Spotify Hit Song Analysis and Prediction

Is it possible to predict a song's potential popularity before it is released? Through analyzing key features of over 6,000 songs from Spotify I hope to discover what these popular songs have in common, and be able to predict whether a song has the potential to become a hit.

My clients for this project are record industry professionals and songwriters who are interested in creating hit songs, as well as evaluating unreleased songs for potential commercial success. Many times a recording artist will be presented with a selection of songs to choose from, and this project gives quantifiable metrics to assess the potential success on a song by song basis.

For this project I will be using the "Spotify Hit Predictor" dataset. (https://www.kaggle.com/theoverman/the-spotify-hit-predictor-dataset) This dataset contains audio features from over 6,000 songs ranging from the 1960s to the 2010s, as well as a "Target" column which notes if the song was featured in the Billboard Top 100 charts or not.

It is important to note that this project solely focuses on the musical features of this data - there are many factors which contribute to a song's success in the Billboard charts such as marketing, promotion, image of the artist, and more. I aim to assess a song's "Hit Potential" based entirely on the musical features. The resulting model will show if a song's musical content is similar to those which made the top 100.

Data Wrangling

For this project, my main dataset is sourced from Kaggle and came in a very clean and usable format already. Initial inspection of the data in pandas revealed no missing values, and a clearly labeled, easy to work with dataframe. While some columns contain info about the songs, such as the artist and track name, most of the data is pertaining to musical features of the songs. The 'Target' column refers to whether the song made it into the Top 100 Billboard charts or not. I will refer to songs in the Billboard charts as 'Hit', and those that did not as 'Flop'.

The data came in 6 different csv files by decade ranging from the 1960s to the 2010s. All datasets contained the same columns and scoring metrics. To make a single dataframe containing data from all decades I used the pandas concatenate method.

The main features of the dataset outside of basic track info like artist and track name are Danceability, Energy, Key, Loudness, Mode, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo, Duration, Time Signature, Chorus_hit, and Sections.

Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

Key is the estimated overall key of the track. 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.

Loudness is the overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.

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.

Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

Acousticness is rated as a confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

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”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

Liveness detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

Valence is a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Tempo contains the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

Time Signature is an estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

Chorus Hit contains the the author's best estimate of when the chorus would start for the track. It is the timestamp of the start of the third section of the track (in milliseconds). This feature was extracted from the data received by the API call for Audio Analysis of that particular track.

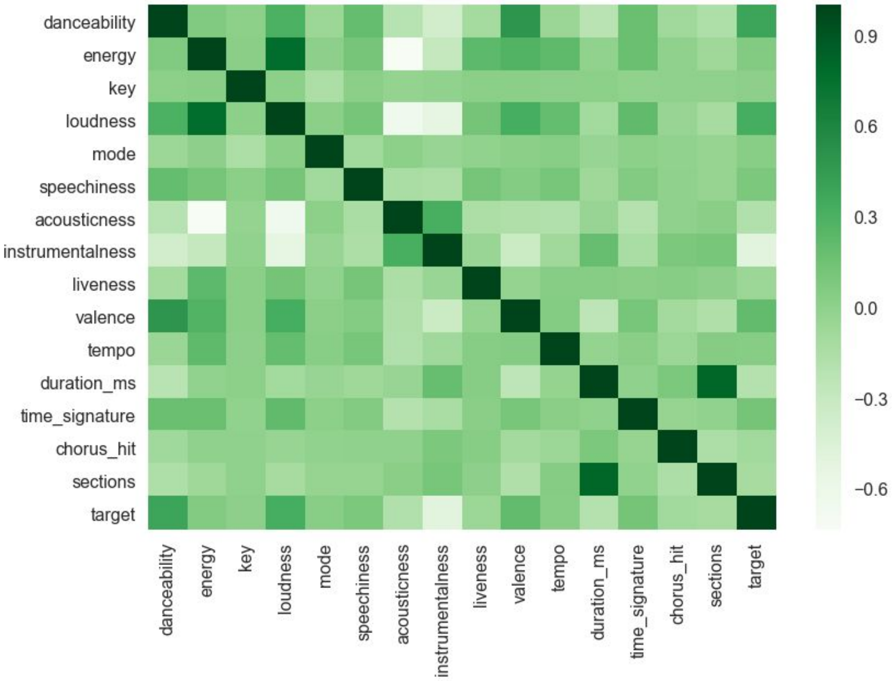
Sections is the number of sections the particular track has. This feature was extracted from the data received by the API call for Audio Analysis of that particular track.

To check for outliers, I ran a quick boxplot using the seaborn library on all of the relevant columns in the dataset. I noticed a few outliers, specifically in the ‘tempo’, ‘speechiness’, and ‘instrumentalness’ columns. I decided to leave these outliers as is, due to the nature of this project and music in general variation such as this is to be expected.

Exploratory Data Analysis

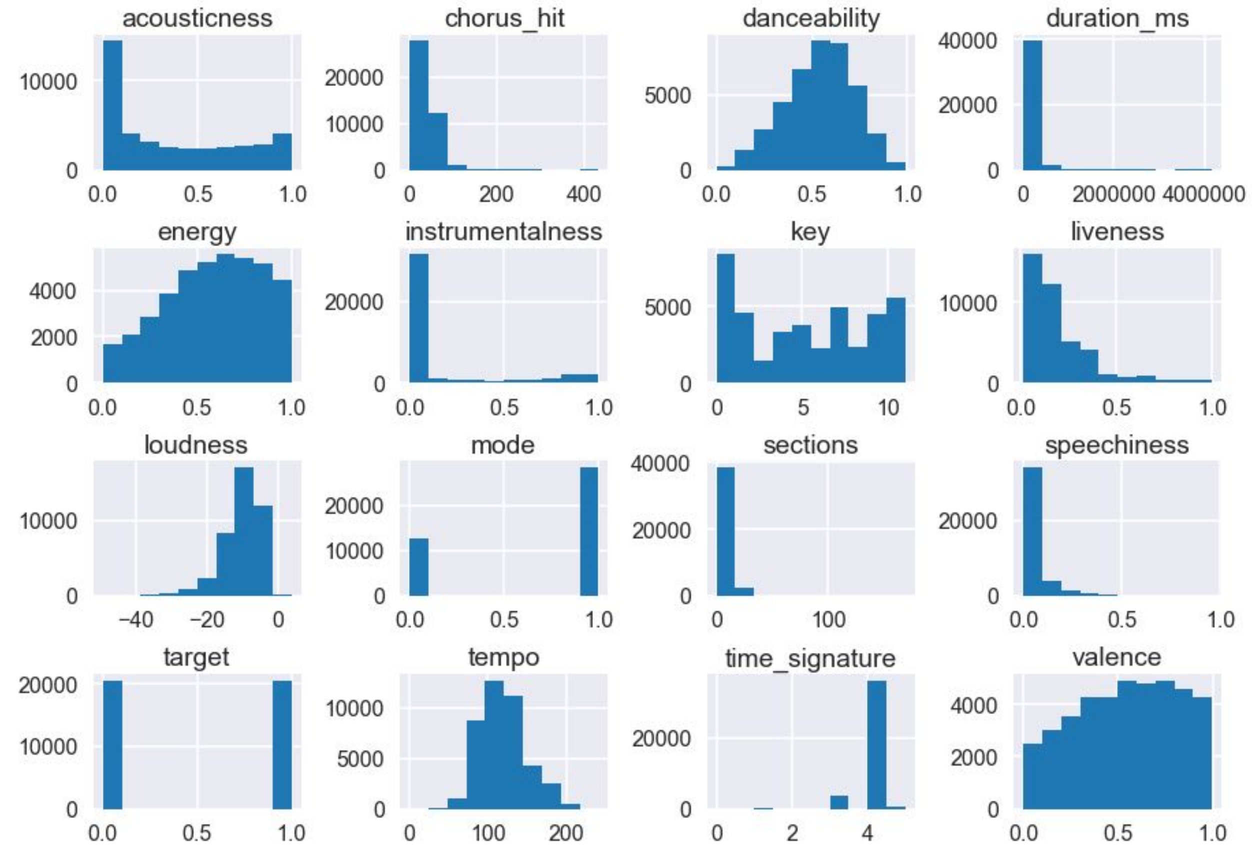
Initially I began EDA by making a heatmap of the data to identify possible correlations between features, and to examine which features appear to influence the target variable most. The strongest correlation shown was between the energy and loudness variables. Unsurprisingly, the number of sections per song and it’s duration are correlated. There was also a correlation between danceability and valence, indicating that tracks with more positive lyrical content are more danceable.

I also observed negative correlations between loudness and acousticness, as well as acousticness and energy. This suggests that songs with more electronic instrumentation as opposed to traditional instruments are generally louder and more energetic.



The strongest correlations with the target variable were danceablility and loudness. Valence also had a slight correlation. This suggests that dance songs are most likely to have hit song potential, especially if they have a positive message. The loudness correlation is most likely due to a phenomenon known in the music industry as the “Loudness Wars”, where songs have been mastered progressively louder over time to stand out to the listener over other songs. The only obvious negative correlation with the target was instrumentalness, indicating that it is highly unlikely for a song with a small amount of vocals or an all instrumental song to be a hit.

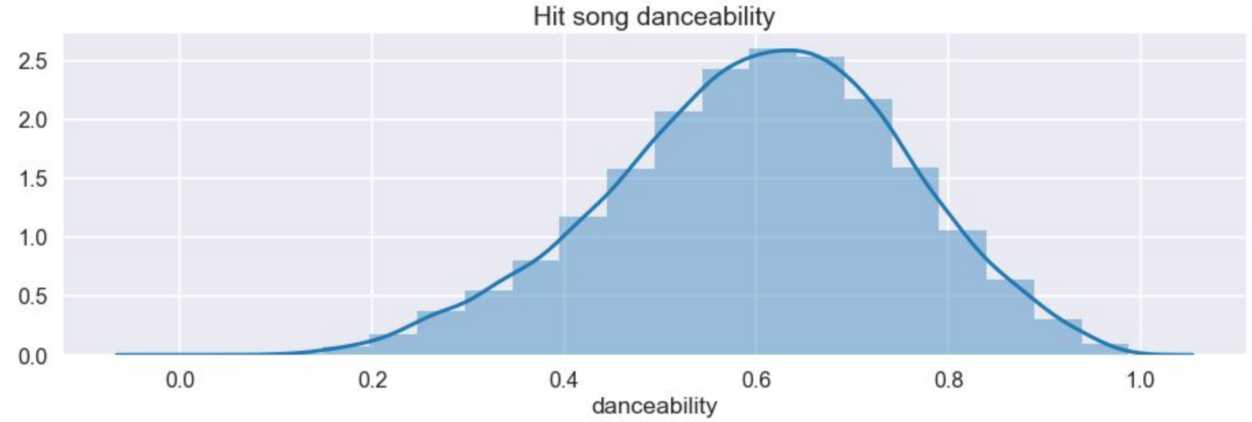
I then examined the distribution of each feature. Interesting standouts from this are that major keys appear to be more popular than minor keys, though the key of a song does not appear to affect its relationship with the target variable. Also of note was the fact that the key of D# appears significantly less in this dataset than other keys, which were fairly uniformly distributed.

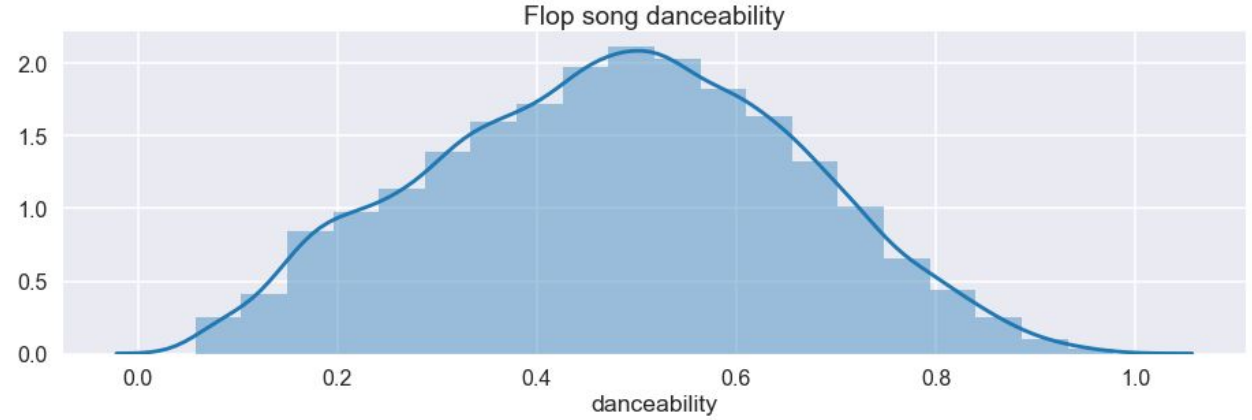


I also examined each feature by decade to identify changes over time. Acousticness has significantly decreased over the decades due to the increasing popularity of electronic instrumentation. Song duration has varied throughout time, with shorter songs in the 1960s and 70s. The overall duration increased in the 80s and 90s, but shortened again in the 2000s. In the 2010s song duration seems to be rising again. Energy has significantly increased over time, energy ratings in the 1960s are mostly clustered around 0.4, rising to almost 1.0 by the 2010s. Loudness has steadily increased, again most likely due to the “Loudness Wars” phenomenon. Valence has decreased over time, and it appears this trend started in the 1990s. Modern songs seem to have less positive lyrical content.

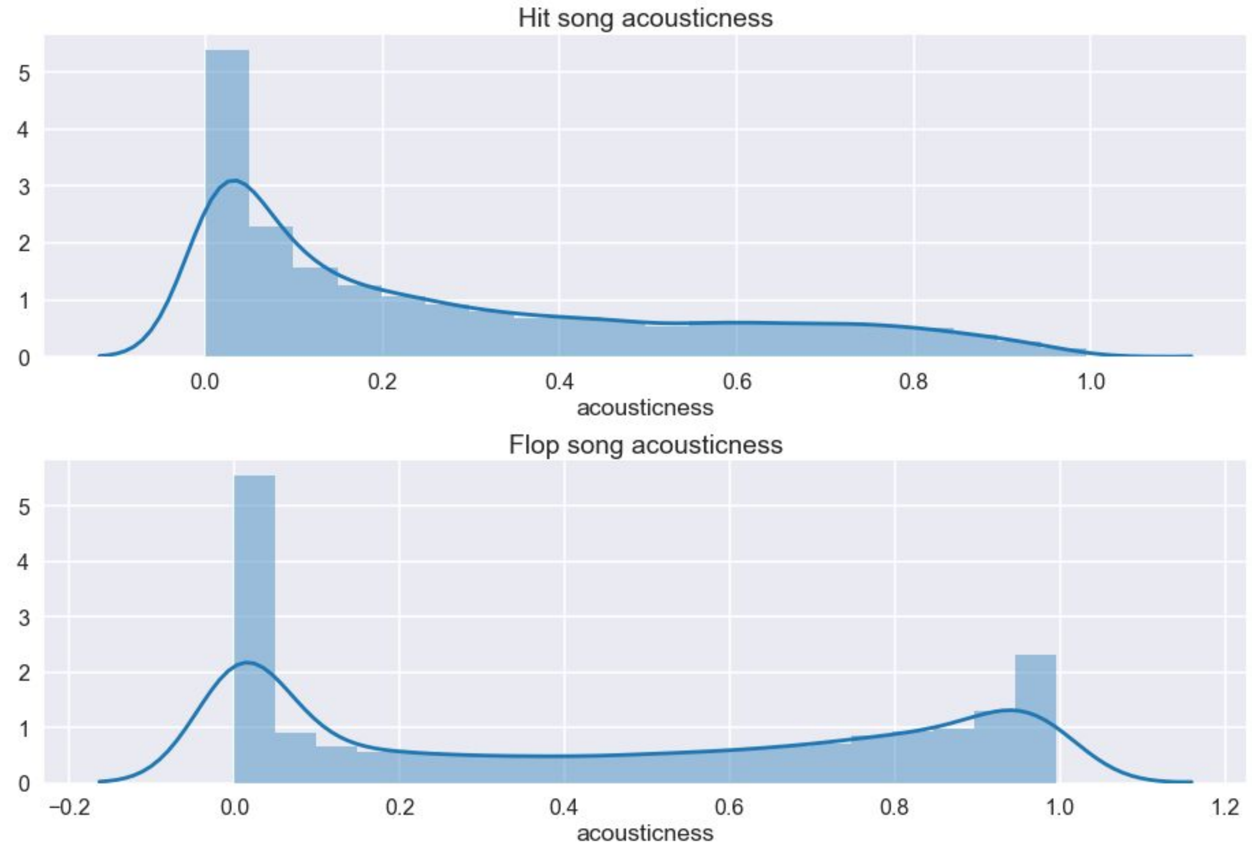
The heatmap was useful for identifying which features are most strongly correlated with the target, but I also compared histograms of feature between hit and flop songs to identify further differences. Some interesting trends that were not shown by the heatmap are that hit songs tend to have a lower acousticness rating overall. In hit songs, the chorus generally occurs earlier in the song. The duration of hit songs tends to be longer, and the instrumentalness rating is lower. The valence was also much higher.

I decided to further analyze some of the features which display the most difference between hit and flop songs, starting with danceability:



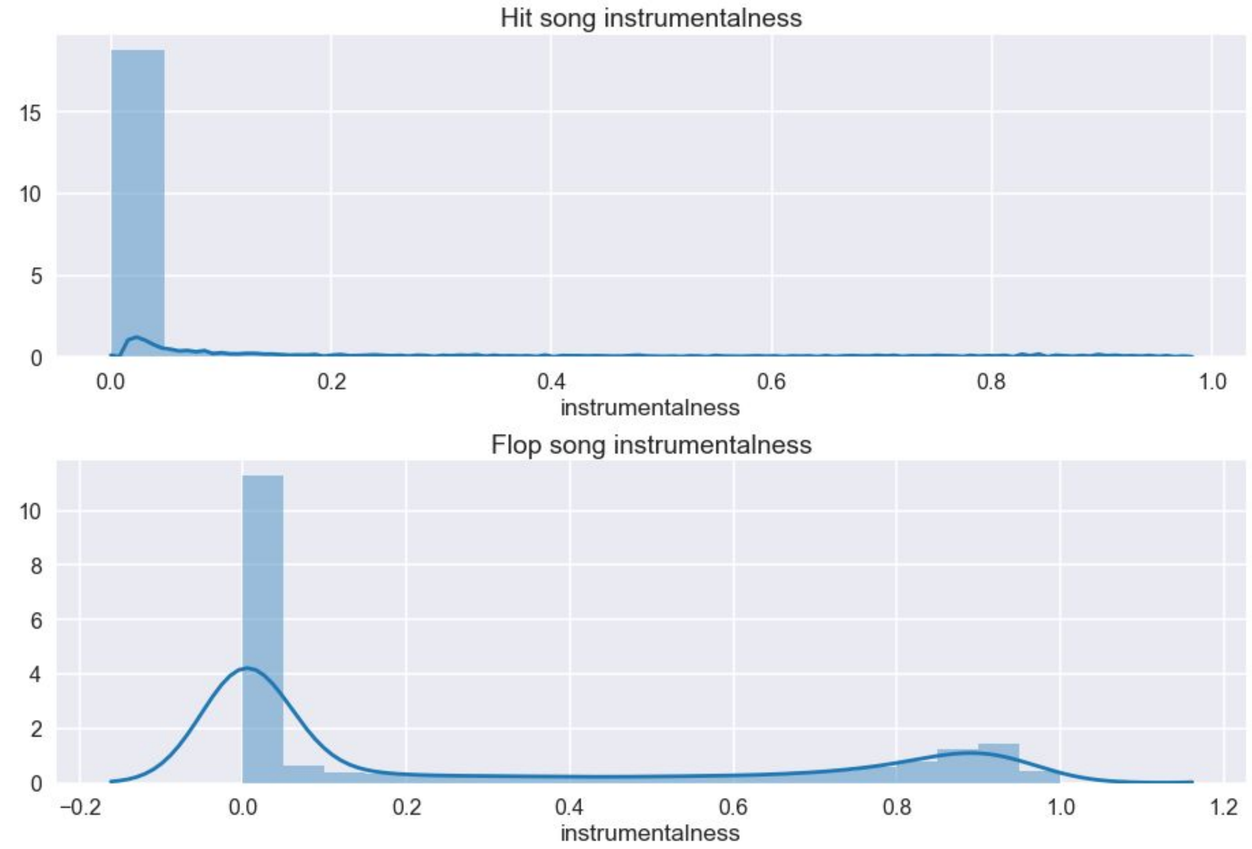


As we can see here, the peak of hit song danceability lies around 0.6, whereas flop song danceability is closer to 0.5, and the left tail of the distribution is much longer.

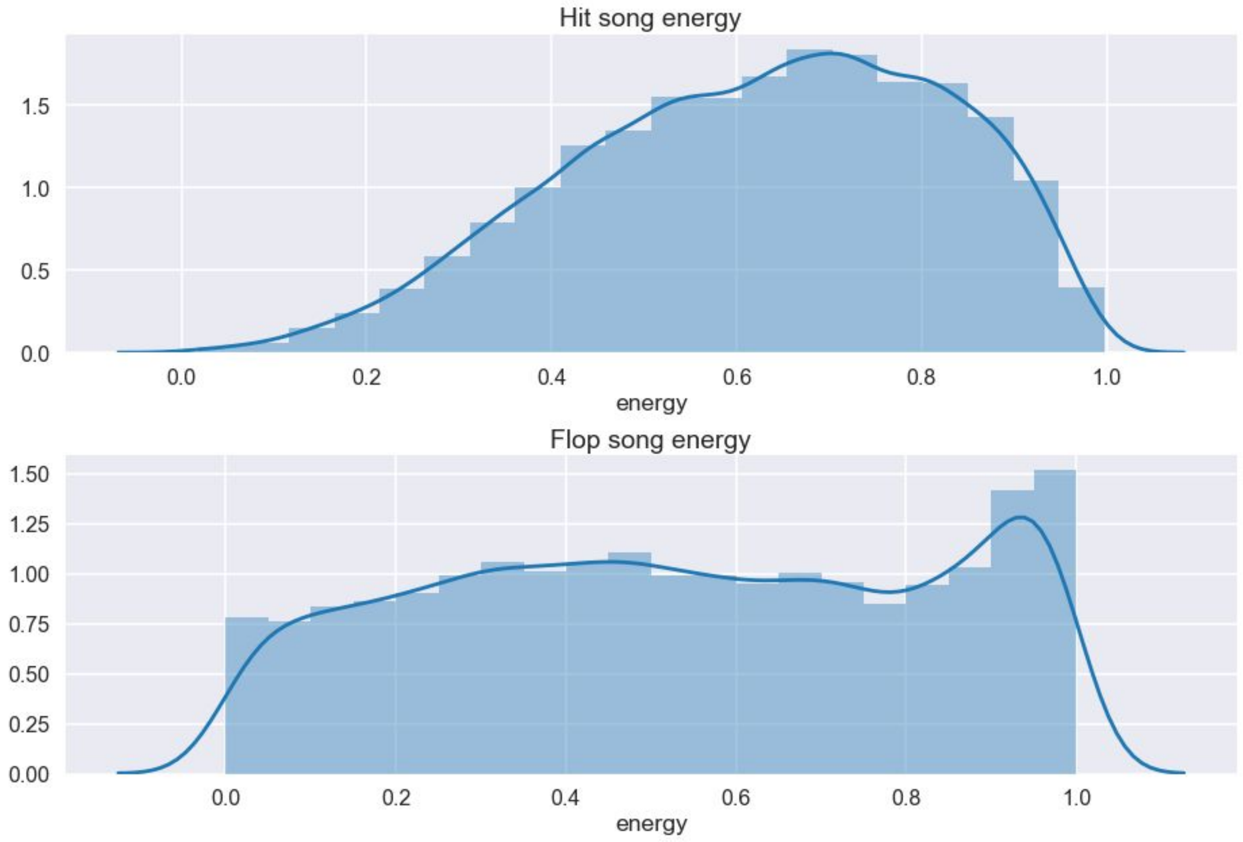


While the majority of both hit and flop songs are closer to 0 acousticness, we can see that there is a spike towards 1.0 in the flop songs. This indicates that songs with more electronic instruments are overwhelmingly more likely to be a hit than those with mainly acoustic instruments.

The instrumentalness category is very similar to acousticness in that the majority of songs in both categories have a rating of 0, there is a spike towards 1 in flop songs. This suggests that it is highly improbable that an instrumental song enter the billboard charts.



The energy feature is interesting in that the the difference between hit and flop songs is not quite as easy to quantify as with some of the other features. The peak of the distribution in hit songs is around 0.7, and while the left tail is longer it appears fairly close to the normal distribution. When looking at the flop songs, the values are relatively evenly distributed from 0 to 0.8, with a noticeable spike between 0.9 and 1.0. This suggests that there is a “sweet spot” for energy ratings for a song to become a hit - too much energy can be negatively associated with hit potential, but not enough also decreases the chance of making the billboard charts.



Statistical Inference

After visually inspecting the data, I used bootstrap statistics to determine the confidence interval of differences in several key features between hit and flop songs. I identified that the features with the most variance between hit and flop were danceability, loudness, instrumentalness, acousticness, chorus hit, valence, and number of sections.

The bootstrap inference revealed that the 95% confidence interval of difference in mean danceability between hit and flop songs is between 11 and 12 percent. This makes sense given that the heatmap identified danceability as the variable that correlated with the target most strongly.

The other confidence intervals further confirmed what I learned in the visual analysis, with the largest difference being displayed in the instrumentalness category, with a difference of between 24 and 25% lower instrumentalness in hit songs. This is unsurprising as visual analysis revealed that while the hit songs dataframe contained no entries over 0.5 instrumentalness, the flop dataframe, while still largely containing lower ratings did have some entries on the higher end of the range.

Machine Learning

To predict a song's hit potential, I initially tested accuracy with a random forest classifier. I split the data with a test size of 30%. The only tuning was iterating over a list of different estimator numbers. The untuned model got 73% accuracy on the data and did not appear to be overfitting the training set. I then inspected the feature importances of this relatively untuned model:

```
danceability : 0.1961608604247969
energy : 0.10186251245010265
key : 0.0
loudness : 0.10804753662129706
mode : 0.0012185736458913317
speechiness : 0.02482497234444852
acousticness : 0.1508940885169172
instrumentalness : 0.2650231879735413
liveness : 0.0017610270658901823
valence : 0.06123975608998724
tempo : 0.001843649491325074
duration_ms : 0.04616232259887296
time_signature : 0.007187765634488112
chorus_hit : 0.0013198312574551716
sections : 0.032453915884986216
```

This confirms what I learned in the exploratory data analysis, with the danceability, energy, loudness, acousticness, and instrumentalness having the most effect on the model.

While 73% accuracy is a good score for a relatively untuned model, a fully tuned model should be able to increase this. I performed a 3 fold cross-validated grid search over the following parameters to further tune the random forest model:

```
 'bootstrap': [True],
'max_depth': [80, 90, 100, 110],
'max_features': [2, 3, 5],
'min_samples_leaf': [3, 4, 5],
'min_samples_split': [8, 10, 12],
'n_estimators': [100, 200, 300, 1000]
```

The resulting grid search returned this as the best estimator:

```
'bootstrap': True,
 'max_depth': 100,
 'max_features': 3,
 'min_samples_leaf': 3,
 'min_samples_split': 8,
 'n_estimators': 1000
```

This model produced a 78% accuracy score on the test data and an f1 score of 0.8. However, the accuracy score on the training data was 96%, indicating significant overfitting. To combat this, I reduced the max_depth to 90, max_features to 2 and min_samples_split to 5. This did not significantly reduce accuracy on test or training data, and with the training data scoring 18% higher than the test data, it appears that this model has very high variance.

My next approach was to try an XGBoost classifier model. I performed the same 3 fold cross-validated grid search over the following parameters:

```
  "eta" : [0.05, 0.10, 0.15, 0.20, 0.25, 0.30 ] ,
 "max_depth"        : [ 3, 4, 5, 6, 8, 10, 12, 15],
 "min_child_weight" : [ 1, 3, 5, 7 ],
 "gamma"            : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ],
 "colsample_bytree" : [ 0.3, 0.4, 0.5 , 0.7 ]
```

The result of the grid search was a classifier with the following parameters:

```
'colsample_bytree': 0.7,
 'eta': 0.05,
 'gamma': 0.3,
 'max_depth': 8,
 'min_child_weight': 5
```

This model achieved the same 78% accuracy score on the test set as the earlier random forest, with 85% accuracy on the training set, a marked improvement over the random forest’s 96% accuracy on training. The feature importances of this model are:

```
danceability : 0.10043808
energy : 0.0479939
key : 0.020840691
loudness : 0.04576259
mode : 0.07882381
speechiness : 0.05622998
acousticness : 0.11864295
instrumentalness : 0.25530356
liveness : 0.02467498
valence : 0.043647353
tempo : 0.027719742
duration_ms : 0.061845098
time_signature : 0.05478216
chorus_hit : 0.02297637
sections : 0.040318668
```

Interestingly, this model reduces the importance of energy and loudness, while danceability, acousticness and instrumentalness still appear to be the most important features for predicting hit potential.

I then applied some other accuracy metrics to the XGBoost model. It generated a ROC AUC score of 0.78, a precision score of 0.75, an F1 score of 0.8, and the following confusion matrix:

```
[4385, 1752],
[ 870, 5325]
```

It appears from the confusion matrix that this model is more likely to generate a false positive than a false negative, but overall the accuracy scores are looking fairly good, especially considering the data contains songs from a wide range of decades with changing musical tastes. As we saw in the visual analysis, the features of hit songs have changed from decade to decade. I decided to test this model by decade rather than on the entire dataset to see if this would increase accuracy.

To evaluate the model decade by decade, I re-split the data with the same 30% test size by decade, then trained the model on each decade's data and calculated the scores. The results were as follows:

1960s - 77% test, 92% train
1970s - 76% test, 94% train
1980s - 80% test, 94% train
1990s - 77% test, 92% train
2000s - 85% test, 95% train
2010s - 84% test, 95% train

While there is variance in the amount of accuracy by decade, the accuracy on the training data for all decades is at least 10% higher than the test accuracy in all cases. It also appears that it is easier to predict hits for songs from the 1980s, 2000s, and 2010s than other decades.

Due to the overfitting by decade, I decided to further tune the model by reducing colsample_bytree to 0.5 and max_depth to 3. On the full dataset with all decades included, this model achieved 77% accuracy on the test data, and 80% on the training data. While we lost .01% accuracy on the test set, there is significantly less overfitting in this model. I then tested it out on the 1990s dataset, which resulted in 76% test accuracy, and 80% training accuracy, again losing about 0.01 accuracy on the test data. I feel this is a good trade-off. This model also had an F1 score of 0.79, and a ROC AUC score of 0.77, again losing 0.01% compared to the original model.

Ultimately I ended up with a model that can predict a song's hit potential with about 77% accuracy on the entire dataset, and up to 84% on certain decades. Interesting next steps would be to determine why it performs better on some decades than others, and trying to further increase the accuracy.