Harmonizing Hit Predictions: Leveraging Audio Features to Forecast Spotify's Top Songs

Link to the colab:
https://colab.research.google.com/drive/17shSZtfZwrFMzsSofDNQWH9G_cW_xOcv?usp=sharing

Project by: Team 05

- Dylan Kakkanad
- Priyanka Chaudhari
- Saachi Dholakia
- Sahasra Konkala

Refresh

1. Problem Statement

In the dynamic realm of the music industry, accurately predicting a song's fate on the Spotify Global Top 50 chart is a pivotal endeavor for artists, record companies, and music enthusiasts. This project is dedicated to the development of a sophisticated classification model, leveraging a dataset meticulously sourced from data.world and enriched through the Spotify API. The primary objective is to equip stakeholders with a predictive tool that assigns a song to specific popularity classes, enabling a nuanced understanding of its potential success and aiding strategic decision-making.

As we navigate the intricacies of the dataset, challenges such as the skewed distribution of popularity scores have come to the forefront. Addressing this, our approach involves implementing a thoughtful stratified sampling strategy and supplementing the dataset with additional information gleaned from the Spotify API. Beyond the technicalities, our investigation seeks to unravel the essence of song popularity by scrutinizing the interplay between various audio features and the likelihood of a song being deemed 'popular.' The ultimate aim is to distill insights that contribute not only to revenue and success but also align with the evolving trend of data-driven decision-making in the music industry. This classification model, designed to delineate songs into distinct popularity categories, holds the promise of offering a comprehensive view of a song's potential impact, thereby enriching the experiences of music enthusiasts and providing a strategic edge for industry stakeholders.

2. Executive Summary

Building on our comprehensive approach to predicting a song's position on the Spotify Global Top 50 chart, we employ a diverse set of machine learning models, including Random Forest Classifier (RFC), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Logistic Regression. This multi-model strategy ensures a thorough exploration of the complex relationships within the dataset. Following model fitting, we fine-tune hyperparameters to enhance predictive performance. Notably, we prioritize evaluating balanced accuracy to account for the imbalanced nature of the

popularity distribution, as identified in our data exploration. This meticulous process aims to provide a robust predictive framework that goes beyond individual model biases, offering stakeholders a reliable tool for gauging a song's likelihood of success on the global music charts.

3. Importance of the Problem

The significance of predicting a song's placement on the Spotify Global Top 50 chart extends far beyond the realm of musical preference; it is a pivotal challenge with profound implications for artists, record companies, and the entire music industry. The ability to forecast a song's trajectory not only holds the key to heightened revenue and success but also empowers stakeholders to make informed, data-driven decisions. In an industry increasingly reliant on analytics, predictive models offer a competitive advantage by guiding strategic releases, optimizing promotional efforts, and enhancing user experiences on streaming platforms. This undertaking aligns with the industry's shift toward data-driven decision-making, promising to revolutionize how artists, record labels, and music enthusiasts navigate the complex landscape of musical trends and popularity.

4. Data Description

Variable Name	Data Type	Description	Variable Type	Dependent/Independent v
track_id	Object	Unique ID for each Track	Categorical	Independent variable
artists	Object	Name of artist/s	Categorical	Independent variable
album_name	Object	Name of album	Categorical	Independent variable
track_name	Object	Song name	Categorical	Independent variable
duration_ms	int64	Track duration (milliseconds)	Numeric	Independent variable
popularity	int64	Spotify metric (range 0-100)	Numeric	`Dependent variable`
explicit	Boolean	Is track explicit (bool 0 or 1)	Categorical	Independent variable
danceability	float64	How suitable a track is for dancing (range 0-1)	Numeric	Independent variable
energy	float64	Perceptual measure of intensity and activity (range 0-1)	Numeric	Independent variable
key	int64	The key the track is in (range -1 : 11)	Numeric	Independent variable
loudness	float64	Loudness of a track in decibels (range -60 : 0)	Numeric	Independent variable
mode	int64	The modality (major or minor) of a track (0 or 1)	Categorical	Independent variable
speechiness	float64	Presence of spoken words in a track (range 0-1)	Numeric	Independent variable
acousticness	float64	Whether the track is acoustic (range 0-1)	Numeric	Independent variable
instrumentalness	float64	Whether a track contains no vocals (range 0-1)	Numeric	Independent variable
liveness	float64	Presence of an audience (range 0-1)	Numeric	Independent variable
valence	float64	Describing the musical positiveness (range 0:1)	Numeric	Independent variable
tempo	float64	The speed or pace of a given piece in beats per minute	Numeric	Independent variable

Variable Name	Data Type	Description	Variable Type	Dependent/Independent v
time_signature	int64	The time signature specifying beats in each bar	Numeric	Independent variable
track_genre	Object	Genre of Track	Categorical	Independent variable

- The number of rows is 92233
- The number of columns is 20

5. Importing Libraries

5.1 Libraries

```
!pip3 install spotipy
```

```
    Collecting spotipy
      Downloading spotipy-2.23.0-py3-none-any.whl (29 kB)
    Collecting redis>=3.5.3 (from spotipy)
      Downloading redis-5.0.1-py3-none-any.whl (250 kB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 250.3/250.3 kB 6.1 MB/s eta 0:00:0
    Requirement already satisfied: requests>=2.25.0 in /usr/local/lib/python3.10/dis
    Requirement already satisfied: six>=1.15.0 in /usr/local/lib/python3.10/dist-pac
    Requirement already satisfied: urllib3>=1.26.0 in /usr/local/lib/python3.10/dist
    Requirement already satisfied: async-timeout>=4.0.2 in /usr/local/lib/python3.10
    Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-pa
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/d
    Installing collected packages: redis, spotipy
    Successfully installed redis-5.0.1 spotipy-2.23.0
```

```
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials
import urllib
import re
import os
import pandas as pd
import numpy as np
import sys
from datetime import datetime
import seaborn as sns
import matplotlib.pyplot as plt
import time
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, balanced_accuracy_score, f1_score, prec
import sklearn.metrics as metrics
```

5.2 Mounting Google Drive

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call d
```

6. Understanding the Spotify Dataset:

```
spotify_df = pd.read_csv('/content/gdrive/MyDrive/BA_810/810 Project/spotify_datase
spotify_df
```

```
                        track_id       artists  album_name  track_name  popularity
```

```
spotify_df = spotify_df.drop_duplicates(subset=['track_id'])
```

```
                                                                  Ghost        Ghost -
```

```
spotify_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 89741 entries, 0 to 113999
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   track_id          89741 non-null  object 
 1   artists           89740 non-null  object 
 2   album_name        89740 non-null  object 
 3   track_name        89740 non-null  object 
 4   popularity        89741 non-null  int64  
 5   duration_ms       89741 non-null  int64  
 6   explicit          89741 non-null  bool   
 7   danceability      89741 non-null  float64
 8   energy            89741 non-null  float64
 9   key               89741 non-null  int64  
 10  loudness          89741 non-null  float64
 11  mode              89741 non-null  int64  
 12  speechiness       89741 non-null  float64
 13  acousticness      89741 non-null  float64
 14  instrumentalness  89741 non-null  float64
 15  liveness          89741 non-null  float64
 16  valence           89741 non-null  float64
 17  tempo             89741 non-null  float64
 18  time_signature    89741 non-null  int64  
 19  track_genre       89741 non-null  object 
dtypes: bool(1), float64(9), int64(5), object(5)
memory usage: 13.8+ MB
```

```
113999  2hETkH7cOfqmz3LqZDHZf5  Cesária Evora  ...  Barbincor  2
```

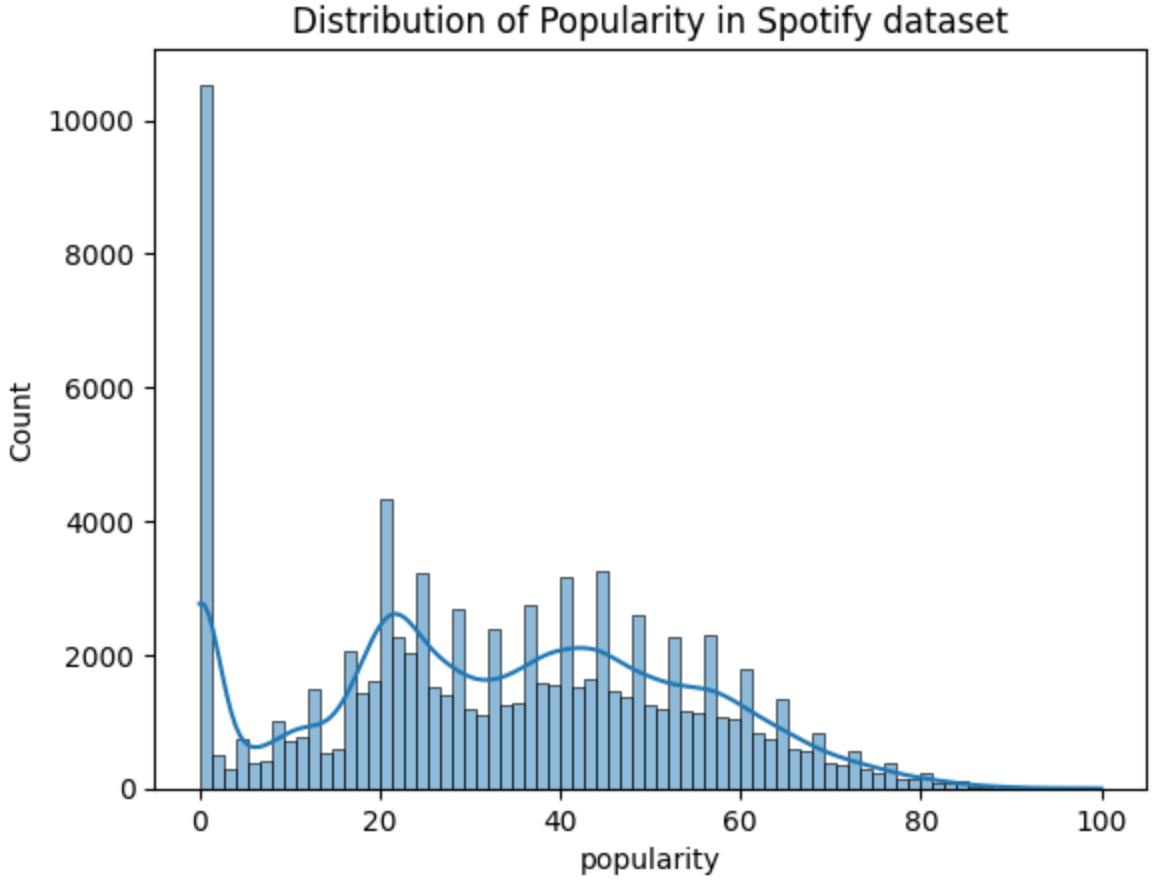
```
spotify_df.dropna(inplace=True)
spotify_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 89740 entries, 0 to 113999
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   track_id          89740 non-null  object 
 1   artists           89740 non-null  object 
 2   album_name        89740 non-null  object 
 3   track_name        89740 non-null  object 
 4   popularity        89740 non-null  int64  
 5   duration_ms       89740 non-null  int64  
 6   explicit          89740 non-null  bool   
 7   danceability      89740 non-null  float64
 8   energy            89740 non-null  float64
 9   key               89740 non-null  int64  
 10  loudness          89740 non-null  float64
```

```
 11  mode              89740 non-null  int64
 12  speechiness       89740 non-null  float64
 13  acousticness      89740 non-null  float64
 14  instrumentalness  89740 non-null  float64
 15  liveness          89740 non-null  float64
 16  valence           89740 non-null  float64
 17  tempo             89740 non-null  float64
 18  time_signature    89740 non-null  int64
 19  track_genre       89740 non-null  object
dtypes: bool(1), float64(9), int64(5), object(5)
memory usage: 13.8+ MB
<ipython-input-28-d2374c05b066>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  spotify_df.dropna(inplace=True)
```

```
sns.histplot(spotify_df.popularity,kde=True)
plt.title('Distribution of Popularity in Spotify dataset')
```

```
Text(0.5, 1.0, 'Distribution of Popularity in Spotify dataset')
```



We can observe the right-skewed data which is why to over-come the skewness we would now pull data from Spotify using Spotify api.

7. Data collection through Spotify API:

```
# Set up the client credentials manager
client_id =  '63ae4cb856e04da3a12d0b1be61329c9' #'9a40b4f3712a407ba31e9e9f4d5508ce'
client_secret =  '8b2febadbdcf45e7989943fb5cc76fa2' #'eb2e29cf811a4eb4a86d4037696f3
client_credentials_manager = SpotifyClientCredentials(client_id=client_id, client_s
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)



# Get the playlist ID
playlist_id = '37i9dQZEVXcNuIGtTaIMH8' #'37i9dQZF1DXe2bobNYDtW8' #'37i9dQZF1DWUUeLCl

# Get the playlist details
playlist = sp.playlist(playlist_id)

# Get the number of songs in the playlist
num_songs = playlist['tracks']['total']
```

Retrieving Details of Each Song in the Playlist

```
# Get the details of each song in the playlist
results = []
now = datetime.now()

offset = 0
while offset < num_songs:
    tracks = sp.playlist_tracks(playlist_id, offset=offset)
    results += tracks['items']
    offset += len(tracks['items'])
```

Function to Clean Track Names

```
def clean_names(name):
    name = name.lower()
    # remove non-alphanumeric characters
    name = re.sub(r'[^a-zA-Z0-9\s]', '', name)
    # remove spaces and replace with underscore
    name = re.sub(r'\s+', '_', name)

    return name
print(results)
```

```
[{'added_at': '2023-12-04T05:00:00Z', 'added_by': {'external_urls': {'spotify':
```

Extract Track Information

```
track_names = [results[i]["track"]["name"] for i in range(len(results))]
track_add_date = [results[i]["added_at"] for i in range(len(results))]
multiple_artists_bool = [len(results[i]["track"]["artists"]) > 1 for i in range(len
name_of_artists = [[results[i]["track"]["artists"][j]["name"] for j in range(len(res
album_name = [results[i]["track"]["album"]["name"] for i in range(len(results))]
album_release_date = [results[i]["track"]["album"]["release_date"] for i in range(le
album_release_date_precision = [results[i]["track"]["album"]["release_date_precisio
number_of_tracks_in_album = [results[i]["track"]["album"]["total_tracks"] for i in
position_in_playlist = [i+1 for i in range(len(results))]
track_duration_ms = [results[i]["track"]["duration_ms"] for i in range(len(results)
id = [results[i]["track"]["id"] for i in range(len(results))]
track_popularity = [results[i]["track"]["popularity"] for i in range(len(results))]
track_explicit = [results[i]["track"]["explicit"] for i in range(len(results))]
date_scrapping = now.strftime("%Y-%m-%d")
data_collection_date = [date_scrapping for i in range(len(position_in_playlist))]
```

To check if we have missing data

```
if len(id) == len(data_collection_date) == len(track_names) == len(track_add_date)
    print("All arrays have the same length.")
else:
    print("Arrays do not have the same length.")
```

```
All arrays have the same length.
```

Creating a DataFrame for new songs

```
dataset = pd.DataFrame({'track_id':id,"data_collection_date":data_collection_date,"
```

```
dataset["name_of_artists"] = dataset["name_of_artists"].astype(str)
dataset
```

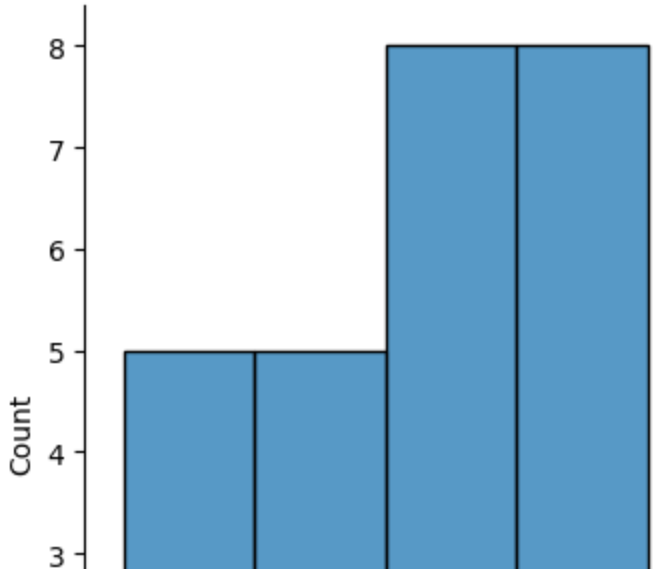
	track_id	data_collection_date	track_name	track_add_date	t
0	3JWneiKPrQOEJQ8eXJWDq9	2023-12-06	Poor Madeline	2023-12-04T05:00:00Z	
1	2Nd1dImwW0VVN5HJ9MfvUd	2023-12-06	Name Something Better	2023-12-04T05:00:00Z	
2	53Pu1MH8VqR8mkPUkPJEbw	2023-12-06	I Have Considered the Lilies	2023-12-04T05:00:00Z	
3	0T0fkqReu2aO2H0VPUPWTo	2023-12-06	Baby I'm Yours - Digitally Remastered	2023-12-04T05:00:00Z	
4	5H3tZqdgoP10JunTpcUFvL	2023-12-06	God Help The Girl	2023-12-04T05:00:00Z	
5	5XOmmOQZFpzeHnWcsFAEiX	2023-12-06	Blue	2023-12-04T05:00:00Z	
6	58emzxS7apqOn2amBFfDyr	2023-12-06	No One Has the Answers	2023-12-04T05:00:00Z	
7	7A3uhvZBmoLzCCFQT14lGU	2023-12-06	The Giver	2023-12-04T05:00:00Z	
8	4mL3gs1HONGGLaZyW6OYMq	2023-12-06	Close One	2023-12-04T05:00:00Z	
9	3nrTr8RJU8yYbJo57xXnh0	2023-12-06	Come Monday Night	2023-12-04T05:00:00Z	
10	2h9fzaFksNxT7WaR5RqwXs	2023-12-06	Blur	2023-12-04T05:00:00Z	
11	5qgyfyptP3pLeWs9maKPkg	2023-12-06	Rambling Man	2023-12-04T05:00:00Z	
12	71eUN154oDoZ2OSvS4GaL7	2023-12-06	Fucking Married	2023-12-04T05:00:00Z	
13	3bW3FVztCl5BoY8QooHHzj	2023-12-06	Little Queenie (2023 Remastered Version)	2023-12-04T05:00:00Z	
14	19jTCAxeVnkwKnWHSsdg0i	2023-12-06	Lucky for You	2023-12-04T05:00:00Z	
15	1jJvNlkbQmtRpG9uIUpiYA	2023-12-06	Teal	2023-12-04T05:00:00Z	

16	2tUzZiFwDCpIx6Hfa8ofoW	2023-12-06	The Magdalene Laundries	2023-12-04T05:00:00Z
17	6lXTGkDn7tTNR38b7Spqgs	2023-12-06	Ignore the Days	2023-12-04T05:00:00Z
18	6andWFPbmPzbgdKcEamIq9	2023-12-06	Your Mother's Name	2023-12-04T05:00:00Z
19	5wIYMYa6syn9xxL3i0mVx9	2023-12-06	Superstar	2023-12-04T05:00:00Z
20	0InIoa05NmkSGlRD6wiQBN	2023-12-06	California - Live	2023-12-04T05:00:00Z
21	6ONTW7En776d3rHoJ2oynA	2023-12-06	Want Me	2023-12-04T05:00:00Z
22	3j2kzar3MebQqiwLE2XOkO	2023-12-06	So Sleepy	2023-12-04T05:00:00Z
23	69inNA6QmJ1uWzBIAxyC26	2023-12-06	Low	2023-12-04T05:00:00Z
24	7kofQ4W8kA6nOzyXdPz7ny	2023-12-06	mirrored heart	2023-12-04T05:00:00Z
25	7EIDrlo32lf1vE7pmIdACo	2023-12-06	I Love You	2023-12-

8. Exploring the playlist:

8.1 Checking the popularity tracks distribution in the playlist #13:

```
sns.displot(dataset.track_popularity,kind='hist')
dataset=dataset[dataset.track_popularity>40]
```



The above graph shows the distribution of a particular playlist we have pulled. Since

dataset

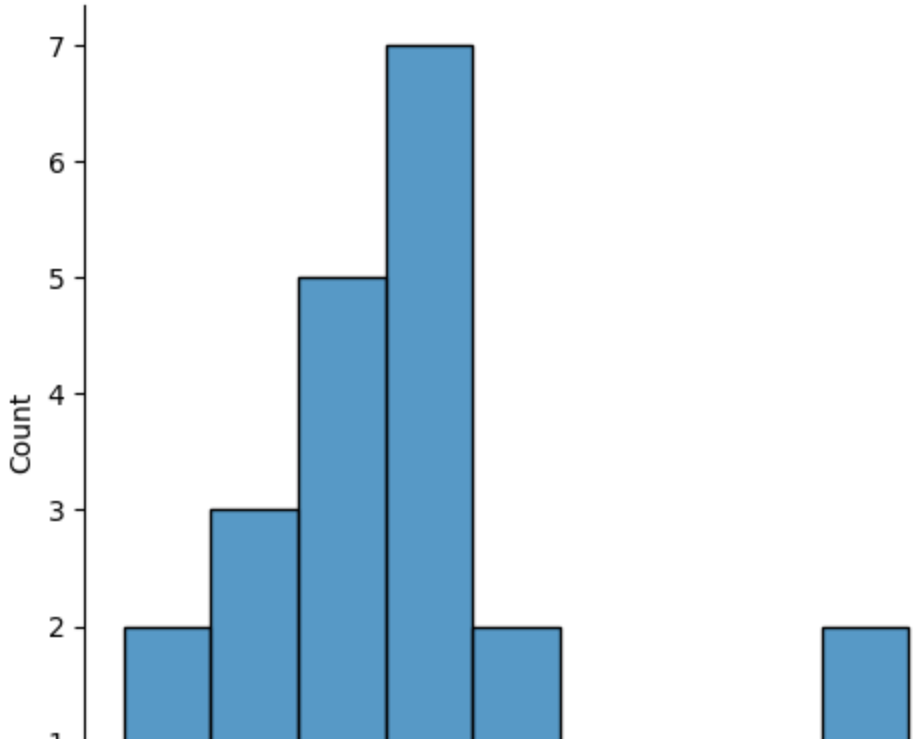
	track_id	data_collection_date	track_name	track_add_date	t
0	3JWneiKPrQOEJQ8eXJWDq9	2023-12-06	Poor Madeline	2023-12-04T05:00:00Z	
1	2Nd1dImwW0VVN5HJ9MfvUd	2023-12-06	Name Something Better	2023-12-04T05:00:00Z	
3	0T0fkqReu2aO2H0VPUPWTo	2023-12-06	Baby I'm Yours - Digitally Remastered	2023-12-04T05:00:00Z	
5	5XOmmOQZFpzeHnWcsFAEiX	2023-12-06	Blue	2023-12-04T05:00:00Z	
7	7A3uhvZBmoLzCCFQT14lGU	2023-12-06	The Giver	2023-12-04T05:00:00Z	
8	4mL3gs1HONGGLaZyW6OYMq	2023-12-06	Close One	2023-12-04T05:00:00Z	
10	2h9fzaFksNxT7WaR5RqwXs	2023-12-06	Blur	2023-12-04T05:00:00Z	
11	5qgyfyptP3pLeWs9maKPkg	2023-12-06	Rambling Man	2023-12-04T05:00:00Z	
12	71eUN154oDoZ2OSvS4GaL7	2023-12-06	Fucking Married	2023-12-04T05:00:00Z	
13	3bW3FVztCl5BoY8QooHHzj	2023-12-06	Little Queenie (2023 Remastered Version)	2023-12-04T05:00:00Z	
14	19jTCAxeVnkwKnWHSsdg0i	2023-12-06	Lucky for You	2023-12-04T05:00:00Z	
15	1jJvNlkbQmtRpG9uIUpiYA	2023-12-06	Teal	2023-12-04T05:00:00Z	

```
dataset.reset_index(inplace=True)
```

Laundries

```
sns.displot(dataset.track_popularity,kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x7a84dcf87fd0>
```




```
dataset.corr()
```

```
<ipython-input-58-c187c74d1e71>:1: FutureWarning: The default value of numeric_c
  dataset.corr()
```

	index	track_add_time	number_of_tracks_in_album
index	1.000000e+00	-9.268110e-17	0.375850
track_add_time	-9.268110e-17	1.000000e+00	-0.038861
number_of_tracks_in_album	3.758498e-01	-3.886082e-02	1.000000
position_in_playlist	1.000000e+00	-9.671071e-17	0.375850
track_duration_ms	-1.006314e-01	1.054560e-01	-0.022086
track_popularity	-1.614121e-01	2.948394e-01	-0.040495
track_explicit	6.763272e-02	-1.490712e-01	0.148233

This is the analysis for playlist no.13. We had done the same for the 12 playlists before this and downloaded and concated each of them.

The concated playlist would be added to Spotify Dataset

```
# dataset.to_csv('temp.csv')


# dataset=pd.read_csv('/content/temp.csv',index_col=0)
# dataset


features = []
for i in range(0, len(dataset['track_id']), 50):
    track_ids = dataset['track_id'][i:i+50]
    track_features = sp.audio_features(tracks=track_ids)
    features.extend(track_features)
    #time.sleep(40) # Add a delay of 1 second

# Create a DataFrame from the features
df = pd.DataFrame(features)
```

The above code performs a second API call to get track-level audio feature data for each track in the playlist.

```
df.head()
```

pe	id	uri	
es	3JWneiKPrQOEJQ8eXJWDq9	spotify:track:3JWneiKPrQOEJQ8eXJWDq9	https://api.spotify.com/
es	2Nd1dImwW0VVN5HJ9MfvUd	spotify:track:2Nd1dImwW0VVN5HJ9MfvUd	https://api.spotify.com/
es	0T0fkqReu2aO2H0VPUPWTo	spotify:track:0T0fkqReu2aO2H0VPUPWTo	https://api.spotify.com
es	5XOmmOQZFpzeHnWcsFAEiX	spotify:track:5XOmmOQZFpzeHnWcsFAEiX	https://api.spotify.com/v
es	7A3uhvZBmoLzCCFQT14lGU	spotify:track:7A3uhvZBmoLzCCFQT14lGU	https://api.spotify.com

9. Merging Spotify Dataset and 13 Playlist:

```
# Merge the features DataFrame with the playlist dataset
dataset = pd.concat([dataset, df], axis=1)


dataset
```

	index	track_id	data_collection_date	track_name	track_add_
0	0	3JWneiKPrQOEJQ8eXJWDq9	2023-12-06	Poor Madeline	202 04T05:0
1	1	2Nd1dImwW0VVN5HJ9MfvUd	2023-12-06	Name Something Better	202 04T05:0
2	3	0T0fkqReu2aO2H0VPUPWTo	2023-12-06	Baby I'm Yours - Digitally Remastered	202 04T05:0
3	5	5XOmmOQZFpzeHnWcsFAEiX	2023-12-06	Blue	202 04T05:0
4	7	7A3uhvZBmoLzCCFQT14lGU	2023-12-06	The Giver	202 04T05:0
5	8	4mL3gs1HONGGLaZyW6OYMq	2023-12-06	Close One	202 04T05:0
6	10	2h9fzaFksNxT7WaR5RqwXs	2023-12-06	Blur	202 04T05:0
7	11	5qgyfyptP3pLeWs9maKPkg	2023-12-06	Rambling Man	202 04T05:0
8	12	71eUN154oDoZ2OSvS4GaL7	2023-12-06	Fucking Married	202 04T05:0
9	13	3bW3FVztCl5BoY8QooHHzj	2023-12-06	Little Queenie (2023 Remastered Version)	202 04T05:0
10	14	19jTCAxeVnkwKnWHSsdg0i	2023-12-06	Lucky for You	202 04T05:0
11	15	1jJvNlkbQmtRpG9uIUpiYA	2023-12-06	Teal	202 04T05:0
12	16	2tUzZiFwDCpIx6Hfa8ofoW	2023-12-06	The Magdalene Laundries	202 04T05:0
13	18	6andWFPbmPzbgdKcEamIq9	2023-12-06	Your Mother's Name	202 04T05:0
14	19	5wIYMYa6syn9xxL3i0mVx9	2023-12-06	Superstar	202 04T05:0
15	21	6ONTW7En776d3rHoJ2oynA	2023-12-06	Want Me	202 04T05:0

16	23	69inNA6QmJ1uWzBIAxyC26	2023-12-06	Low	202 04T05:0
17	24	7kofQ4W8kA6nOzyXdPz7ny	2023-12-06	mirrored heart	202 04T05:0
18	26	1usL7pG5XVHub0QIoJ1Lrt	2023-12-06	Last of the Loving	202 04T05:0
19	27	1SNc2qOJzugq25SfARkAPR	2023-12-06	Blue, Pt. 2	202 04T05:0
20	28	6ydPTlLqw0Y1hWmEqCiRQK	2023-12-06	Control	202 04T05:0
21	29	4fn4imVYZcfM5s4cSCDRTm	2023-12-06	A Well Respected Man	202 04T05:0

22 rows × 33 columns

```
print(dataset.columns)
dataset.rename({'track_popularity':'popularity','name_of_artists':'artists','track_
dataset.columns
```

```
Index(['index', 'track_id', 'data_collection_date', 'track_name',
       'track_add_date', 'track_add_time', 'name_of_artists', 'album_name',
       'album_release_date', 'album_release_date_precision',
       'number_of_tracks_in_album', 'position_in_playlist',
       'track_duration_ms', 'track_popularity', 'track_explicit',
       'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo',
       'type', 'id', 'uri', 'track_href', 'analysis_url', 'duration_ms',
       'time_signature'],
      dtype='object')
Index(['index', 'track_id', 'data_collection_date', 'track_name',
       'track_add_date', 'track_add_time', 'artists', 'album_name',
       'album_release_date', 'album_release_date_precision',
       'number_of_tracks_in_album', 'position_in_playlist',
       'track_duration_ms', 'popularity', 'explicit', 'danceability', 'energy',
       'key', 'loudness', 'mode', 'speechiness', 'acousticness',
       'instrumentalness', 'liveness', 'valence', 'tempo', 'type', 'id', 'uri',
       'track_href', 'analysis_url', 'duration_ms', 'time_signature'],
      dtype='object')
```

9.1 Cleaning the Merged Dataset

```
merge = pd.merge(spotify_df,dataset,how='outer', on=['track_id', 'artists', 'album_
       'duration_ms', 'danceability', 'energy', 'key', 'loudness',
       'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness',
       'valence', 'tempo', 'time_signature','explicit'])
```

```
columns_to_keep = ['track_id', 'artists', 'album_name', 'track_name', 'popularity',
merge = merge.loc[:, columns_to_keep]
```

```
merge = merge.drop_duplicates(subset=['track_id'])
```

```
merge = merge.dropna(subset=['track_id'])
```

```
merge.shape
```

```
sns.displot(merge.popularity,kind='kde')
```

```
merge.shape
```

```
merge.to_csv('new_dataset (13).csv')
```

10. Analysis of Merged Data

```
merge=pd.read_csv('/content/gdrive/MyDrive/BA_810/810 Project/new_dataset (13).csv'
```

```
merge.shape
```

```
    (92233, 20)
```

```
merge.drop_duplicates(inplace=True)
```

```
merge
```

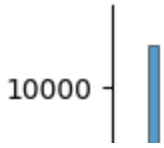
	Unnamed: 0	track_id	artists	album_name	track_name	p
0	0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	
1	1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	
2	2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	
3	3	6lfxq3CG4xtTiEg7opyCyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Sou...	Can't Help Falling In Love	
4	4	5vjLSffimiIP26QG5WcN2K	Chord Overstreet	Hold On	Hold On	
...	...	...	...	...	...	
92228	92509	1y5beX4vn7QPfg36r4jiWo	['Maninder Buttar', 'Zara Khan', 'Tanishk Bagc...	Sakhiyan2.0 (From "BellBottom") - Single	Sakhiyan2.0 (From "BellBottom")	

```
len(merge[merge.popularity>=10])
```

```
78367
```

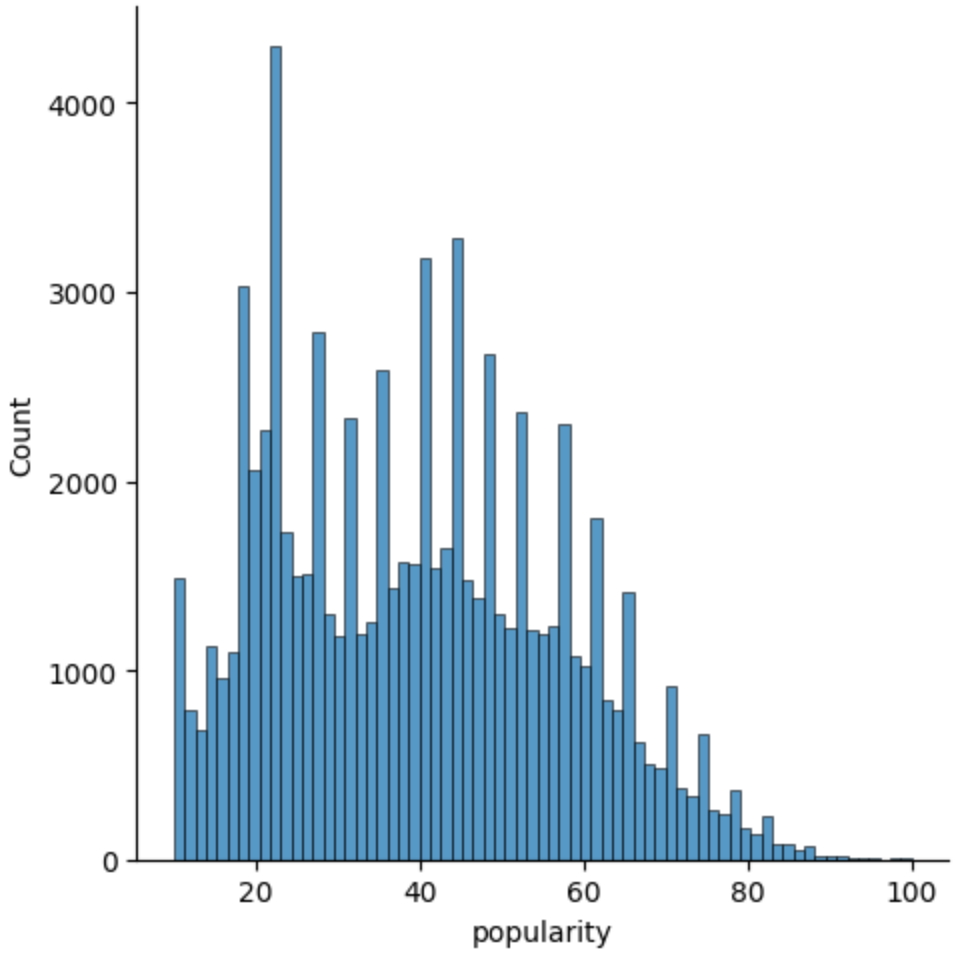
```
sns.displot(merge.popularity,kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x7a84ea0e3520>
```




```
merge = merge[merge.popularity >=10]
sns.displot(merge.popularity,kind='hist')
```

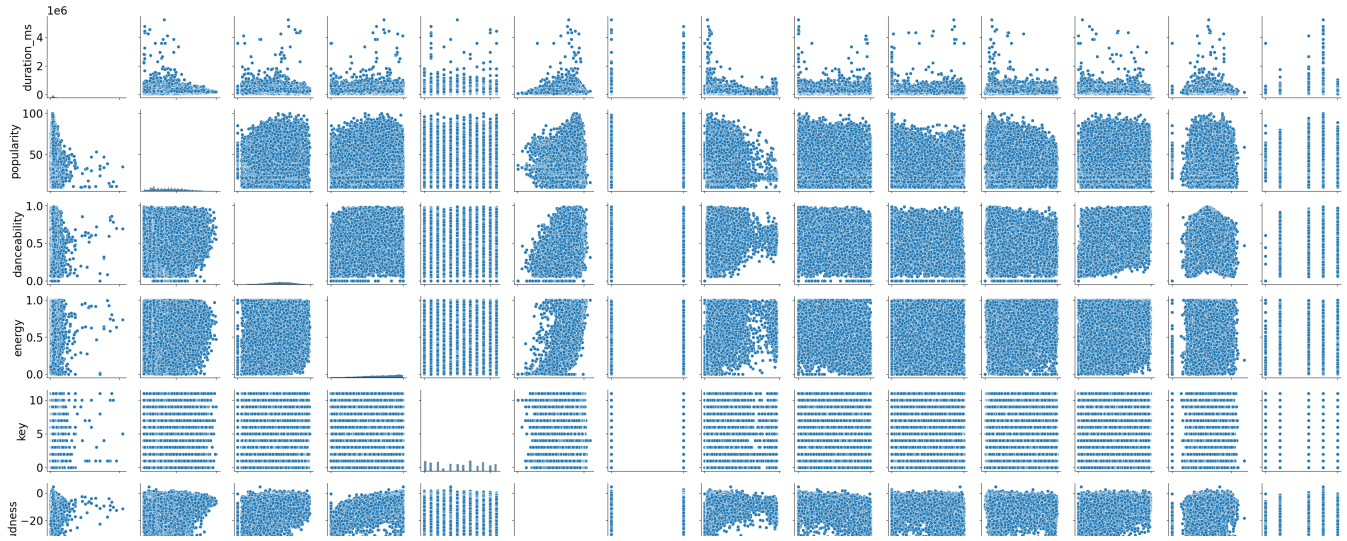
```
<seaborn.axisgrid.FacetGrid at 0x7a84d14d8e50>
```




```
merge['mult_artists'] = merge['artists'].str.contains('[,;]').astype(int)
merge = merge.assign(is_popular=(merge['popularity'] >= 55).astype(int))
merge['num_artists'] = merge['artists'].str.count('[,;]') + 1
```

```
<ipython-input-93-152f23c3d074>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  merge['mult_artists'] = merge['artists'].str.contains('[,;]').astype(int)
```

```
numeric_variables = ['duration_ms', 'popularity', 'danceability', 'energy', 'key',
                     'mode', 'speechiness', 'acousticness', 'instrumentalness', 'li
                     'valence', 'tempo', 'time_signature']

numeric_df = merge[numeric_variables]


sns.pairplot(numeric_df)
plt.show()
```




```
merge.head(3)
```

	Unnamed: 0	track_id	artists	album_name	track_name	popul
0	0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	
1	1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	
2	2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	

3 rows × 23 columns

```
len(merge[merge.is_popular==1])
```

```
17348
```

```
# Select rows with is_popular = 0
df_0 = merge[merge['is_popular'] == 0]

# Select rows with is_popular = 1
df_1 = merge[merge['is_popular'] == 1]

# Concatenate the two DataFrames
new_df = pd.concat([df_0, df_1])


new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 78367 entries, 12 to 92231
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
```

```
 0   Unnamed: 0        78367 non-null  int64
 1   track_id          78367 non-null  object
 2   artists           78367 non-null  object
 3   album_name        78367 non-null  object
 4   track_name        78367 non-null  object
 5   popularity        78367 non-null  float64
 6   duration_ms       78367 non-null  float64
 7   danceability      78367 non-null  float64
 8   energy            78367 non-null  float64
 9   key               78367 non-null  float64
 10  loudness          78367 non-null  float64
 11  mode              78367 non-null  float64
 12  speechiness       78367 non-null  float64
 13  acousticness      78367 non-null  float64
 14  explicit          78367 non-null  bool
 15  instrumentalness  78367 non-null  float64
 16  liveness          78367 non-null  float64
 17  valence           78367 non-null  float64
 18  tempo             78367 non-null  float64
 19  time_signature    78367 non-null  float64
 20  mult_artists      78367 non-null  int64
 21  is_popular        78367 non-null  int64
 22  num_artists       78367 non-null  int64
dtypes: bool(1), float64(14), int64(4), object(4)
memory usage: 13.8+ MB
```

10.1 Correlation Matrix:

```
new_df = new_df.drop(['popularity','Unnamed: 0'], axis=1)
new_df.corr()
```

```
<ipython-input-98-4918099cc672>:2: FutureWarning: The default value of numeric_c
  new_df.corr()
```

	duration_ms	danceability	energy	key	loudness	mode
duration_ms	1.000000	-0.062071	0.062250	0.011673	0.019747	-0.032268
danceability	-0.062071	1.000000	0.093849	0.030124	0.236944	-0.063687
energy	0.062250	0.093849	1.000000	0.041144	0.754808	-0.079942
key	0.011673	0.030124	0.041144	1.000000	0.034088	-0.141729
loudness	0.019747	0.236944	0.754808	0.034088	1.000000	-0.045442
mode	-0.032268	-0.063687	-0.079942	-0.141729	-0.045442	1.000000

```
new_df.columns
```

```
Index(['track_id', 'artists', 'album_name', 'track_name', 'duration_ms',
       'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'explicit', 'instrumentalness', 'liveness', 'valence',
       'tempo', 'time_signature', 'mult_artists', 'is_popular', 'num_artists'],
      dtype='object')
```

liveness	0.012553	-0.140426	0.190844	-0.000923	0.074078	0.018237

```
#new_df = new_df.select_dtypes(include='number')
new_df = new_df.drop(['album_name', 'track_name','track_id', 'artists'], axis=1)
new_df.columns
```

```
Index(['duration_ms', 'danceability', 'energy', 'key', 'loudness', 'mode',
       'speechiness', 'acousticness', 'explicit', 'instrumentalness',
       'liveness', 'valence', 'tempo', 'time_signature', 'mult_artists',
       'is_popular', 'num_artists'],
      dtype='object')
```

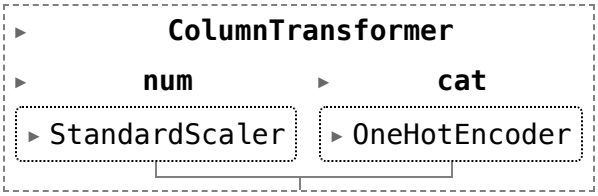
num_artists	0.026479	0.057736	-0.029227	0.006112	0.002638	-0.036730

11. Preprocessing Pipeline:

11.1 Transforming numeric and categorical variables

```
X = new_df.drop('is_popular', axis=1)
y = new_df['is_popular'].copy()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .30, random_s
```

```
from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer, make_column_selector

from sklearn import set_config
set_config(display='diagram') # shows the pipeline graphically when printed

# Define the columns that need to be scaled
numeric_features = ['instrumentalness', 'energy', 'liveness', 'danceability', 'spee

# Define the columns that need to be one-hot encoded
categorical_features = ['key', 'mult_artists', 'num_artists', 'explicit','mode']
num_pipeline = Pipeline([
        #('imputer', SimpleImputer(strategy='median')),
        ('scaler', StandardScaler())
    ])
cat_pipeline = Pipeline([
        #('imputer', SimpleImputer(strategy='most_frequent')),
        ('cat_encoder', OneHotEncoder(drop='first',handle_unknown='ignore'))
    ])

prep_pipeline = ColumnTransformer([
    ('num', num_pipeline, numeric_features),
    ('cat', cat_pipeline, categorical_features)
    ])


prep_pipeline
```



For numerical data we are using StandardScalar while for categorical we are using OneHotEncoder

11.2 Feature Selection

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif

kbest = SelectKBest(score_func=f_classif, k=10)

k_best_pipe = Pipeline([
    ('prep', prep_pipeline),
    ('select', kbest)
])

k_best_pipe.fit(X_train, y_train)

selected_features = k_best_pipe.named_steps['prep'].get_feature_names_out()[k_best_
print(f'The selected features are {selected_features}')
```

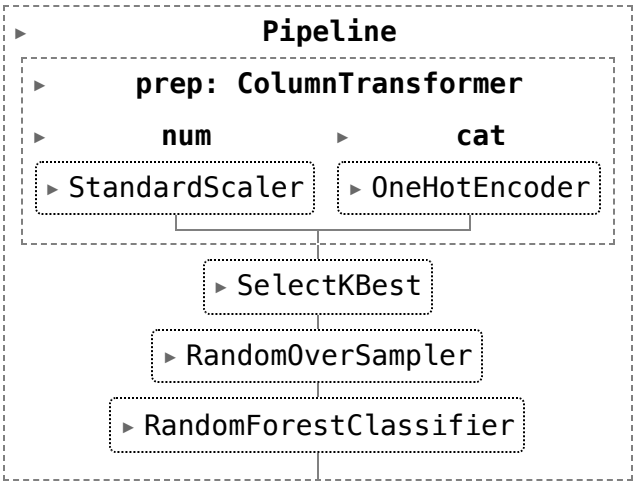
```
The selected features are ['num__instrumentalness' 'num__liveness' 'num__danceab
 'num__duration_ms' 'num__loudness' 'num__acousticness'
 'cat__mult_artists_1' 'cat__num_artists_2' 'cat__num_artists_3'
 'cat__explicit_True']
```

```
def print_scores(y_test, y_pred):
  plt.rc("font", size=20)
  cmd = ConfusionMatrixDisplay.from_predictions(y_test, y_pred, cmap=plt.cm.Blues, 
  accuracy = accuracy_score(y_test, y_pred)
  balanced_accuracy = balanced_accuracy_score(y_test, y_pred)
  print(f'Accuracy={accuracy:.4f}, Balanced Accuracy={balanced_accuracy:.4f}')
  recall = recall_score(y_test, y_pred, pos_label=1)
  f1 = f1_score(y_test, y_pred, pos_label=1)
  precision = precision_score(y_test, y_pred)
  print(f'Precision={precision:.4f}, Recall={recall:.4f}, F1-score={f1:.4f}')
```

12. Models

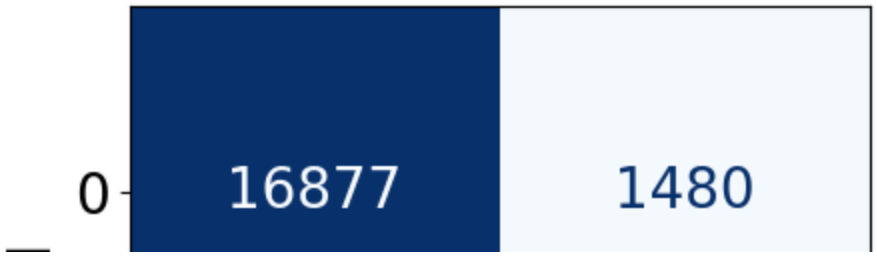
12.1 Random Forest Classifier

```
from imblearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import RandomOverSampler

rfc_pipeline = Pipeline([
    ('prep', prep_pipeline),
    ('select', SelectKBest(score_func=f_classif, k=10)),
    ('sample', RandomOverSampler(random_state=42) ),
    ('model', RandomForestClassifier())
])

rfc_pipeline.fit(X_train, y_train)
```




```
y_pred = rfc_pipeline.predict(X_test)
print_scores(y_test, y_pred)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
Accuracy=0.7726, Balanced Accuracy=0.5846
Precision=0.4653, Recall=0.2499, F1-score=0.3252
```



12.1.1 Halving Random Search

```
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingRandomSearchCV
from sklearn.experimental import enable_halving_search_cv
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import RandomOverSampler

rfc_pipeline = Pipeline([
    ('prep', prep_pipeline),
    ('select', SelectKBest(score_func=f_classif, k=10)),
    ('sample', RandomOverSampler(random_state=42)),
    ('model', RandomForestClassifier())
])

param_distributions = {
    'model__max_depth': range(5, 16),
    'model__min_samples_leaf': range(1, 4)
}

halving_random_search = HalvingRandomSearchCV(
    rfc_pipeline,
    param_distributions=param_distributions,
    n_candidates='exhaust',
    factor=2,
    random_state=42,
    scoring='balanced_accuracy'
)

halving_random_search.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:305:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
```

```
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
```

```
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
```

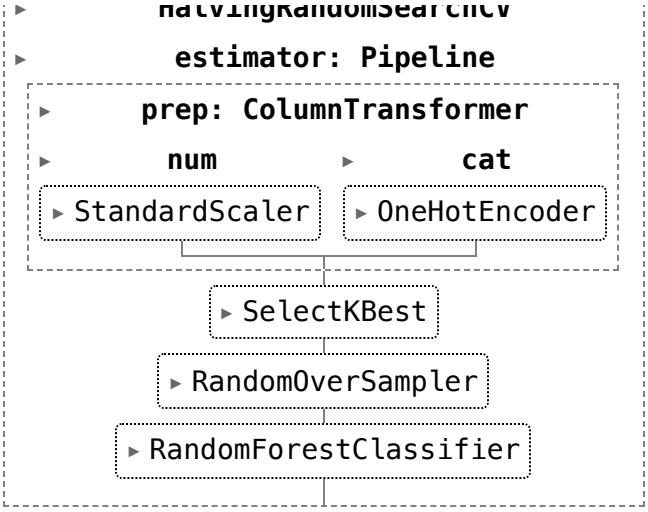
```
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
```

```
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
```

```
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarning
/usr/local/lib/python3.10/dist-packages/sklearn/feature_selection/_univariate_se
  f = msb / msw
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:2184:
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
```

HalvingRandomSearchCV




```
print("Best parameters: ", halving_random_search.best_params_)
print(f'Best score: {halving_random_search.best_score_}')
```

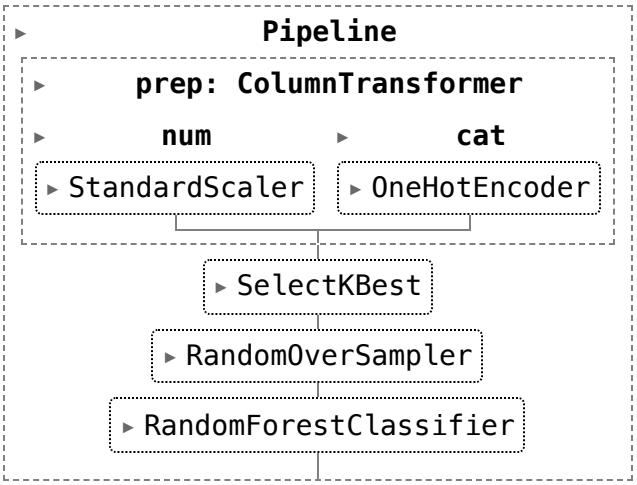
```
Best parameters:  {'model__min_samples_leaf': 2, 'model__max_depth': 5}
Best score: 0.5858606603668304
```

12.1.2 Halving Grid Search

```
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingGridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest
from sklearn.preprocessing import StandardScaler
from imblearn.pipeline import Pipeline

# Create the pipeline
rfc_pipeline = Pipeline([
    ('prep', prep_pipeline),
    ('select', SelectKBest(score_func=f_classif, k=10)),
    ('sample', RandomOverSampler(random_state=42)),
    ('model', RandomForestClassifier())
])

# Define the parameter grid
param_grid = {
    'model__max_depth': [5, 10, 15],
    'model__min_samples_leaf': [1, 2, 4]
}

# Create the HalvingGridSearchCV object
halving_grid_search = HalvingGridSearchCV(
    rfc_pipeline, param_grid, scoring='balanced_accuracy', n_jobs=-1, cv=5, factor=
)

# Fit the HalvingGridSearchCV object
halving_grid_search.fit(X_train, y_train)

# Print the best parameters and score
print(f'Best parameters: {halving_grid_search.best_params_}')
print(f'Best score: {halving_grid_search.best_score_}')
```

```
Best parameters: {'model__max_depth': 10, 'model__min_samples_leaf': 2}
Best score: 0.6430663175269392
```

```
from imblearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import RandomOverSampler

rfc_pipeline = Pipeline([
    ('prep', prep_pipeline),
    ('select', SelectKBest(score_func=f_classif, k=10)),
    ('sample', RandomOverSampler(random_state=42) ),
    ('model', RandomForestClassifier(min_samples_leaf = 2, max_depth = 11))
])

rfc_pipeline.fit(X_train, y_train)
```




```
y_pred = rfc_pipeline.predict(X_test)
print_scores(y_test, y_pred)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
Accuracy=0.6170, Balanced Accuracy=0.6436
Precision=0.3246, Recall=0.6911, F1-score=0.4417
```

The balanced accuracy score of 0.6436 obtained from the Random Forest Classifier suggests that the model performs reasonably well in handling the imbalanced nature of the popularity distribution in the dataset. This score represents the average of sensitivity (true positive rate) and specificity (true negative rate), providing an overall assessment of the model's ability to make accurate predictions for both popular and non-popular songs.

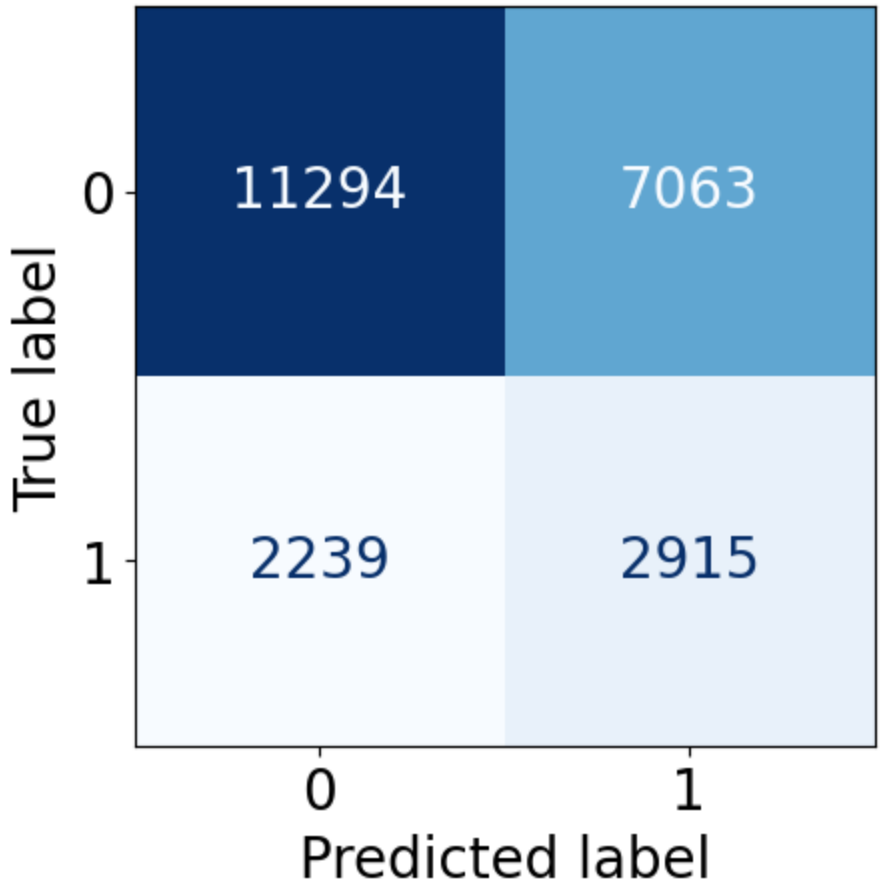
From the confusion matrix, the true negatives (10,944) and true positives (3,562) represent instances where the model correctly identified non-popular and popular songs, respectively. On the other hand, false negatives (7,413) and false positives (1,592) indicate instances where the model misclassified songs as non-popular or popular.

In the hyperparameter tuning phase, the optimized configuration for the model was identified with a maximum depth of 10 and a minimum sample leaf of 2, resulting in the best score of 0.64306. These parameters strike a balance between capturing the complexity of the data (max depth) and preventing overfitting (min depth).

12.2 K Nearest Neighbour

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from imblearn.pipeline import Pipeline

# Create a pipeline for KNN classifier
# f_classif is employed, which stands for the F-statistic.
# The F-statistic assesses the linear dependency between the feature and the target
# or each feature, computes the ratio of the variance between classes to the varian

knn_pipeline = Pipeline([
    ('prep', prep_pipeline),
    ('select', SelectKBest(score_func=f_classif, k=10)),
    ('sample', RandomOverSampler(random_state=42) ),
    ('knn', KNeighborsClassifier())  # KNN classifier
])

# Train the model
knn_pipeline.fit(X_train, y_train)

y_pred = knn_pipeline.predict(X_test)
print_scores(y_test, y_pred)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
Accuracy=0.6044, Balanced Accuracy=0.5904
Precision=0.2921, Recall=0.5656, F1-score=0.3853
```



12.2.1 Halving Grid Search

```
# Import the necessary modules
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingGridSearchCV
from sklearn.neighbors import KNeighborsClassifier

# Define the parameter grid
param_grid = {
    'knn__n_neighbors': [7 , 9, 11],
    'knn__weights': ['uniform', 'distance']
}

# HalvingGridSearchCV for hyperparameter tuning
halving_grid_search = HalvingGridSearchCV(knn_pipeline, param_grid, cv=5, scoring='
halving_grid_search.fit(X_train, y_train)

best_params_halving_grid = halving_grid_search.best_params_
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
```

```
best_knn_model_grid = halving_grid_search.best_estimator_
best_knn_model_grid.fit(X_train, y_train)

y_pred = best_knn_model_grid.predict(X_test)
print_scores(y_test, y_pred)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
Accuracy=0.6115, Balanced Accuracy=0.6051
Precision=0.3029, Recall=0.5937, F1-score=0.4012
```

12.2.2 Grid Search

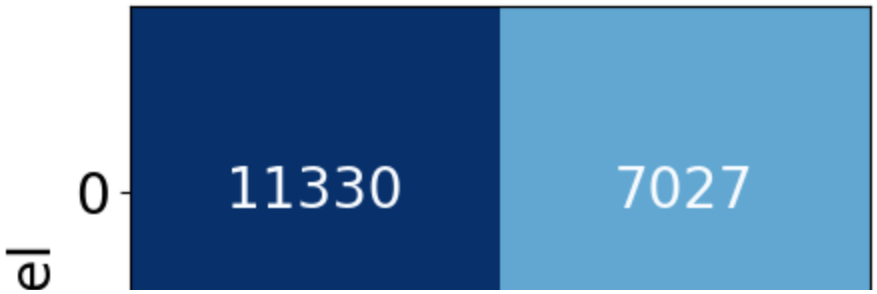
```
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import train_test_split, GridSearchCV, RandomizedSearcl
from sklearn.neighbors import KNeighborsClassifier

param_grid = {
    'knn__n_neighbors': [7 , 9, 11],
    'knn__weights': ['uniform', 'distance']
}

grid_search = GridSearchCV(knn_pipeline, param_grid, cv=5, scoring='balanced_accura
grid_search.fit(X_train, y_train)

best_params_grid = grid_search.best_params_

print("Best Hyperparameters (Grid Search):", best_params_grid)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
Best Hyperparameters (Grid Search): {'knn__n_neighbors': 7, 'knn__weights': 'dis
```

```
# Train the model with the best hyperparameters from GridSearchCV
best_knn_model_grid = grid_search.best_estimator_
best_knn_model_grid.fit(X_train, y_train)

# Predictions on the test set after GridSearchCV
y_pred = best_knn_model_grid.predict(X_test)
print_scores(y_test, y_pred)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:202:
  warnings.warn(
Accuracy=0.6065, Balanced Accuracy=0.5927
Precision=0.2942, Recall=0.5683, F1-score=0.3877
```



12.2.3 Random Search

```
from scipy.stats import randint
from imblearn.pipeline import Pipeline
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingRandomSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.feature_selection import SelectKBest
from imblearn.over_sampling import RandomOverSampler
from sklearn.feature_selection import f_classif

knn_pipeline = Pipeline([
    ('prep', prep_pipeline),
    ('select', SelectKBest(score_func=f_classif, k=10)),
    ('sample', RandomOverSampler(random_state=42)),
    ('model', KNeighborsClassifier())
])

param_distributions = {
    'model__n_neighbors': randint(5,13),
    'model__weights': ['uniform', 'distance']
}

halving_random_search_knn = HalvingRandomSearchCV(
    knn_pipeline,
    param_distributions=param_distributions,
    n_candidates='exhaust',
    factor=2,
    random_state=42,
    scoring='balanced_accuracy'
)

halving_random_search_knn.fit(X_train, y_train)
```