Spotify Genre Classifier

By Harsha Malireddy

Problem

- Spotify has about 40-60,000 new songs added to the platform everyday which makes it very difficult for the platform to manually label each and every track that is uploaded.
- A solution to this is to use a machine learning genre classification model that can make predictions of the genre for each new track that is added.
- By adding this tag based genre classification feature across all platforms (web, computer, mobile) that use Spotify, Spotify could potentially convert free users of the platform into paying premium users as user satisfaction could increase due to the added feature.
- Free users might convert to premium users as they find Spotify to be the platform that offers the best and largest number of features to customize their listening experience.

Data

- **Kaggle Dataset**: https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db
- **232,725 tracks, 18 columns, 26 genres**
- **7 categorical**: genre, artist_name, track_name, track_id, key, mode, time_signature
- **11 numerical**: popularity, acousticness, danceability, duration_ms, energy, instrumentalness, liveness, loudness, speechiness, tempo, valence
- **Genre**: There are 26 genres of music
- **Artist_name**: Name of artist, **Track_name**: Name of track, **Track_id**: ID of track
- **Key**: ['D' 'C' 'F' 'B' 'E' 'G' 'G#' 'A#' 'C#' 'A' 'F#' 'D#']. The key or pitch the track is in
- **Mode**: [Major, Minor]. Modality of track, the type of scale from which its melodic content is derived.
- **Time_signature**: ['4/4' '3/4' '5/4' '1/4' '0/4']. 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).
- **Popularity**: Scale: [0, 100]. The popularity of the track with users on Spotify. 0 to 100 is an increase in popularity.
- **Acousticness**: Scale: [0.0, 1.0]. A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- **Danceability**: Scale: [0.0, 1.0]. 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.
- **Duration_ms**: The duration of the track in milliseconds.
- **Energy**: Scale: [0.0, 1.0]. It represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.
- **Instrumentalness**: Scale: [0.0, 1.0]. 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 isto 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.
- **Liveliness**: [0.0, 1.0]. 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.
- **Loudness**: Scale: [-60, 0]. The overall loudness of a track in decibels (dB).
- **Speechiness**: Scale: [0.0, 1.0]. 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.
- **Tempo**: The overall estimated tempo of a track in beats per minute (BPM).
- **Valence**: Scale: [0.0, 1.0]. Describes the musical positiveness conveyed by the track. High valence tracks sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	key
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.6110	0.389	99373	0.9100	0.000000	C#

liveness	loudness	mode	speechiness	tempo	time_signature	valence
0.3460	-1.828	Major	0.0525	166.969	4/4	0.8140

```
Genre:
 Comedy                 9681
Soundtrack              9646
Indie                   9543
Jazz                    9441
Pop                     9386
Electronic              9377
Children's Music        9353
Folk                    9299
Hip-Hop                 9295
Rock                    9272
Alternative             9263
Classical               9256
Rap                     9232
World                   9096
Soul                    9089
Blues                   9023
R&B                     8992
Anime                   8936
Reggaeton               8927
Ska                     8874
Reggae                  8771
Dance                   8701
Country                 8664
Opera                   8280
Movie                   7806
Children's Music        5403
A Capella                119
Name: genre, dtype: int64
```

```
genre                object
artist_name          object
track_name           object
track_id             object
popularity            int64
acousticness        float64
danceability        float64
duration_ms           int64
energy              float64
instrumentalness    float64
key                  object
liveness            float64
loudness            float64
mode                 object
speechiness         float64
tempo               float64
time_signature       object
valence             float64
dtype: object
```

Data Wrangling

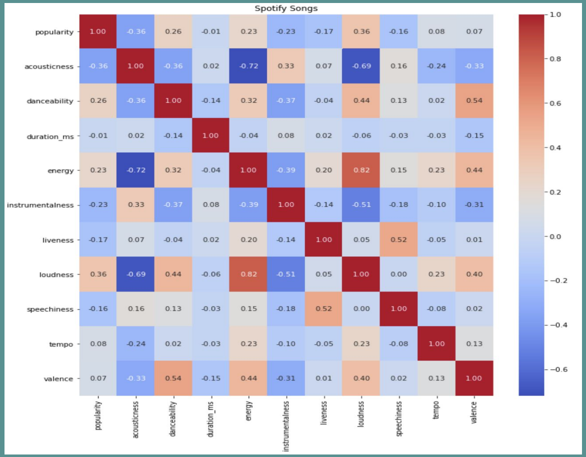
- No null values nor duplicate rows.
- I deleted the rows containing the ‘movie’ genre from the data set as I felt the label of ‘movie’ wasn’t a very useful or proper genre tag for which the model could classify on.
- I combined the genre labels of “Children’s Music” and “Children’s Music” since the apostrophes in the data file for each of the labels was different, leading them to seen as different labels. I gave one specific apostrophe character for all of the ‘Children’s Music’ labels to solve this issue.
- I converted 3 categorical columns (mode, key, time_signature) to numerical values by assigning integer values to each of the unique categorical values and then merged the numerical conversions into the dataset dataframe as additional columns.
- Next, I eliminated any rows that classified the same song into more than one genre since I’m treating this as a multi classification problem and not as a multi-label classification problem. This elimination left only one row for each unique song that classifies it into one genre.
    - To do this, first I changed the artist name and track name columns to uppercase strings so no two same strings with different capitalizations would be treated as different tracks.
    - Then, I used the pandas ‘drop_duplicates()’ function with ‘track_id’ to find and drop rows with the same track id.
    - Lastly, I used ‘drop_duplicates()’ again but with the subset [‘artist_name’, ‘track_name]. This would identify identical songs that have the different ‘track_id’ values for reasons such as appearing in different albums, among others. These songs could be potentially classified into different genres and thus this final filtering ensures each unique track has one row and genre associated with it.
    - The result of all the filtering created a final data set of 152,864 tracks from the original 232,725 tracks.

```
Genre:
 ['R&B' 'A Capella' 'Alternative' 'Country' 'Dance' 'Electronic' 'Anime'
 'Folk' 'Blues' 'Opera' 'Hip-Hop' 'Children’s Music' 'Rap' 'Indie'
 'Classical' 'Pop' 'Reggae' 'Reggaeton' 'Jazz' 'Rock' 'Ska' 'Comedy'
 'Soul' 'Soundtrack' 'World']

Key:
 ['D' 'C' 'F' 'B' 'E' 'G' 'G#' 'A#' 'C#' 'A' 'F#' 'D#']

Time Signature:
 ['4/4' '3/4' '5/4' '1/4' '0/4']

Mode:
 ['Minor' 'Major']
```

key_num	mode_num	time_signature_num
2	0	4
0	0	3
5	0	4

EDA

- I plotted a seaborn correlation heatmap to understand the relationship between the 11 numerical attributes of a track.
- I found that energy and loudness had a strong correlation.



- I created a table to show the summary statistics of each of the attributes of a track.

	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
count	224919.000000	224919.000000	224919.000000	2.249190e+05	224919.000000	224919.000000	224919.000000	224919.000000	224919.000000	224919.000000	224919.000000
mean	42.132354	0.357150	0.556557	2.359802e+05	0.577908	0.149095	0.214534	-9.452503	0.121159	117.795684	0.455164
std	17.487794	0.351677	0.185452	1.142341e+05	0.261571	0.303435	0.198291	5.976156	0.185723	30.909935	0.259384
min	0.000000	0.000000	0.056900	1.538700e+04	0.000020	0.000000	0.009670	-52.457000	0.022200	30.379000	0.000000
25%	30.000000	0.034700	0.439000	1.844780e+05	0.399000	0.000000	0.097100	-11.530000	0.036800	92.992000	0.239000
50%	44.000000	0.216000	0.573000	2.213870e+05	0.614000	0.000047	0.128000	-7.643000	0.050400	115.970000	0.445000
75%	55.000000	0.697000	0.694000	2.665470e+05	0.791000	0.037700	0.263000	-5.450000	0.106000	139.479000	0.660000
max	100.000000	0.996000	0.989000	5.552917e+06	0.999000	0.999000	1.000000	3.744000	0.967000	242.903000	1.000000

EDA

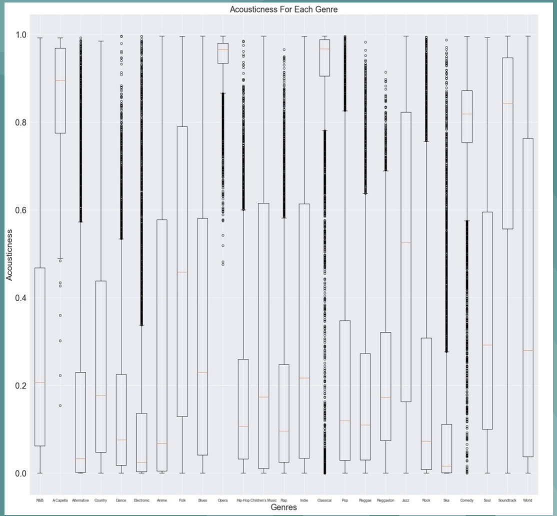
- I created a table to show the summary statistics grouped by genre of each attribute of a track.

	popularity								acousticness		...	tempo		valence				
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max	count	mean	std	min	25%
genre																		
A Capella	119.0	9.302521	7.868145	0.0	4.0	8.0	13.0	44.0	119.0	0.829941	...	129.68300	181.714	119.0	0.328724	0.255005	0.0380	0.1
Alternative	9263.0	50.213430	7.661040	0.0	45.0	49.0	55.0	83.0	9263.0	0.162313	...	143.96550	213.788	9263.0	0.449590	0.216426	0.0321	0.2
Anime	8936.0	24.258729	9.648703	0.0	17.0	23.0	30.0	65.0	8936.0	0.286843	...	149.99125	220.276	8936.0	0.441682	0.249619	0.0000	0.2
Blues	9023.0	34.742879	9.755328	0.0	28.0	33.0	40.0	80.0	9023.0	0.327840	...	140.63550	242.903	9023.0	0.579425	0.224677	0.0315	0.4
Children's Music	14756.0	36.202426	25.536529	0.0	2.0	49.0	56.0	86.0	14756.0	0.320112	...	140.09250	220.119	14756.0	0.532251	0.250929	0.0000	0.3
Classical	9256.0	29.282195	14.133541	0.0	25.0	32.0	38.0	69.0	9256.0	0.868843	...	127.18125	212.923	9256.0	0.214463	0.200275	0.0000	0.0
Comedy	9681.0	21.342630	8.428764	0.0	15.0	20.0	26.0	61.0	9681.0	0.793098	...	115.12800	207.157	9681.0	0.412764	0.207258	0.0237	0.2
Country	8664.0	46.100416	9.745975	0.0	39.0	45.0	52.0	82.0	8664.0	0.270172	...	144.48075	217.538	8664.0	0.535160	0.219819	0.0395	0.3
Dance	8701.0	57.275256	11.208370	0.0	51.0	57.0	64.0	100.0	8701.0	0.152888	...	134.03000	218.081	8701.0	0.517754	0.226822	0.0340	0.3
Electronic	9377.0	38.056095	9.741981	0.0	31.0	37.0	44.0	96.0	9377.0	0.119839	...	144.99000	220.169	9377.0	0.388129	0.236938	0.0205	0.1
Folk	9299.0	49.940209	8.218284	0.0	44.0	49.0	55.0	84.0	9299.0	0.463201	...	136.30850	236.799	9299.0	0.440237	0.241416	0.0277	0.2
Hip-Hop	9295.0	58.423131	8.269761	14.0	52.0	57.0	63.0	98.0	9295.0	0.176172	...	141.98650	214.126	9295.0	0.473381	0.222325	0.0336	0.3
Indie	9543.0	54.701561	7.355754	0.0	49.0	54.0	59.0	97.0	9543.0	0.331214	...	137.93400	219.331	9543.0	0.428665	0.221606	0.0277	0.2
Jazz	9441.0	40.824383	9.588840	0.0	35.0	40.0	46.0	79.0	9441.0	0.499606	...	127.87100	239.848	9441.0	0.508961	0.251218	0.0266	0.3
Opera	8280.0	13.335628	8.460264	0.0	7.0	11.0	17.0	63.0	8280.0	0.945202	...	120.66900	236.735	8280.0	0.189864	0.172322	0.0207	0.0
Pop	9386.0	66.590667	7.248797	3.0	62.0	66.0	71.0	100.0	9386.0	0.224819	...	140.01575	213.990	9386.0	0.481371	0.225029	0.0277	0.3
R&B	8992.0	52.308719	9.246359	0.0	46.0	51.0	58.0	92.0	8992.0	0.288216	...	134.17175	216.636	8992.0	0.450346	0.215387	0.0321	0.2
Rap	9232.0	60.533795	8.177226	14.0	55.0	59.0	65.0	99.0	9232.0	0.168080	...	142.00050	216.115	9232.0	0.455918	0.213913	0.0331	0.2
Reggae	8771.0	35.589328	10.779762	0.0	28.0	34.0	42.0	78.0	8771.0	0.185783	...	142.31400	218.184	8771.0	0.679665	0.198141	0.0331	0.5
Reggaeton	8927.0	37.742915	13.544414	0.0	28.0	35.0	46.0	98.0	8927.0	0.218923	...	146.03850	234.923	8927.0	0.659439	0.202052	0.0381	0.5
Rock	9272.0	59.619392	7.474083	0.0	54.0	59.0	64.0	95.0	9272.0	0.196429	...	142.00875	219.331	9272.0	0.517113	0.231137	0.0277	0.3
Ska	8874.0	28.612351	10.757129	5.0	21.0	27.0	35.0	74.0	8874.0	0.099728	...	155.57725	221.578	8874.0	0.653472	0.223245	0.0331	0.4
Soul	9089.0	47.027836	9.235035	0.0	41.0	46.0	53.0	85.0	9089.0	0.360679	...	132.44500	216.636	9089.0	0.480562	0.245857	0.0287	0.2
Soundtrack	9646.0	33.954800	8.639645	0.0	28.0	33.0	39.0	72.0	9646.0	0.717349	...	126.14200	216.429	9646.0	0.118483	0.139913	0.0000	0.0
World	9096.0	35.524077	9.399816	0.0	29.0	34.0	41.0	76.0	9096.0	0.393341	...	141.55675	212.923	9096.0	0.295657	0.230914	0.0000	0.1

- I displayed the summary statistics for each of the 9 important numerical features of a track for each genre: 'acousticness', 'danceability', 'energy', 'instrumentalness', 'liveness', 'loudness', 'speechiness', 'tempo', and 'valence.'

	count	mean	std	min	25%	50%	75%	max
genre								
A Capella	119.0	0.829941	0.181866	0.154000	0.775500	0.89500	0.96900	0.992
Alternative	9263.0	0.162313	0.241155	0.000001	0.001950	0.03350	0.23050	0.992
Anime	8936.0	0.286843	0.362341	0.000000	0.005050	0.06795	0.57725	0.996
Blues	9023.0	0.327840	0.309981	0.000001	0.041100	0.22900	0.58100	0.996
Children's Music	14756.0	0.320112	0.339331	0.000001	0.011200	0.17350	0.61525	0.996
Classical	9256.0	0.868843	0.255269	0.000001	0.905000	0.96700	0.98800	0.996
Comedy	9681.0	0.793098	0.130313	0.000363	0.753000	0.81900	0.87200	0.995
Country	8664.0	0.270172	0.262801	0.000028	0.048000	0.17700	0.43800	0.985
Dance	8701.0	0.152888	0.184252	0.000004	0.018500	0.07600	0.22500	0.996
Electronic	9377.0	0.119839	0.200477	0.000002	0.003610	0.02460	0.13700	0.995
Folk	9299.0	0.463201	0.334784	0.000001	0.129000	0.45800	0.79000	0.995
Hip-Hop	9295.0	0.176172	0.188891	0.000015	0.033000	0.10700	0.26000	0.985
Indie	9543.0	0.331214	0.321618	0.000001	0.034100	0.21700	0.61400	0.995
Jazz	9441.0	0.499606	0.337637	0.000001	0.163000	0.52500	0.82300	0.996
Opera	8280.0	0.945202	0.057516	0.476000	0.934000	0.96500	0.98000	0.996
Pop	9386.0	0.224819	0.250306	0.000006	0.029200	0.12000	0.34800	0.995
R&B	8992.0	0.288216	0.262520	0.000030	0.062075	0.20700	0.46800	0.992
Rap	9232.0	0.168080	0.189780	0.000007	0.025700	0.09600	0.24825	0.965
Reggae	8771.0	0.185783	0.204736	0.000004	0.030000	0.11000	0.27300	0.982
Reggaeton	8927.0	0.218923	0.180025	0.000010	0.074850	0.17300	0.32100	0.914
Rock	9272.0	0.196429	0.252861	0.000001	0.008770	0.07310	0.30825	0.994
Ska	8874.0	0.099728	0.174256	0.000001	0.001550	0.01625	0.11200	0.987
Soul	9089.0	0.360679	0.288262	0.000014	0.100000	0.29200	0.59500	0.993
Soundtrack	9646.0	0.717349	0.292065	0.000004	0.557000	0.84250	0.94700	0.996
World	9096.0	0.393341	0.363125	0.000002	0.037300	0.28000	0.76300	0.996

- I plotted box plots grouped genre for each of the 9 important numerical features of a track for each genre: 'acousticness', 'danceability', 'energy', 'instrumentalness', 'liveness', 'loudness', 'speechiness', 'tempo', and 'valence.'
- he data showed many outliers when grouped by each genre as seen in the box plots.



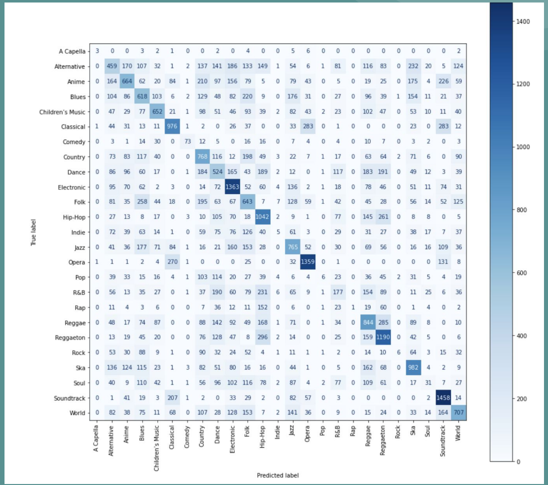
Modeling

- I set the features that would be used to train the model to be the columns: ['acousticness', 'danceability', 'energy', 'instrumentalness', 'key_num', 'liveness', 'loudness', 'mode_num', 'speechiness', 'tempo', 'time_signature_num', 'valence'].
- I removed outliers by finding the upper and lower bounds by (mean + cutoff (3 * std)) and (mean - cutoff (3 * std)) respectively.
- I used the StandardScaler() method to fit and transform my training set and used the training set scaler on the testing set to avoid data leakage.
- I trained and tested on 4 ML models: logistic regression (multinomial), Random Forest Classifier, Gradient Boosting Classifier, and Linear Support Vector Classifier.
- I took the following 5 steps to implement each of the 4 ML models:
  - 1. I used GridSearchCV for hyperparameter tuning to pick the best model version
  - 2. I fit the model and madee predictions for the test set.
  - 3. Printed the classification report displaying the accuracy, precision, recall, and F1-scores for each unique class variable ('genre').
  - 4. Calculated the accuracy, precision, recall, F1-score, and log loss scores, and ROC-AUC (area under the curve) for the model.
  - 5. Lastly, displayed the confusion matrix to see the distribution of predictions being made across all genres.

Classification Report

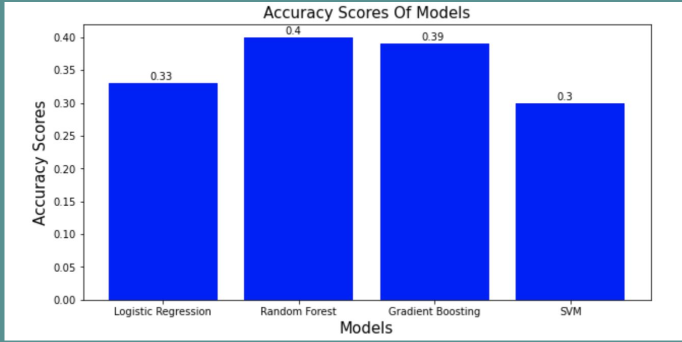
	precision	recall	f1-score	support
A Capella	0.60	0.11	0.18	28
Alternative	0.26	0.20	0.23	2240
Anime	0.39	0.31	0.34	2177
Blues	0.28	0.31	0.29	2000
Children's Music	0.50	0.42	0.45	1569
Classical	0.56	0.55	0.55	1777
Comedy	0.78	0.35	0.48	210
Country	0.31	0.43	0.36	1802
Dance	0.25	0.27	0.25	1974
Electronic	0.45	0.61	0.52	2245
Folk	0.27	0.33	0.30	1968
Hip-Hop	0.40	0.57	0.47	1829
Indie	0.11	0.01	0.01	820
Jazz	0.36	0.39	0.38	1937
Opera	0.68	0.74	0.71	1836
Pop	0.32	0.01	0.02	596
R&B	0.21	0.14	0.16	1308
Rap	1.00	0.00	0.01	359
Reggae	0.34	0.40	0.37	2108
Reggaeton	0.44	0.57	0.50	2095
Rock	0.55	0.01	0.02	543
Ska	0.45	0.51	0.48	1924
Soul	0.13	0.03	0.05	1075
Soundtrack	0.57	0.75	0.64	1954
World	0.47	0.38	0.42	1842
accuracy			0.40	38216
macro avg	0.43	0.33	0.33	38216
weighted avg	0.39	0.40	0.38	38216

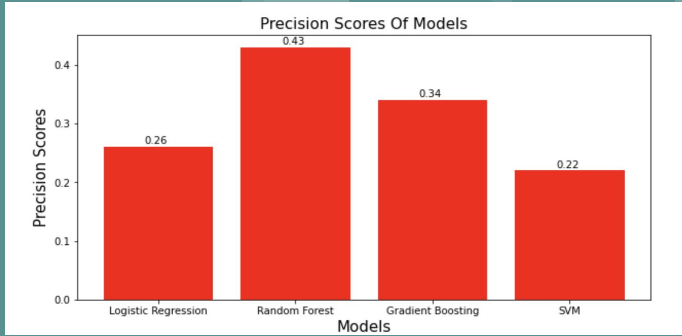
Confusion Matrix

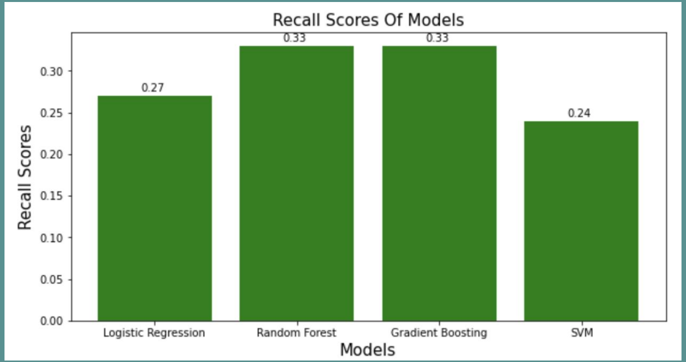


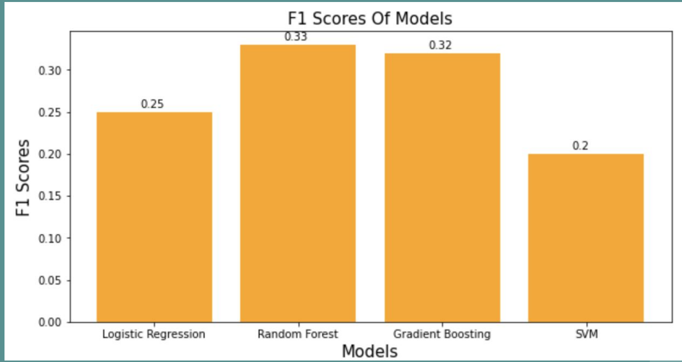
Models Comparison

- The accuracy, precision, recall, and F1 scores are compared for all 4 models.
- The delta between the best and worst model in accuracy: 1%, precision: 9%, recall: 9%, f1: 13%



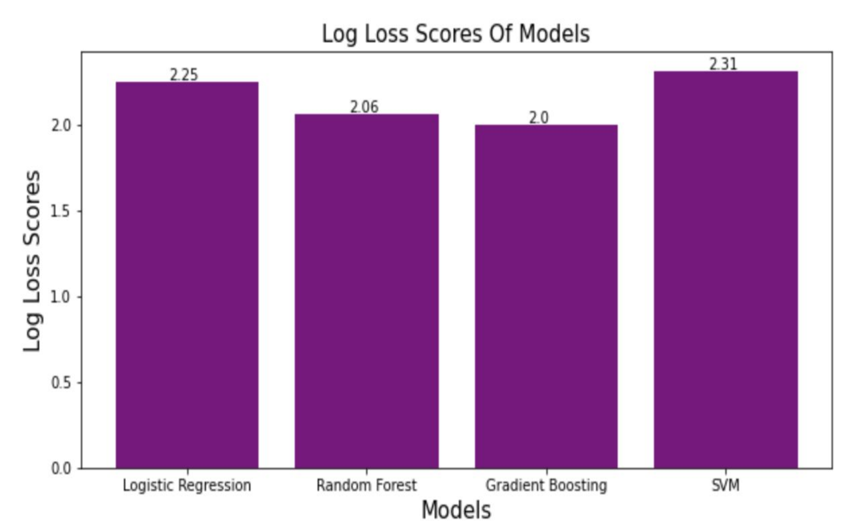


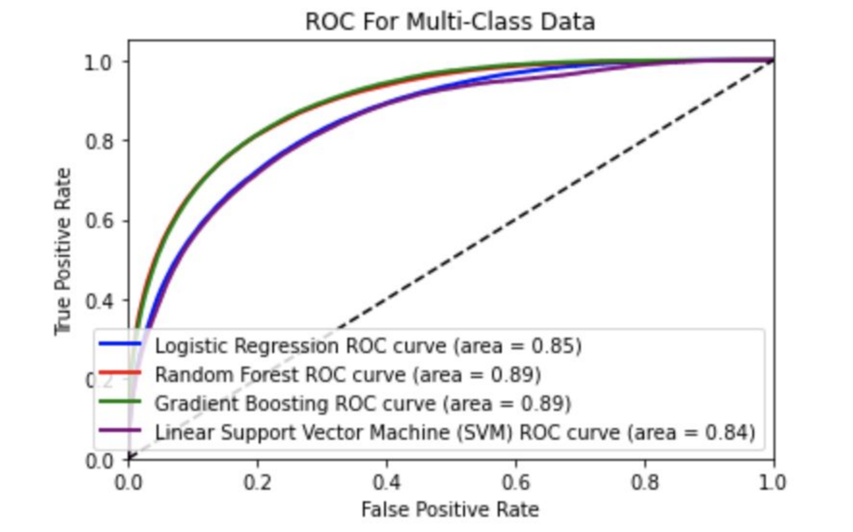




Models Comparison

- The lowest Log Loss score was 2.0 for the Gradient Boosting Classifier model but the Random Forest Classifier model was close with a negligible 0.06 delta at 2.06.
- The ROC curves and the area under the curve (AUC) is the largest at 0.89 for both the gradient boosting and random forest classifiers.





Genre Prediction & Similar Tracks Recommender Functions

- I created a function that classifies song inputs from users into their predicted genre label and the next 2 most likely genres the song can be classified into in case Spotify decides to classify songs into more than one genre based on likelihood.
- To accomplish this I built methods called 'get_token(),' 'search_for_track(),' 'get_features(),' and 'predict_genres()' to use the API and get the required audio features in cases where tracks inputted by a user don't exist in the dataset.
- Then, I used the random forest model to predict on the track and get the genre it predicts is the classification.
- Lastly, I use the 'predict_proba()' method to get the likelihood of classification into the different genres and display the top 2 likelist ones after the actual predicted genre label.

```
In [211]: predict_genres()

Please enter track name to find its genre:
sad
Please enter artist name of song to find its genre:
bo burnham

The genre the song can be classified into is: Blues at 13.80% probaility.
The next 2 likeliest genres the song can be classified into are: Jazz at 10.2
0% probability and Children's Music at 9.00% probability.
```

- I created a function that recommends the top K (user inputted integer K) songs similar to the one the user inputs that the user might want to listen to.
- I asked for the inputs of artist name and track name as well as the k number of songs they want recommended to them.
- To accomplish this I mimicked the KNN (K-Nearest Neighbor) algorithm by finding the euclidean distance of each of the inputted track's features with every other songs features in the dataset and recommending the top k songs in order of smallest euclidean distance

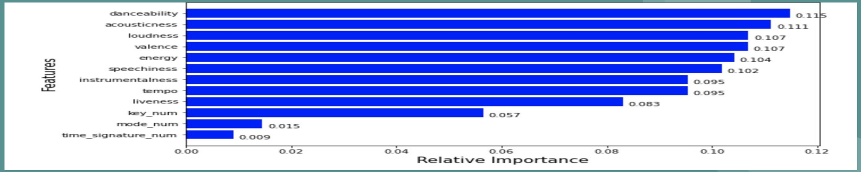
```
In [24]: recommend_songs()

Please enter the number of similar songs you would like to get recommended be
tween 1 and 152864:
10
Please enter track name to find its 10 closest similar songs:
bambi
Please enter artist name of song to find its 10 closest similar songs:
baekhyun

Your top 10 song recommendations are:

1. TRIO FOR HORN, VIOLIN AND PIANO IN EB, OP. 40; I by TOM'S MUSIC BOX
2. LACRIMOSA DOMINAE by IMMEDIATE
3. テルーの唄(ゲド戦記より) by YUKA
4. RIDE TO THE NAZI HIDEOUT by JOHN WILLIAMS
5. JESUS ON THE MOUNT by MATT BRAUNGER
6. INTRODUCTION - LIFE OF BRIAN / SOUNDTRACK VERSION by MONTY PYTHON
7. JACQUARD CAUSEWAY by BOARDS OF CANADA
8. EVERYTHING'LL CHANGE by MICHL
9. LEON WITH CLAIRE by CAPCOM SOUND TEAM
10. CAT RESTAURANT by BECKY DONOHUE
```

Takeaways

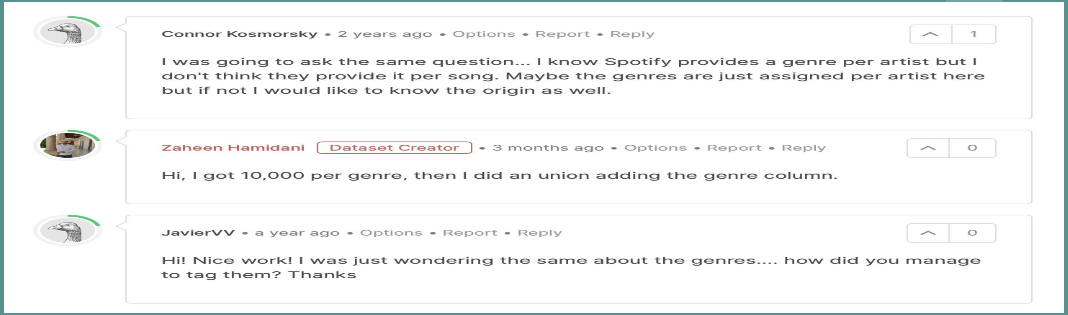
- The random forest classifier was the best model although the gradient boosting classifier was also very close in performance with a 0.06 lower log loss score but was 7% less in precision than the random forest classifier.

Model	Hyperperameters	Accuracy	Precision	Recall	F1-Score	Log Loss	ROC-AUC
Random Forest Classifier	n_estimators = 500, criterion = gini	0.4	0.43	0.33	0.33	2.06	0.89

- Based on the displayed feature importances horizontal bar graph, the three categorical variables seem to not be very useful in training the model.



- There is a possibility that the tracks have not been accurately labeled as this dataset was compiled by a kaggle user who didn't explain the method by which they were able to label all the tracks.
- 'Garbage in, garbage out' could possibly be an accurate saying to represent the reason for the lack of an accurately created model.
- Further experimentation and testing would be necessary to come to this conclusion of whether the data is intrinsically problematic and needs further feature engineering or the data compilation by the creator of the dataset is the issue.



Connor Kosmorsky • 2 years ago • Options • Report • Reply 1

I was going to ask the same question... I know Spotify provides a genre per artist but I don't think they provide it per song. Maybe the genres are just assigned per artist here but if not I would like to know the origin as well.

Zaheen Hamidani Dataset Creator • 3 months ago • Options • Report • Reply 0

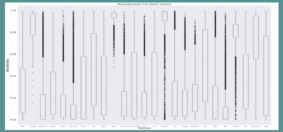
Hi, I got 10,000 per genre, then I did an union adding the genre column.

JavierVV • a year ago • Options • Report • Reply 0

Hi! Nice work! I was just wondering the same about the genres.... how did you manage to tag them? Thanks

Further Research

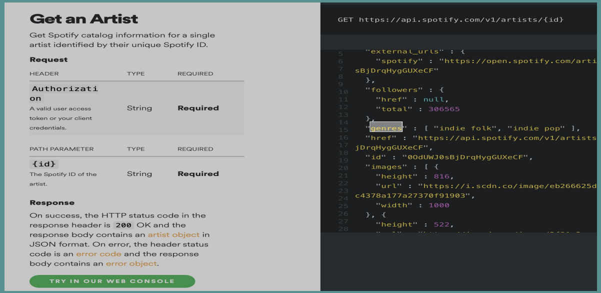
- In the future, I would like to eliminate outliers for each genre grouped by its associated features and see the results of refitting the model with the filtered data. As seen in the EDA step there were many outliers that could potentially have significantly hurt the fit of the random forest classifier.

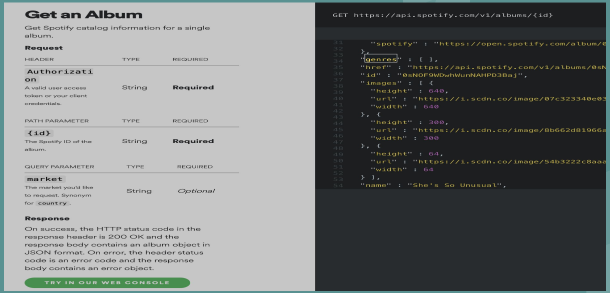


- Another thing to try would be to try and use different binning of features to see if that could help differentiate the genres more from one another.
- This binning could be done with grouping the categorical variables such as keys (ex. C and C#, etc.) or even certain ranges of numerical features could be taken together to represent certain integer values.
- Trying different binning techniques would allow us to see what leads to more distinct stratification of data between genres.

```
Key:
 ['D' 'C' 'F' 'B' 'E' 'G' 'G#' 'A#' 'C#' 'A' 'F#' 'D#']
```

- The third thing to try would be to compile the dataset myself by accessing the API and getting genres based on artist and album and synthesizing them in a way that would lead to more accurate pre-labeled data to use to train the model.
- I already tried to look at genres based on tracks which don't exist, and I similarly couldn’t find genre labels for albums as well.
- Maybe the genre labels for albums are rare in the Spotify database but there seems to be some genre labels for artists. Although using artist overall genre labels might seem improper when constructing a dataset, only further testing will determine how accurate it could be.





Thank You