2))+"%")
```

```
Prediccion: [0]
Probabilidad de Acierto: 88.89%
```



CONCLUSIONES

Al final, pudimos realizar una predicción sobre algunos artistas, para saber si podrian ser el primer lugar en el top Billboard, con lo cuál también se pudo observar que la aplicación de machine learning no es 100% confiable, podría decirse que serviría unicamente como soporte, mas no como decisión final.

Agregando a esto, la generación del árbol de decisiones, puede ser bastante útil para agilizar la clasificación de la información que pueda ser agregada. Esto ya aplicado, puede resultar eficiente para diagnosticar padecimientos en el área medica, como son la diabetes, enfermedades cardiacas, enfermedades mentales, y de ayuda en el área pediátrica en temas como déficit de atención.