Predicting Billboard Hit Songs

Thomas DeNezza, Syed Sabeeh Hassany, Jackson Winslow, Jason Wu

Introduction

Approach

1. **Data generation**
    a. Curating our own dataset by combining information from Spotify and Billboard API's
2. **Model creation and augmentation**
    a. Training base models on a binary classification, hypertuning these models to achieve best prediction accuracy
3. **Analysis**
    a. Analyzing our results in the context of each of our evaluations

Approach

1. Evaluating our ability to **predict whether a song was a hit or not** individually across the four genres as well as all together (and which models perform best)

2. Evaluating **which features are most influential** in a song becoming a hit for each genre

Data Sourcing

Spotify Api

Simple wrapper for the spotify api

Unnamed: 0	name	artist	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	hit
0	KILL BILL	SZA	0.644	0.735	8	-5.747	1	0.0391	0.0521	0.144000	0.161	0.418	88.980	153947	0
1	ANTI-HERO	TAYLOR SWIFT	0.637	0.643	4	-6.571	1	0.0519	0.1300	0.000002	0.142	0.533	97.008	200690	1
2	CALM DOWN	REMA	0.801	0.806	11	-5.206	1	0.0381	0.3820	0.000669	0.114	0.802	106.999	239318	1
3	ROMANTIC HOMICIDE	D4VD	0.571	0.544	6	-10.613	1	0.0299	0.4530	0.008050	0.322	0.216	132.052	132631	0
4	SNOOZE	SZA	0.559	0.551	5	-7.231	1	0.1320	0.1410	0.000000	0.110	0.392	143.008	201800	0

Rap: 22.69% hits, 77.31% non-hits

Jazz: 3.21% hits, 96.79% non-hits

Pop: 16.17% hits, 83.83% non-hits

Country: 23.51% hits, 76.47% non-hits

Mixed: 16.39% hits, 83.61% non-hits

Models

Logistic Regression

K-Nearest Neighbors

Decision Tree

Random Forest

Support Vector Machine (Linear Kernel)

Support Vector Machine (RBF Kernel)

Gradient Boost

AdaBoost (uses Random Forest as base estimator)

Neural Network

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(32, 128)	1664
dense_5 (Dense)	(32, 256)	33024
dense_6 (Dense)	(32, 256)	65792
dense_7 (Dense)	(32, 1)	257

Total params: 100,737
Trainable params: 100,737
Non-trainable params: 0

Hyperparameter Tuning

```python
from sklearn.model_selection import GridSearchCV

# Define the hyperparameter values to search over
param_grid = {
    'max_depth': [3, 5, 7, 11, 15],
    'n_estimators': [50, 100, 200],
}

# Create a random forest classifier object
rfc = RandomForestClassifier()

# Create a GridSearchCV object
grid_search = GridSearchCV(estimator=rfc, param_grid=param_grid, cv=5, n_jobs=-1)

# Fit the GridSearchCV object to the data
grid_search.fit(X_train, y_train)

# Print the best hyperparameters
print("Best hyperparameters: ", grid_search.best_params_)

# Print the best score
print("Best score: ", grid_search.best_score_)
```

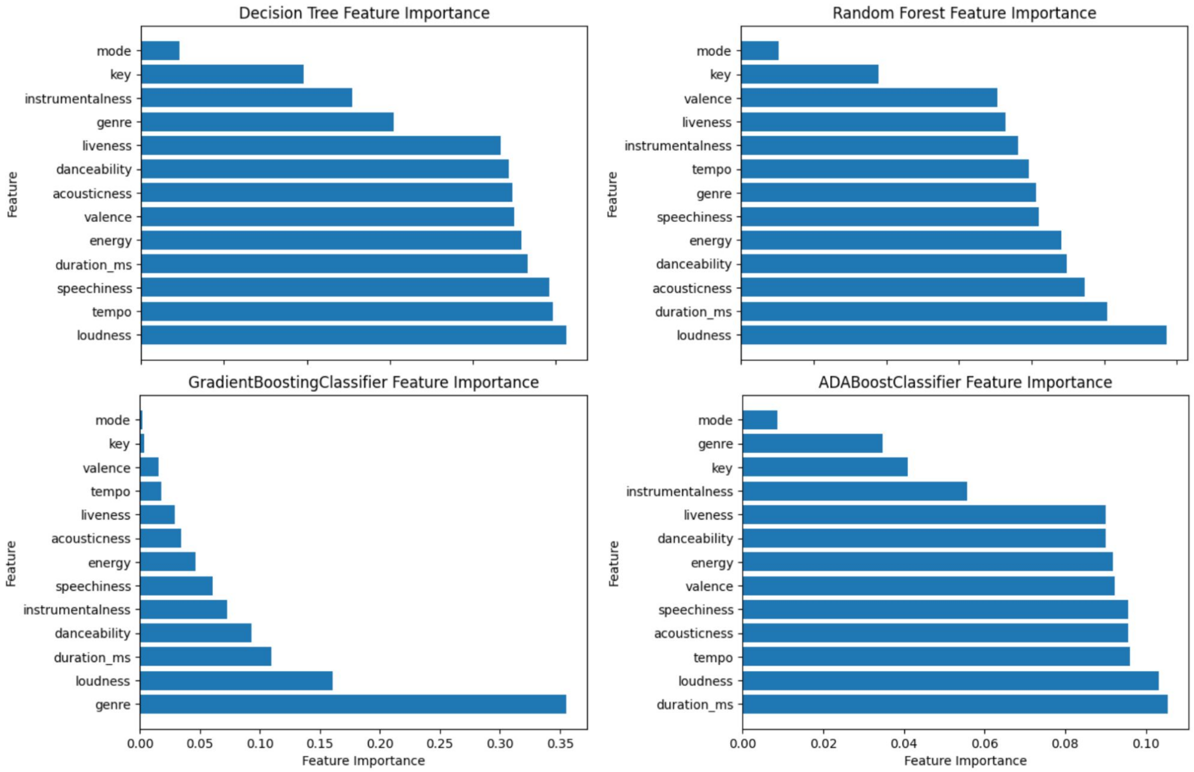
```
Best hyperparameters:  {'max_depth': 15, 'n_estimators': 100}
Best score:  0.8609285714285713
```

```python
acc=0
for n_estimators in [100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600,
  for learning_rate in [.1,.2,.3,.4,.5,.6,.7,.8,.9,1]:
    for max_depth in [1,2,3,4,5,6,7,8,9,None]:
      print(f'Training {n_estimators}, {learning_rate}, {max_depth}')
      gbmodel = GradientBoostingClassifier(n_estimators=n_estimators, learn

      gb_prediction=gbmodel.predict(test_X)
      acc_temp=sklearn.metrics.accuracy_score(test_Y,gb_prediction)
      if(acc_temp>acc):
        acc=acc_temp
        print([n_estimators,learning_rate,max_depth,acc])

# Best [200, 0.5, 10, 0.9156666666666666]
```

Classic Model Analysis

Model \ Genre	Rap	Pop	Jazz	Country	Mix
Logistic Regression	78.03%	84.17%	96.57%	76.63%	83.49%
K-Nearest Neighbors	77.33%	83.67%	96.20%	75.97%	82.45%
Decision Tree	86.43%	89.27%	97.03%	85.53%	89.40%
Random Forest	90.83%	90.23%	97.53%	76.93%	86.13%
Support Vector Machine (Linear Kernel)	78.50%	84.17%	96.60%	76.77%	83.55%
Support Vector Machine (Linear Kernel)	78.80%	84.23%	96.60%	76.77%	83.56%
GradientBoostingClassifier	90.50%	84.43%	97.10%	77.43%	83.65%
ADABoostClassifier	91.43%	93.50%	97.50%	90.17%	91.49%

Feature Importance Analysis



Neural Network Analysis

Genre	Accuracy
Rap	90.65%
Pop	90.70%
Jazz	96.40%
Country	87.35%
All Genres	90.19%
All Genres (with genre as a feature)	91.74%

Future Actions

- An **analysis of changes** in time period
- **Number of artist** followers as a parameter
- Analyze **different genres**
- Create a **balanced labeled dataset**

Conclusion

- Models failed to predict hit songs **effectively**
- More **balanced dataset** recommended for better results
- **Tuning focused** on accuracy, not F1 score
- **Lessons learned** about building quality datasets and model