```
Author:
Created on Sun Dec 8 19:43:57 2019

Dataset Description: Spotify

Data Set Information:

This dataset is composed of a range of acoustic and popularity features
for tracks from 26 different categories. Each track is described by
its popularity, danceability, energy, key, loudness, mode, speechiness,
instrumentalness, liveness, valence, tempo, duration and time_signature.

Tracks in the same classes matched tempo, conditions, duration, and
total number of tracks in each class. Popularity is calculated by
total number of streams and how recent they are.

Each track represented by one of 1,475 tracks recording band album information.

Attribute Information:
- 1 - Artist_name: Artist name (nominal, 1,475 different artists)
- 2 - Track_name: Track name (nominal)
- 3 - Popularity: Popularity of the track. The higher value is the
    number of plays across environment.
- 4 - Danceability: Danceability measure (ratio)
- 5 - Energy: Energy measure (ratio)
- 6 - Key: Key the track is in. Integers map to pitches using standard
    Pitch Class notation. E.g. 0 = C, 1 = C#/D♭, 2 = D, and so on. If no
    key was detected, the value is -1. (integer)
- 7 - Loudness: Loudness measure (dB)
- 8 - Mode: Mode indicates the modality (major or minor) of a track,
    the type of scale from which its melodic content is derived.
    Major is represented by 1 and minor is 0. (binary)
- 9 - Speechiness: Speechiness measure (ratio)
- 10 - Instrumentalness: Predicts whether a track contains no vocals.
    "Ooh" and "aah" sounds are treated as instrumental in this context.
    Rap or spoken word tracks are clearly "vocal". (ratio)
- 11 - Liveness: Detection of the presence of an audience in the recording.
    Higher liveness values represent an increased probability that the
    track was performed live. (ratio)
- 12 - Valence: Measure of mood of a track. Tracks with high valence
    sound more positive (happy, cheerful, euphoric), while tracks with
    low valence sound more negative (sad, depressed, angry). (ratio)
- 13 - Tempo: Tempo measure (bpm)
- 14 - Duration_ms: Duration of the track in milliseconds (ms)
- 15 - Time_signature: Estimated time signature. The time signature (meter)
    is a notational convention to specify how many beats are in each bar
    (or measure). The time signature ranges from 3 to 7 indicating time
    signatures of "3/4", to "7/4". (integer)
- 16 - Class: Class (nominal, 26 different classes)

Source Information:
-Original Owner: Rafael Ferreira (rafaelmferreira@gmail.com)
-Donated by: Rafael Ferreira (rafaelmferreira@gmail.com)
-Date: November, 2019

Relevant Papers:
Kim, Y., Suh, B. and Lee, K. (2014). Poster: Spotify Chart Challenge: A
Preliminary Study. In Proceedings of the 8th ACM Conference on
Recommender Systems (RecSys '14), October 6-10, 2014, Foster City,
Silicon Valley, USA. ACM, New York, NY, USA, 2 pages.
http://dx.doi.org/10.1145/2645710.2645753

Citation Request:
Rafael Ferreira (rafaelmferreira@gmail.com)

Number of Instances: 232,725

Number of Attributes: 16 (15 continuous + 1 nominal)

Attribute Information:
1. Artist_name: Artist name (nominal, 1,475 different artists)
2. Track_name: Track name (nominal)
3. Popularity: Popularity of the track. The higher value is the
    number of plays across environment.
4. Danceability: Danceability measure (ratio)
5. Energy: Energy measure (ratio)
6. Key: Key the track is in. Integers map to pitches using standard
    Pitch Class notation. E.g. 0 = C, 1 = C#/D♭, 2 = D, and so on. If no
    key was detected, the value is -1. (integer)
7. Loudness: Loudness measure (dB)
8. Mode: Mode indicates the modality (major or minor) of a track,
    the type of scale from which its melodic content is derived.
    Major is represented by 1 and minor is 0. (binary)
9. Speechiness: Speechiness measure (ratio)
10. Instrumentalness: Predicts whether a track contains no vocals.
    "Ooh" and "aah" sounds are treated as instrumental in this context.
    Rap or spoken word tracks are clearly "vocal". (ratio)
11. Liveness: Detection of the presence of an audience in the recording.
    Higher liveness values represent an increased probability that the
    track was performed live. (ratio)
12. Valence: Measure of mood of a track. Tracks with high valence
    sound more positive (happy, cheerful, euphoric), while tracks with
    low valence sound more negative (sad, depressed, angry). (ratio)
13. Tempo: Tempo measure (bpm)
14. Duration_ms: Duration of the track in milliseconds (ms)
15. Time_signature: Estimated time signature. The time signature (meter)
    is a notational convention to specify how many beats are in each bar
    (or measure). The time signature ranges from 3 to 7 indicating time
    signatures of "3/4", to "7/4". (integer)
16. Class: Class (nominal, 26 different classes)

Missing Attribute Values: None

Class Distribution:
Classical, Country, Dance, Electronic, Anime, Jazz, Alternative, Blues,
Children's Music, Comedy, Hip-Hop, Indian, Opera, Rock, Soul, Reggae,
Pop, Ska, Soundtrack, World, Blues, Classical, Country, Dance,
Electronic, Anime, Jazz, Alternative, Blues, Children's Music, Comedy,
Hip-Hop, Indian, Opera, Rock, Soul, Reggae, Pop, Ska, Soundtrack, World
```

```
import pandas as pd
import numpy as np
import matplotlib
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import precision_recall_curve, average_precision_score
from sklearn.metrics import cohen_kappa_score, matthews_corrcoef, log_loss
from sklearn.metrics import classification_report, precision_recall_fscore_support
import warnings
warnings.filterwarnings('ignore')
```

```
Using TensorFlow backend.
```

```
df = pd.read_csv('/content/drive/My Drive/spotify.csv')
df.shape
```

```
(232725, 16)
```

```
df.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex: 232725 entries, 0 to 232724
Data columns (total 16):
artist_name         232725 non-null object
track_name          232724 non-null object
popularity          232725 non-null int64
danceability        232725 non-null float64
energy              232725 non-null float64
key                 232725 non-null int64
loudness            232725 non-null float64
mode                232725 non-null int64
speechiness         232725 non-null float64
instrumentalness    232725 non-null float64
liveness            232725 non-null float64
valence             232725 non-null float64
tempo               232725 non-null float64
duration_ms         232725 non-null int64
time_signature      232725 non-null int64
class               232725 non-null object
dtypes: float64(9), int64(6), object(1)
memory usage: 28.4+ MB
```

```
df.describe()
```

	popularity	danceability	energy	key	loudness	mode	speechiness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature
count	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000	232725.000000
mean	41.128502	0.558159	0.578828	5.471063	-8.580862	0.640910	0.093339	0.156943	0.213772	0.455602	117.307471	230451.172031	3.904035
std	18.580475	0.178731	0.251661	3.561726	5.844094	0.479748	0.093573	0.309555	0.189549	0.260928	35.422309	56363.988649	0.432562
min	0.000000	0.000564	0.000020	0.000000	-52.457000	0.000000	0.022200	0.000000	0.012000	0.000375	30.379000	15000.000000	1.000000
25%	30.000000	0.435000	0.383000	2.000000	-10.013000	0.000000	0.035900	0.000000	0.098800	0.250000	88.282000	203091.000000	4.000000
50%	43.000000	0.569000	0.587000	5.000000	-7.223000	1.000000	0.048900	0.000034	0.137000	0.455000	116.989000	227933.000000	4.000000
75%	54.000000	0.690000	0.784000	8.000000	-5.331000	1.000000	0.084200	0.082500	0.294000	0.659000	141.969000	263960.000000	4.000000
max	100.000000	0.966000	0.999000	11.000000	1.342000	1.000000	0.933000	0.996000	0.996000	0.984000	220.062000	5552917.000000	5.000000

```
df.describe(include=['object'])
```

	artist_name	track_name	class
count	232725	232724	232725
unique	1475	148904	26
top	Giuseppe Verdi	Home	Classical
freq	2721	2490	9000

```
df.isnull().sum()
```

```
artist_name         0
track_name          1
popularity          0
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
instrumentalness    0
liveness            0
valence             0
tempo               0
duration_ms         0
time_signature      0
class               0
dtype: int64
```

```
df[df.track_name.isnull()]
```

	artist_name	track_name	popularity	danceability	energy	key	loudness	mode	speechiness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	class
232724	Giuseppe Verdi	NaN	0	0.000564	0.000020	0	-52.457	0	0.0222	0	0.0996	0.000375	30.379	15000	1	Classical

```
df.drop(df.track_name.isnull, inplace=True)
```

```
df.shape
```

```
(232724, 16)
```

```
df.duplicated().sum()
```

```
0
```

```
df.artist_name.nunique()
```

```
1475
```

```
df.class.value_counts()
```

```
Classical    9000
Country      9000
Dance        9000
Electronic   9000
Anime        9000
Jazz         9000
Alternative  9000
Blues        9000
Children's Music  9000
Comedy       9000
Hip-Hop      9000
Indian       9000
Opera        9000
Rock         9000
Soul         9000
Reggae       9000
Pop          9000
Ska          9000
Soundtrack   9000
World        9000
Blues        9000
Classical    9000
Country      9000
Dance        9000
Electronic   9000
Anime        9000
Jazz         9000
Alternative  9000
Blues        9000
Children's Music  9000
Comedy       9000
Hip-Hop      9000
Indian       9000
Opera        9000
Rock         9000
Soul         9000
Reggae       9000
Pop          9000
Ska          9000
Soundtrack   9000
World        9000
Name: class, dtype: int64
```

```
df.popularity.value_counts
```

```
```

```
In [4]: df['Top-level-categories'] = df['Top-level-categories'].map(lambda x: x.rstrip('|'))
        df['Top-level-categories'].value_counts()
```

```
Out[4]: Top-level-categories
        Food                                                    2
        Food|Food-products                                      2
        Food|Food-products|Food-by-type                         2
        Food|Food-products|Food-by-type|Fruit-and-vegetable     2
        Food|Food-products|Food-by-type|Fruit-and-vegetable|    2
        Food|Food-products|Food-by-type|Fruit-and-vegetable|    2
        Food|Food-products|Food-by-type|Fruit-and-vegetable|    2
        Food|Food-products|Food-by-type|Fruit-and-vegetable|    2
        Food|Food-products|Food-by-type|Fruit-and-vegetable|    2
        Food|Food-products|Food-by-type|Fruit-and-vegetable|    2
        Name: count, Length: 1000, dtype: int64
```

```
In [5]: df['Top-level-categories'].unique()
```

```
Out[5]: array(['Food', 'Food|Food-products', 'Food|Food-products|Food-by-type',
        'Food|Food-products|Food-by-type|Fruit-and-vegetable',
        'Food|Food-products|Food-by-type|Fruit-and-vegetable|
```