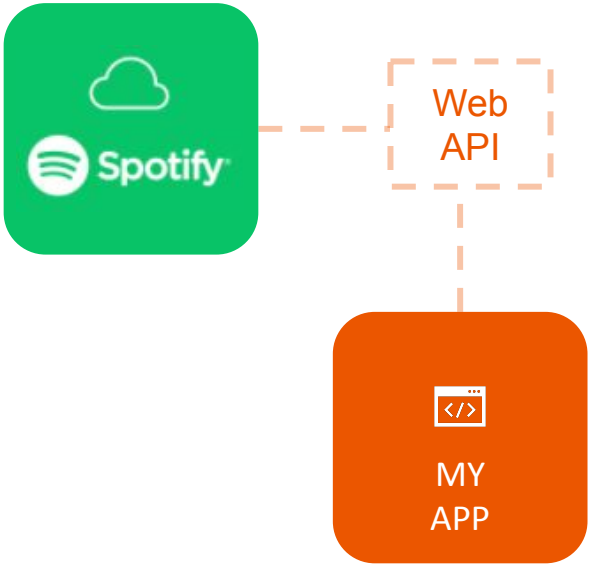
Clasificador de Género Musical

Clasificador de Género Musical

1. Obtención de datos
2. Limpieza y preparación
3. Modelado
4. Resultado

Obtención de Datos

902 Artistas
8329 Albums
132.000 Canciones
360 Minutos de descarga



Limpieza y preparación

Variables Predictivas

- Danceability
- Energy
- Speechiness
- Acousticness
- Instrumentalness
- Popularidad
- Duration_ms
- Valence
- Liveness
- Tempo
- Loudness
- Key
- Time_Signature
- Mode
- Decada

Balanceo de clases

Data set inicial

- Clasica 20%
- Funk 12%
- Rock 11%
- Metal 10%
- Jazz 10%
- Cumbia 10%
- Blues 10%
- Punk 9%
- Pop 8%

Data set Balanceado

- Clasica 11%
- Funk 11%
- Rock 11%
- Metal 11%
- Jazz 11%
- Cumbia 11%
- Blues 11%
- Punk 11%
- Pop 11%

Modelado

Pipeline

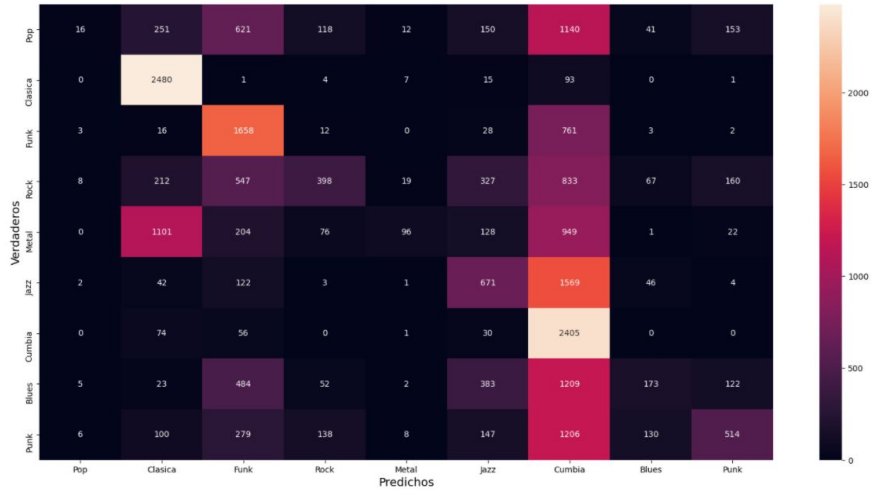
```
numeric_transformer = Pipeline(steps=[('scaler', StandardScaler())])
categorical_transformer = Pipeline(steps=[('onehot', OneHotEncoder(handle_unknown='ignore'))])

engineer = ColumnTransformer(transformers=[('num', numeric_transformer, numerical),
                                           ('cat', categorical_transformer, dummies)])

folds=StratifiedKFold(n_splits=5,shuffle=True, random_state=42)
```

Modelado

Baseline: GaussianNB

```
pipeline_gnb = Pipeline(steps=[('feature_engineering', engineer),  ('gauss', GaussianNB())])
pipeline_gnb.fit(X_train,y_train)
y_pred = pipeline_gnb.predict(X_test)
accuracy(y_test,y_pred)
```

37% Accuracy score



Modelado

Modelo final: KNN

```
pasos1 = [('feature engineering', engineer), ("knn", KNeighborsClassifier())]
pipe_grid_1 = Pipeline(pasos1)
param_grid_1 = {"knn__n_neighbors": range(1,40,1), "knn__weights": ["uniform", "distance"]}
grid1 = GridSearchCV(pipe_grid_1, param_grid_1, cv=folds, verbose=10)
grid1.fit(X_train, y_train)
```

•••

```
print(grid1.best_estimator_)
print(grid1.best_score_)
```

```
Pipeline(steps=[('feature engineering',
                 ColumnTransformer(transformers=[('num',
                                                  Pipeline(steps=[('scaler',
                                                                   StandardScaler())]),
                                                  ['danceability', 'energy',
                                                   'loudness', 'speechiness',
                                                   'acousticness',
                                                   'instrumentalness',
                                                   'liveness', 'valence',
                                                   'tempo', 'duration_ms',
                                                   'popularidad']),
                                                 ('cat',
                                                  Pipeline(steps=[('onehot',
                                                                   OneHotEncoder(handle_unknown='ignore'))]),
                                                  ['key', 'time_signature',
                                                   'mode', 'decada'])])),
                ('knn',
                 KNeighborsClassifier(n_neighbors=16, weights='distance'))])
0.7061504816127935
```

Resultado

71% Accuracy score