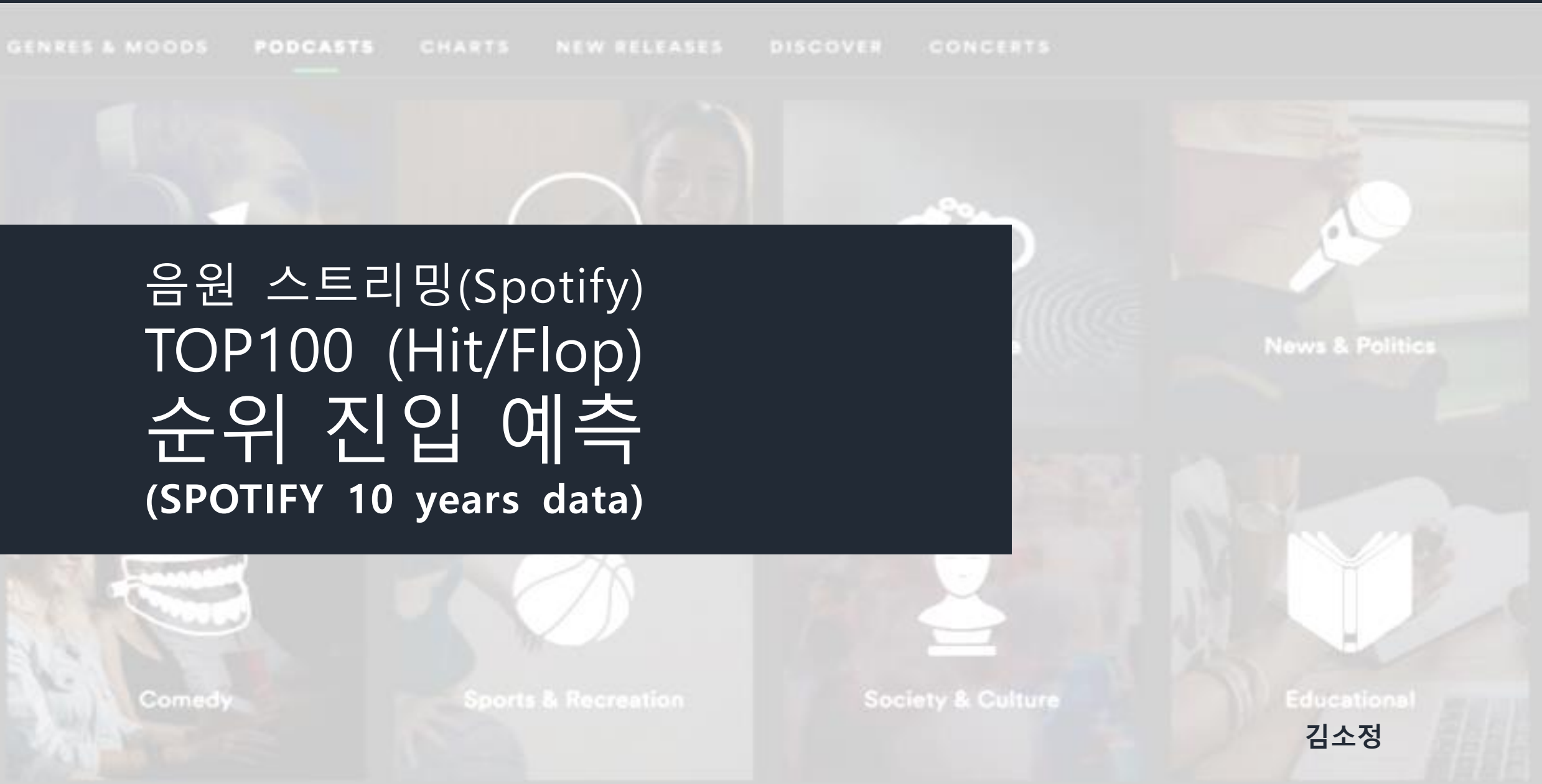


음원 스트리밍(Spotify) TOP100 (Hit/Flop) 순위 진입 예측

(SPOTIFY 10 years data)

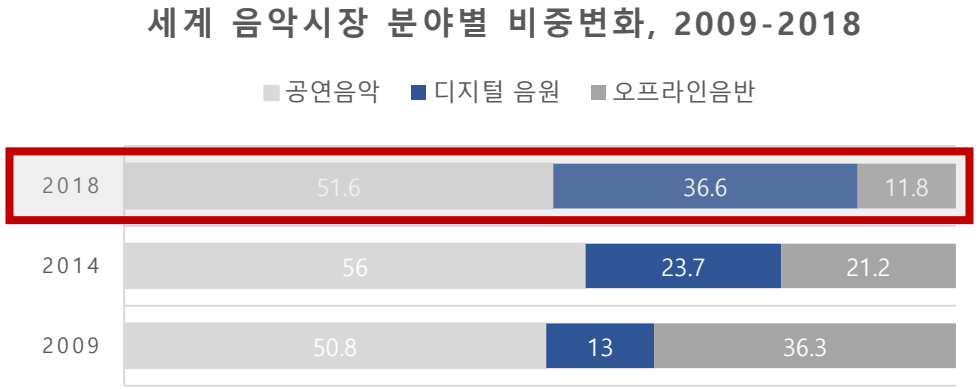
김소정

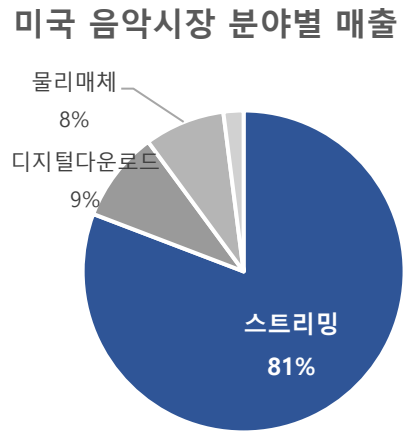
CONTENT

주제선정이유

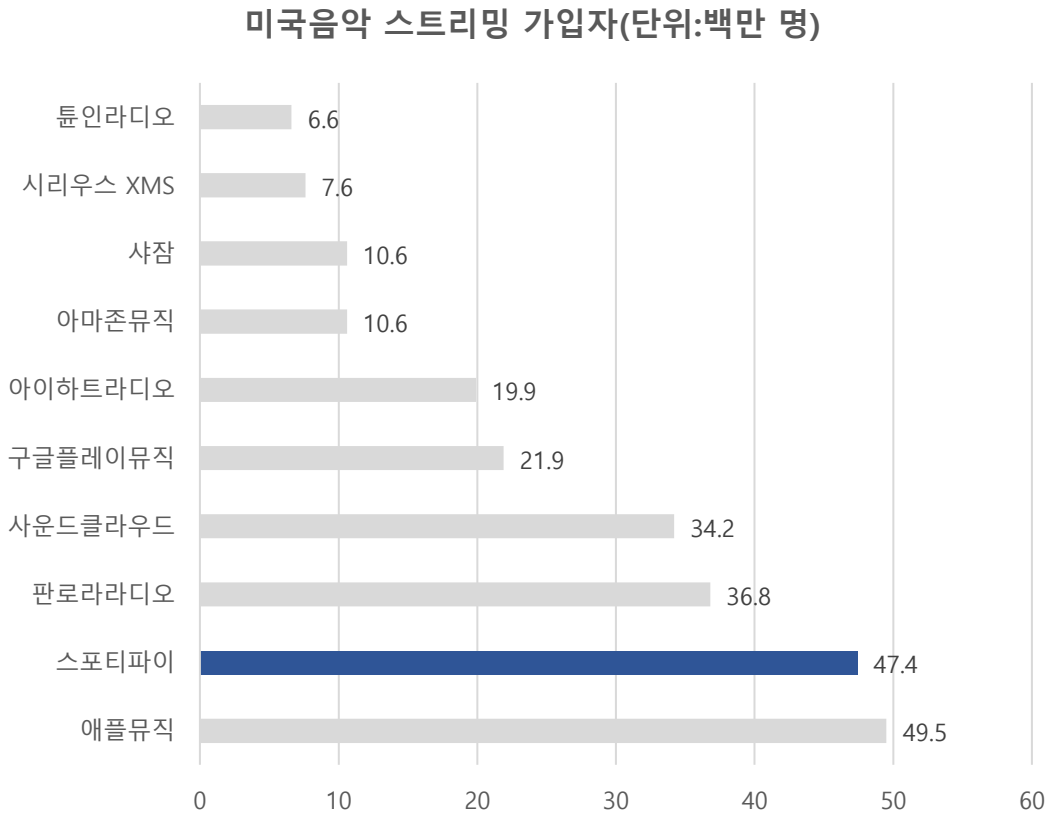
디지털 음악시장 중에서도 스트리밍 방식으로 음악을 소비하는 트렌드로 변화하고 있음

- 미국 시장 조사업체인 프라이스 워터하우스 쿠퍼스(PwC)가 발표한 '2019~2023 글로벌 미디어 엔터테인먼트 전망'에 의하면, 미국 음악시장은 디지털 음악 스트리밍 분야를 중심으로 음악시장의 성장을 견인하고 있으며 전체 음악시장 규모는 2018년 203억달러를 기록하고 있음
- 미국 유료 음악 스트리밍 서비스 시장은 애플뮤직과 스포티파이가 1,2위를 두고 경쟁하고 있음





출처: PwC(2019)
출처: RIAA(2019)

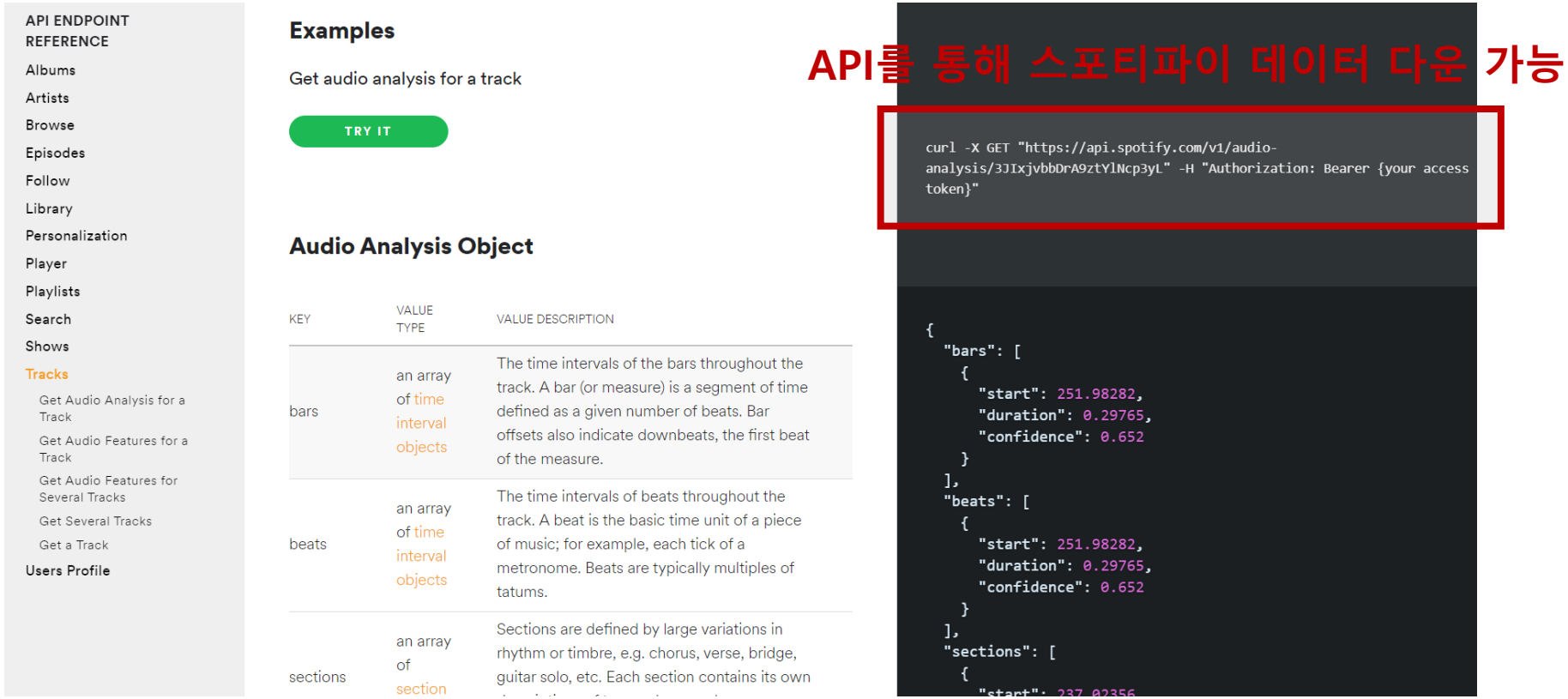


출처: Statista(2018),Most popular music streaming services in the U.S. 2018

데이터 설명

음원스트리밍 서비스가 확대된 2010년부터 2019년까지 스포티파이에서 제공하는 연도별 음원 순위 차트 데이터를 분석하여 TOP 100에 올라간 공통요인을 분석하고 Hit와 Flop을 예측하고자 함

- 기존 분석들은 스포티파이 데이터를 이용하여 음악 장르 분류를 통해 Hit한 장르 트렌드를 분석이 많음
- 기존 사례를 기반으로 Tree를 이용해 Hit한 음원들의 요인변수를 알아보고 Hit와 Flop을 예측하고자 함



데이터 설명 - 변수

19개 변수 사용
데이터 출처: Spotify(https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/)

변수명	설명
Track	음원이름
Artist	가수이름
Uri	Spotify URI (Uniform Resource Indicator)는 Spotify의 모든 트랙, 앨범 또는 아티스트 프로필의 공유 메뉴에서 찾을 수있는 링크
Danceability	템포, 리듬 안정성, 비트 강도, 전반적인 규칙 성을 포함한 음악적 요소의 조합을 기반으로 트랙이 춤에 얼마나 적합한 지 설명
Energy	0에서 1까지의 측정 값이며 강도와 활동의 지각적 측정(일반적으로 활기찬 트랙은 빠르고 시끄러울수록 수치가 1에 가까움)
Key	트랙의 전체 예상 키, 키가 감지되지 않은 경우 값은 -1
Loudness	전체 트랙의 상대적인 데시벨 (dB), 값은 일반적으로 -60 ~ 0db 사이
Mode	콘텐츠가 파생되는 유형, Major는 1 minor는 0 표시
Speechiness	트랙에서 단어를 감지, 범위는 0.0 (비 음성)에서 1.0 (음성과 유사)
Acousticness	트랙이 음향인지 여부에 대한 측정, 음향이 높을수록 1.0
Instrumentalness	트랙에 보컬이 없는지 여부, 1에 가까울수록, 0.5 이상의 값은 악기 트랙
Liveness	값이 높을수록 트랙이 라이브로 수행
Valence	트랙이 전달하는 음악적 긍정성, 높은 점수는 행복함, 쾌활함, 행복감 등을 의미, 낮은 점수의 트랙은 슬프고 우울함 등을 의미
Tempo	트랙의 전체 예상 템포 (BPM)
Duration	밀리 초 단위의 트랙 길이.
Time_signature	트랙의 전체 예상 박자표. 박자 기호 (미터)는 각 마디 (또는 마디)에 비트 수를 지정하는 표기법입니다.
Chorus_hit	트랙에서 언제 코러스가 시작 될지에 대한 추정치, API 호출에서 수신 한 데이터에서 추출
Sections	특정 트랙의 섹션 수
Target	'1'은이 노래가 그 10 년 동안 Hot-100 트랙의 주간 목록 (빌보드 발행)에 한 번 이상 포함되어 '히트'임을 의미, '0'은 트랙이'플랍'임을 의미

데이터 설명 - 변수

19개 변수 중 Track, Artist, Uri를 제외한 16개 변수를 이용하였고 6,257개 데이터로 분석을 진행

- 종속변수(y): Target

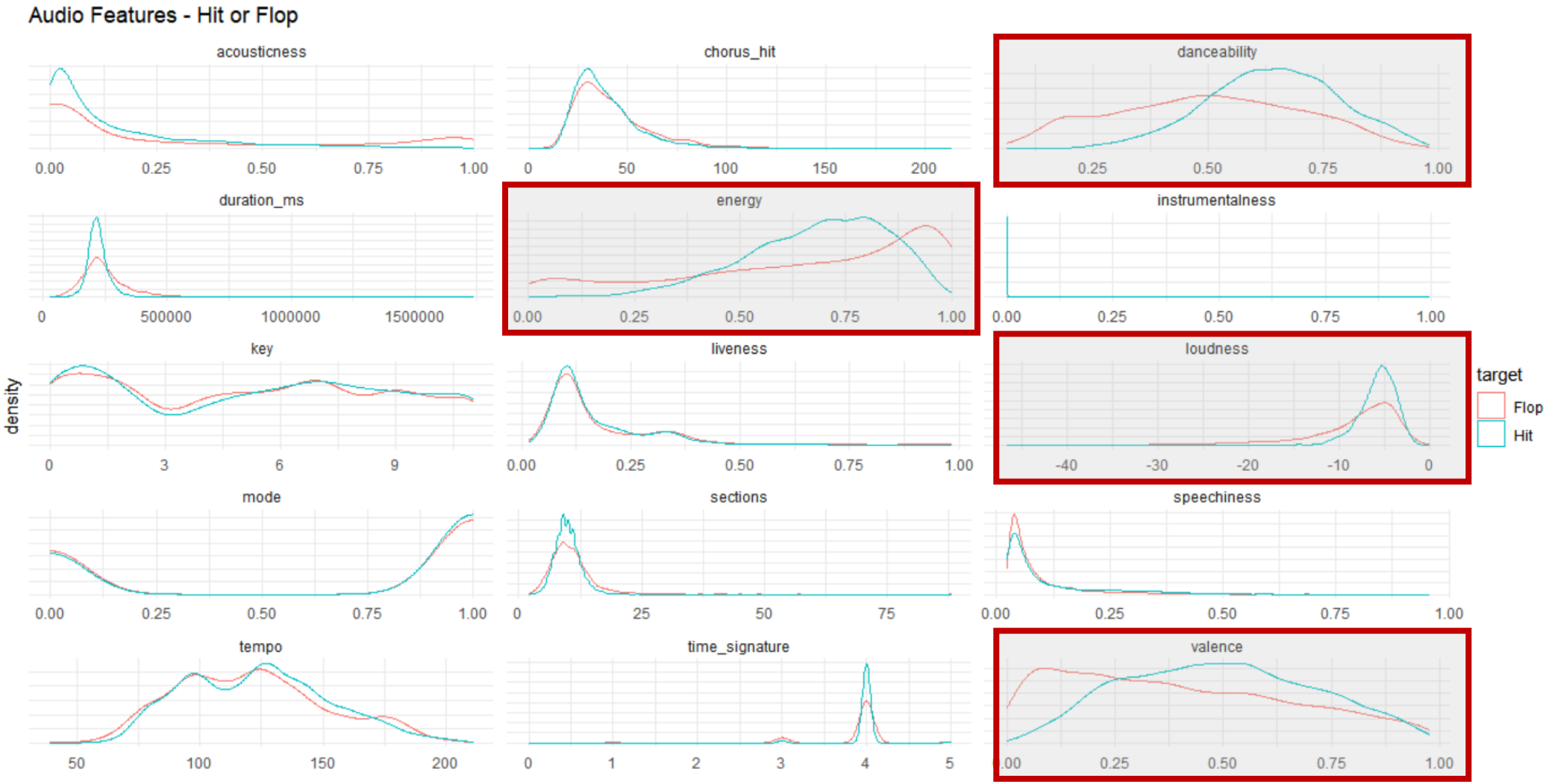
```
> glimpse(spotify)
Observations: 6,398
Variables: 19
$ track            <fct> "Love Me Like Rock and Roll", "I'm Into You
$ artist           <fct> Katie Noel, Jennifer Lopez Featuring Lil Wa
$ uri              <fct> spotify:track:4zO7Nn26sIhg7pClNwQSwx, spoti
$ danceability     <dbl> 0.505, 0.592, 0.364, 0.594, 0.747, 0.360, 0
$ energy           <dbl> 0.610, 0.747, 0.503, 0.637, 0.524, 0.907, 0
$ key              <int> 11, 8, 6, 9, 10, 2, 7, 9, 7, 11, 10, 1, 0,
$ loudness         <dbl> -7.019, -4.439, -18.607, -5.634, -6.8
$ mode             <int> 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0
$ speechiness      <dbl> 0.0446, 0.1350, 0.3460, 0.0826, 0.245
$ acousticness     <dbl> 6.63e-03, 1.81e-02, 4.06e-01, 6.41e-0
$ instrumentalness <dbl> 0.00000, 0.00000, 0.90000, 0.00000, 0.00000
$ liveness         <dbl> 0.1040, 0.0752, 0.1220, 0.0856, 0.2000, 0.1
$ valence          <dbl> 0.2730, 0.7040, 0.4220, 0.2680, 0.3630, 0.2
$ tempo            <dbl> 147.896, 83.929, 95.718, 88.817, 140.053, 1
$ duration_ms      <int> 163749, 200133, 208040, 189200, 213132, 310
$ time_signature   <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4
$ chorus_hit       <dbl> 19.49234, 39.12679, 55.82411, 22.39088, 27.
$ sections         <int> 7, 9, 8, 9, 9, 8, 10, 12, 9, 6, 6, 17, 6, 1
$ target           <int> 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0
```

```
> glimpse(spotify_clean)
Observations: 6,257
Variables: 16
$ target           <fct> Flop, Hit, Flop, Flop, Hit, Flop, Flop, Flop, Flop,
$ danceability     <dbl> 0.505, 0.592, 0.364, 0.594, 0.747, 0.360, 0.560, 0.5
$ energy           <dbl> 0.610, 0.747, 0.503, 0.637, 0.524, 0.907, 0.927, 0.9
$ key              <int> 11, 8, 6, 9, 10, 2, 7, 9, 7, 11, 10, 1, 0, 4, 0, 7,
$ loudness         <dbl> -7.019, -4.439, -18.607, -5.634, -6.807, -4.111, -4.
$ mode             <int> 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1
$ speechiness      <dbl> 0.0446, 0.1350, 0.3460, 0.0826, 0.2450, 0.0949, 0.04
$ acousticness     <dbl> 6.63e-03, 1.81e-02, 4.06e-01, 6.41e-02, 3.06e-02, 5.
$ instrumentalness <dbl> 0.00000, 0.00000, 0.90000, 0.00000, 0.00000, 0.00000
$ liveness         <dbl> 0.1040, 0.0752, 0.1220, 0.0856, 0.2000, 0.1830, 0.08
$ valence          <dbl> 0.2730, 0.7040, 0.4220, 0.2680, 0.3630, 0.2770, 0.45
$ tempo            <dbl> 147.896, 83.929, 95.718, 88.817, 140.053, 139.949, 1
$ duration_ms      <int> 163749, 200133, 208040, 189200, 213132, 310988, 1845
$ time_signature   <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4
$ chorus_hit       <dbl> 19.49234, 39.12679, 55.82411, 22.39088, 27.89104, 13
$ sections         <int> 7, 9, 8, 9, 9, 8, 10, 12, 9, 6, 6, 17, 6, 12, 13, 9,
```

```
> table(is.na(spotify_clean))

 FALSE
100112
> anyNA(spotify_clean)
[1] FALSE
```

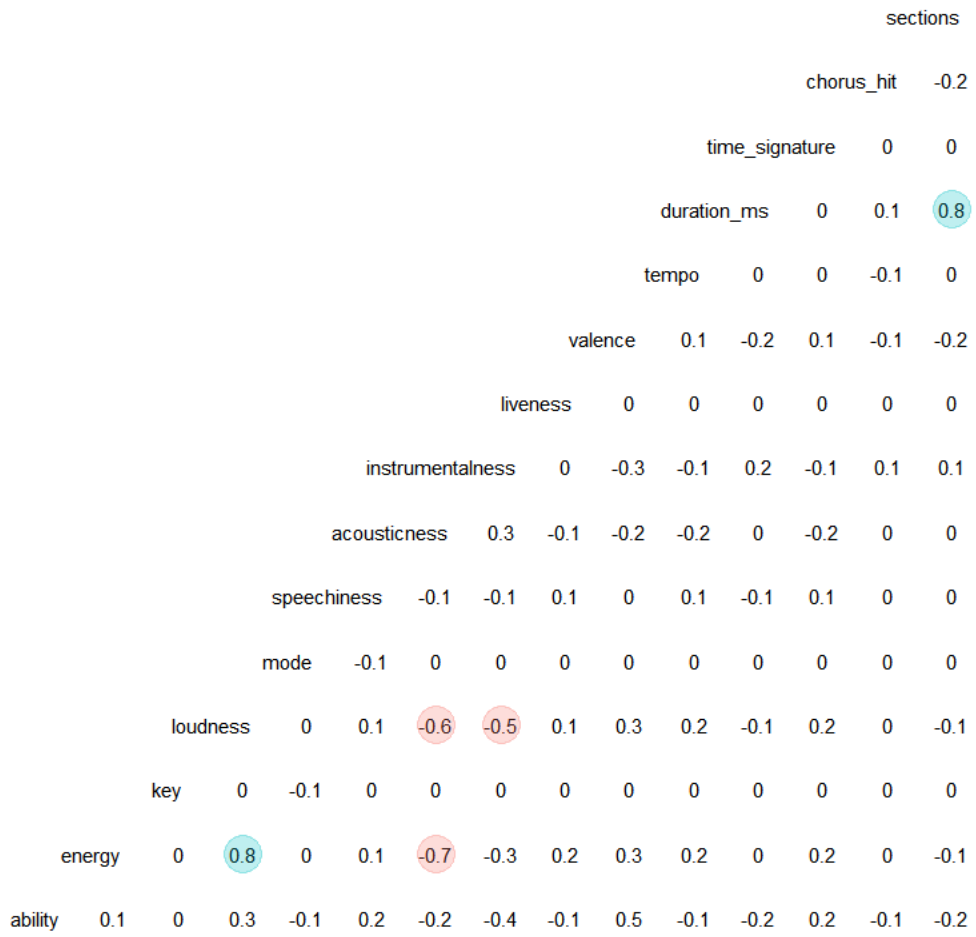
데이터 설명 - 분석

Hit와 변수들의 관계를 살펴보면 danceability가 높고, energy가 조금 높고, loudness가 높고, valence가 적당한 음악이 유행한 것을 볼 수 있음



데이터 설명 - 분석

Ggally를 이용하여 상관계수를 살펴본 결과 loudness와 energy, section와 duration_ms의 상관관계가 높은 것을 확인할 수 있고 energy, section, instrumentalness변수를 삭제함




```
> glimpse(spotify_clean_model)
Observations: 6,257
Variables: 13
$ target          <fct> Flop, Hit, Flop, Flop, Hit, Flop, Flop, Flop, Flop, Hit, Hit, Flop,
$ danceability    <dbl> 0.505, 0.592, 0.364, 0.594, 0.747, 0.360, 0.560, 0.516, 0.488, 0.78
$ energy          <dbl> 0.610, 0.747, 0.503, 0.637, 0.524, 0.907, 0.927, 0.958, 0.510, 0.66
$ key             <int> 11, 8, 6, 9, 10, 2, 7, 9, 7, 11, 10, 1, 0, 4, 0, 7, 3, 0, 1, 1, 9,
$ mode            <int> 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1
$ speechiness     <dbl> 0.0446, 0.1350, 0.3460, 0.0826, 0.2450, 0.0949, 0.0466, 0.1280, 0.0
$ acousticness    <dbl> 6.63e-03, 1.81e-02, 4.06e-01, 6.41e-02, 3.06e-02, 5.87e-04, 2.69e-0
$ liveness        <dbl> 0.1040, 0.0752, 0.1220, 0.0856, 0.2000, 0.1830, 0.0888, 0.0679, 0.1
$ valence         <dbl> 0.2730, 0.7040, 0.4220, 0.2680, 0.3630, 0.2770, 0.4590, 0.4840, 0.3
$ tempo           <dbl> 147.896, 83.929, 95.718, 88.817, 140.053, 139.949, 110.063, 104.717
$ duration_ms     <int> 163749, 200133, 208040, 189200, 213132, 310988, 184581, 232520, 219
$ time_signature  <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4
$ chorus_hit      <dbl> 19.49234, 39.12679, 55.82411, 22.39088, 27.89104, 133.65366, 35.673
```

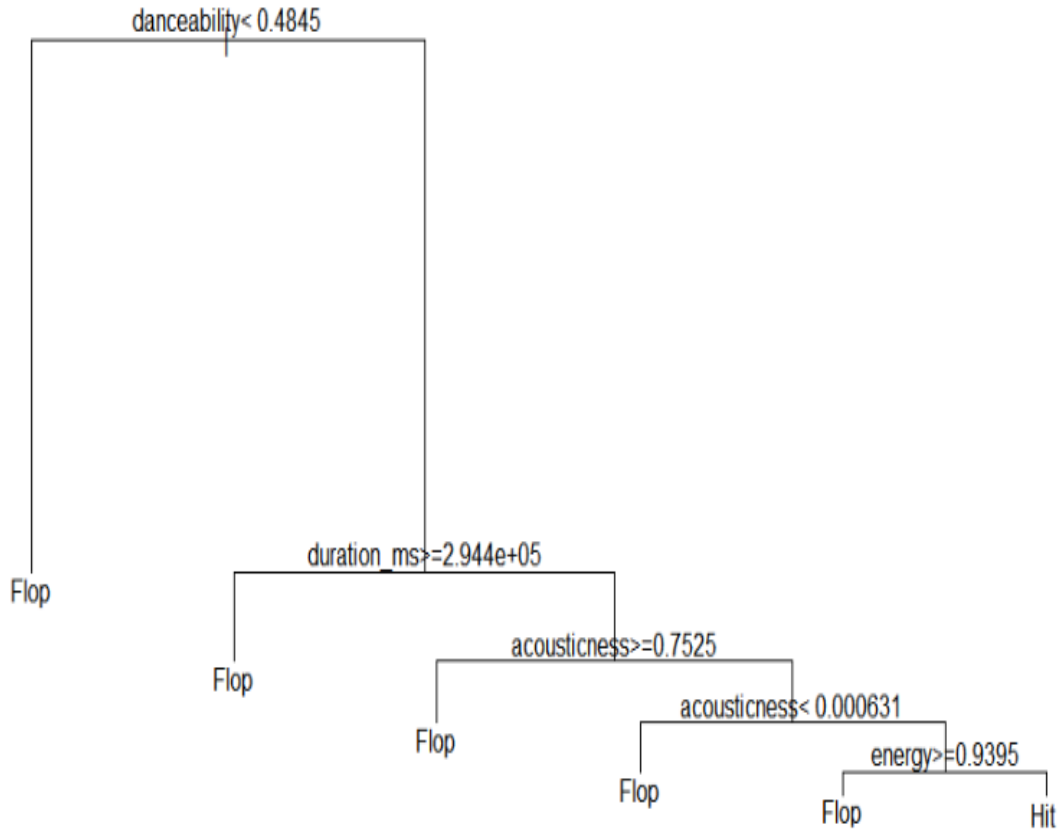
분석 기법 - Rpart

Train 80%와 Test 20%비율로 데이터를 분류하고 변수비율과 Train 데이터로 rpart 시행

- 변수비율: Train데이터에서 Hit이 48% Flop이 52% 비율, Test데이터에서 Hit이 51% Flop이 49% 비율
- Rpart를 이용하여 tree를 그려본 결과 danceability, duration, acousticness, energy 변수로 Flop, Hit을 구분

```
> prop.table(table(spotify_train$target))

     Flop       Hit
0.4831169 0.5168831
> prop.table(table(spotify_test$target))

     Flop       Hit
0.5111821 0.4888179
```

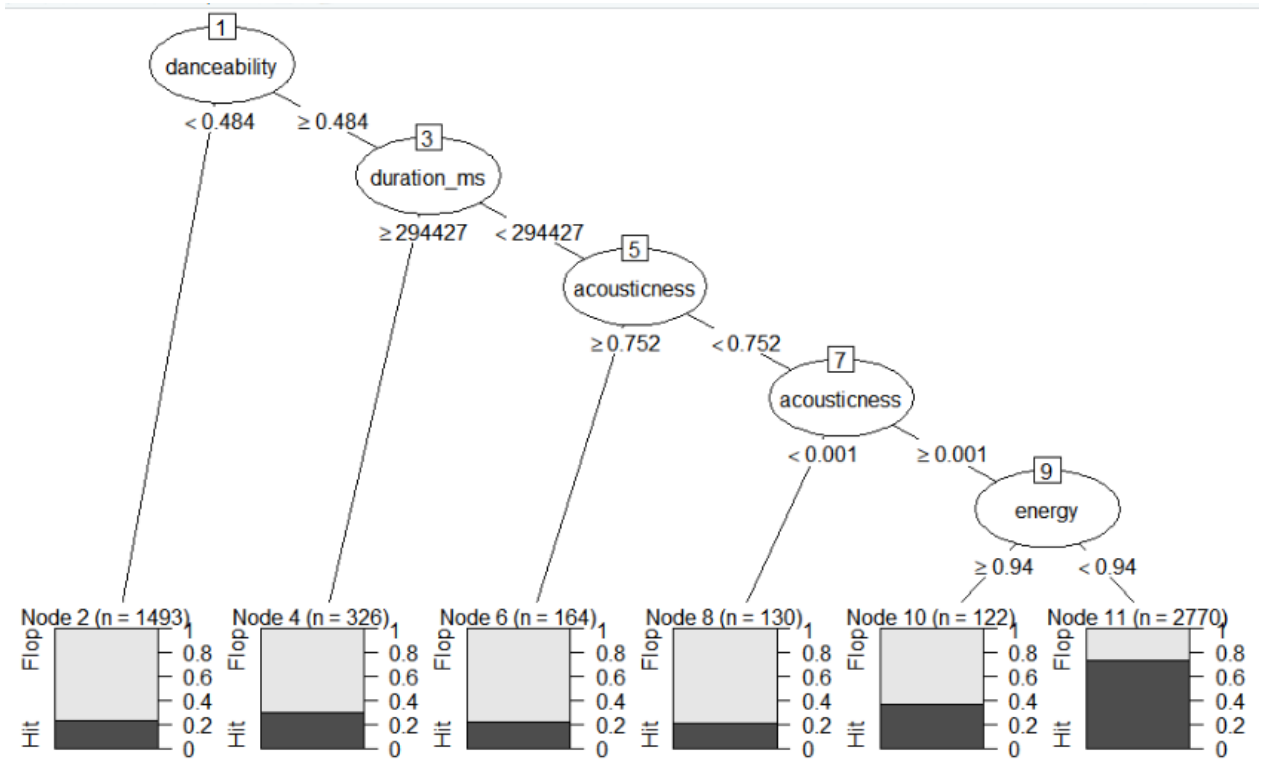
```
> as.party(spotify_dt)

Model formula:
target ~ danceability + energy + key + mode + speechiness + acousticness +
    liveness + valence + tempo + duration_ms + time_signature +
    chorus_hit

Fitted party:
[1] root
|   [2] danceability < 0.4845: Flop (n = 1493, err = 22.7%)
|   [3] danceability >= 0.4845
|   |   [4] duration_ms >= 294427: Flop (n = 326, err = 29.1%)
|   |   [5] duration_ms < 294427
|   |   |   [6] acousticness >= 0.7525: Flop (n = 164, err = 21.3%)
|   |   |   [7] acousticness < 0.7525
|   |   |   |   [8] acousticness < 0.00063: Flop (n = 130, err = 20.8%)
|   |   |   |   [9] acousticness >= 0.00063
|   |   |   |   |   [10] energy >= 0.9395: Flop (n = 122, err = 36.1%)
|   |   |   |   |   [11] energy < 0.9395: Hit (n = 2770, err = 26.5%)

Number of inner nodes:    5
Number of terminal nodes: 6
```

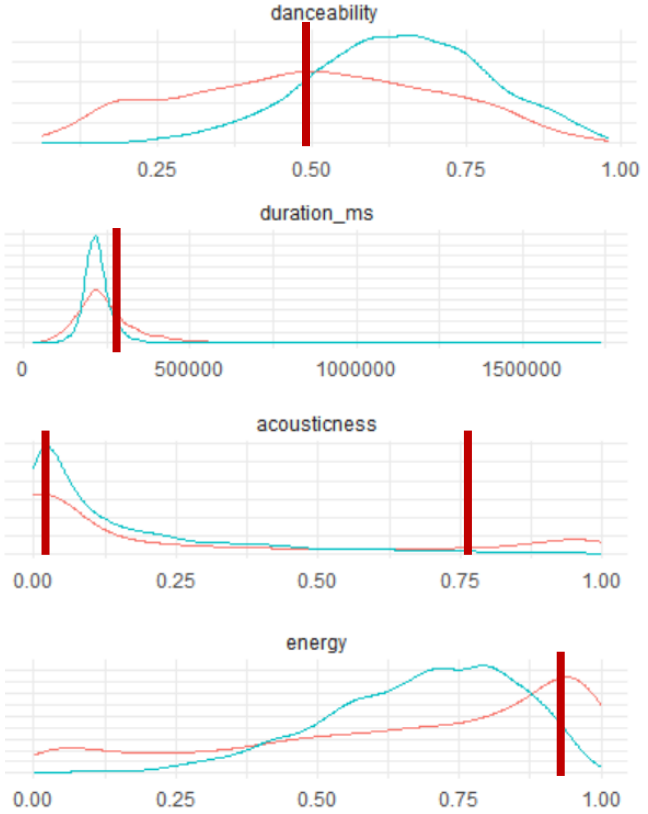


분석 기법 – Rpart

Rpart Plot 결과

- Rpart를 이용하여 tree를 그려본 결과 danceability, duration, acousticness, energy 변수로 Flop, Hit을 구분
- 변수들의 분포도는 오른쪽 그림과 같으며 차이가 나는 부분이 잘 나뉘어지고 있어 tree분류가 잘 된것을 알 수 있음





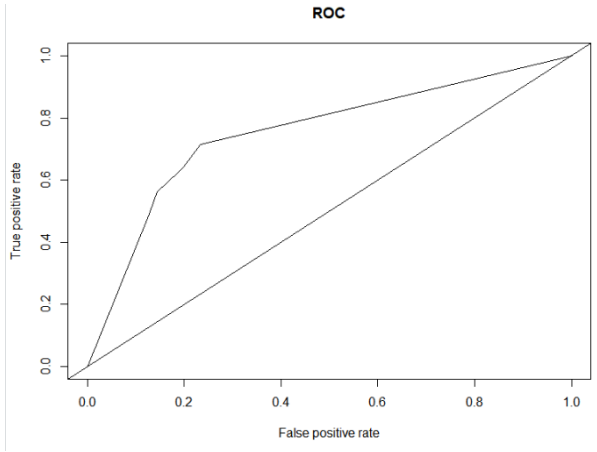
분석 기법 - Rpart

- Confusion Matrix and statistics

		Predicted Values	
		Flop(0)	Hit(1)
Actual Values	Flop(0)	458	182
	Hit(1)	143	469

- **Accuracy = 0.740**
- Sensitivity = 0.716
- Specificity = 0.766
- Precision = 0.762

- ROC 커브 및 AUC 결과




```
> auc_ROCR_dt
[1] 0.730025
```

분석 기법 – Random Forest

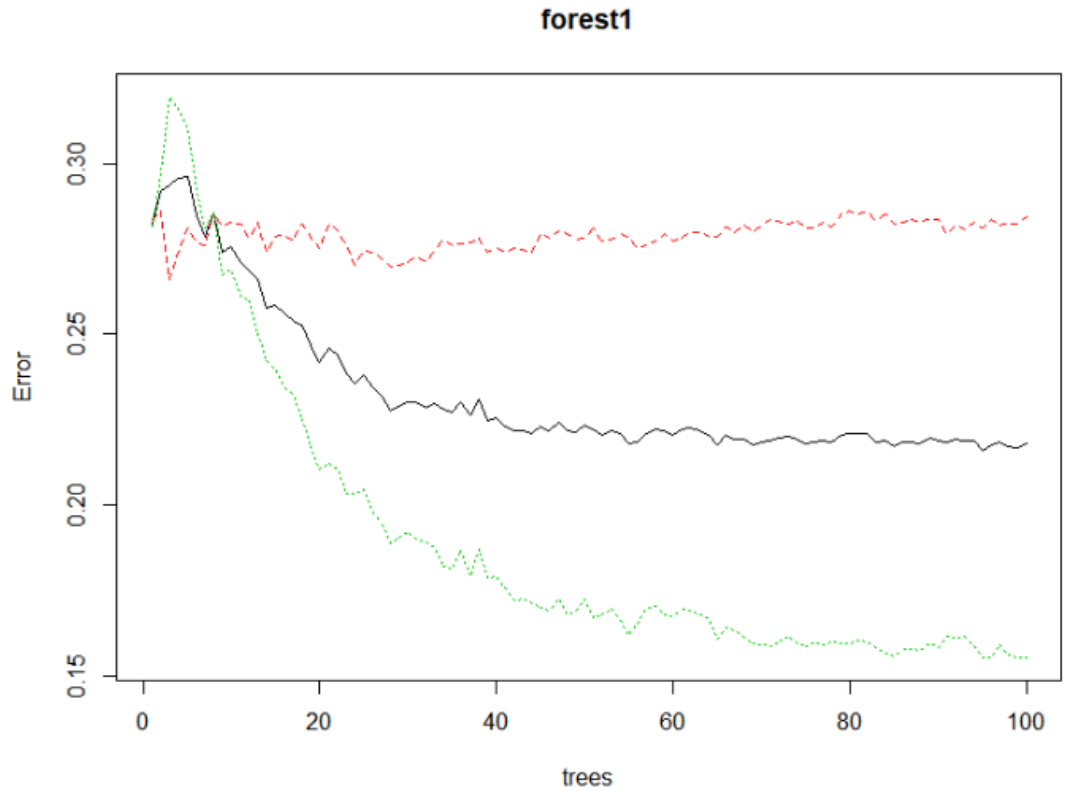
Bagging을 활용하는 Random Forest의 적합성을 살펴봄
- Bagging은 변수들을 하나씩 제거하면서 예측력을 높이는데 초점을 맞춘 분석기법임

```
> forest1 <- randomForest(target ~ ., ntree = 100, importance = TRUE, data = spotify_train)
> forest1

Call:
 randomForest(formula = target ~ ., data = spotify_train, ntree = 100,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 3

        OOB estimate of  error rate: 21.78%
Confusion matrix:
     Flop  Hit class.error
Flop 1730  688   0.2845327
Hit   402 2185   0.1553923
```

분석 기법 – Random Forest

오분류 그래프를 다음과 같이 그려볼 수 있고 70번 이후 안정적인 그래프를 보여주고 있음

- OOB(검은색)
- Hit(초록색)
- Flop(빨간색)

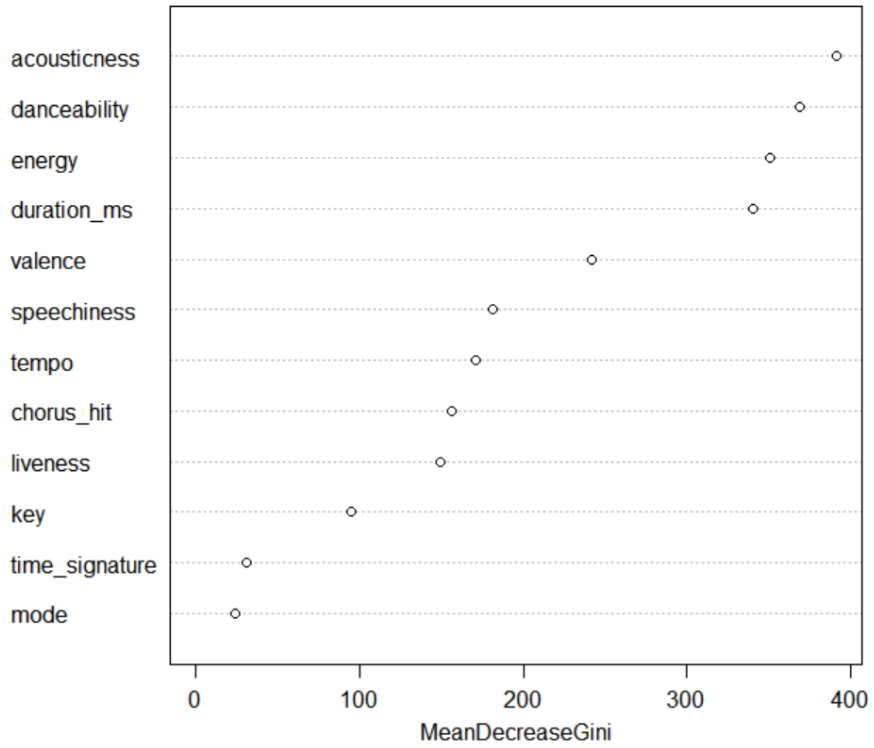



```
> head(forest1$err.rate)
            OOB      Flop       Hit
[1,] 0.2822888 0.2833530 0.2813765
[2,] 0.2917913 0.2861111 0.2970045
[3,] 0.2937400 0.2658863 0.3194444
[4,] 0.2956170 0.2735338 0.3162748
[5,] 0.2961890 0.2811634 0.3102111
[6,] 0.2845580 0.2767936 0.2917522
```

분석 기법 – Random Forest

Importance는 변수가 얼마만큼 영향을 미치는지 나타낸 것으로 acousticness, danceability, energy순으로 많은 영향을 미치고 있고, 오른쪽은 이를 그래프화 시킨 모습임

	MeanDecreaseGini
danceability	368.85777
energy	350.42565
key	94.67043
mode	23.54259
speechiness	181.72583
acousticness	391.65769
liveness	149.28606
valence	242.26761
tempo	170.58623
duration_ms	340.27554
time_signature	30.60944
chorus_hit	156.28199



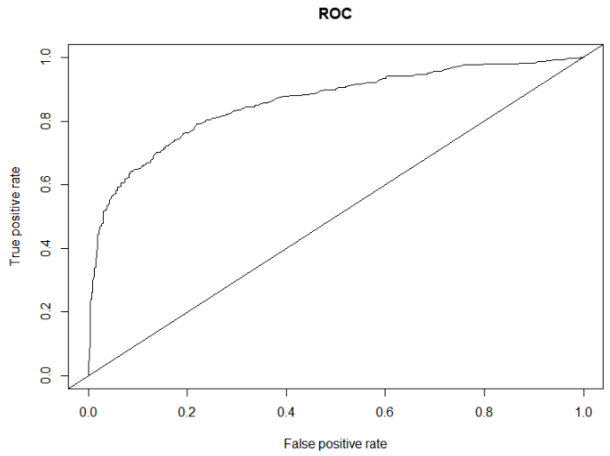
분석 기법 – Random Forest

- Confusion Matrix and statistics

		Predicted Values	
		Flop(0)	Hit(1)
Actual Values	Flop(0)	449	191
	Hit(1)	85	527

- **Accuracy = 0.780**
- Sensitivity = 0.702
- Specificity = 0.861
- Precision = 0.841

- ROC 커브 및 AUC 결과




```
> auc_ROCR_f
[1] 0.8521055
```

분석 기법 – xgboost

적합결여도를 낮춰주는 xgboost를 사용한 결과 199번째부터 error값이 동일함

- Xgboost는 변수들을 하나씩 추가하면서 적합도를 높이는데 초점을 맞춘 분석기법임

```
> bst <- xgboost(    data          = x_train,
+                    label         = y_train,
+                    nrounds       = 400,
+                    num_class     = 2
+ )
[1]     train-merror:0.214386
[2]     train-merror:0.199800
[3]     train-merror:0.190809
[4]     train-merror:0.185215
[5]     train-merror:0.179021
[6]     train-merror:0.174426
[7]     train-merror:0.172228
[8]     train-merror:0.167832
[9]     train-merror:0.160040
[10]    train-merror:0.152448
[11]    train-merror:0.152048
[12]    train-merror:0.151848
```

```
[162]   train-merror:0.002398
[163]   train-merror:0.001998
[164]   train-merror:0.001998
[165]   train-merror:0.001798
[166]   train-merror:0.001798
[167]   train-merror:0.001998
[168]   train-merror:0.001798
[169]   train-merror:0.001598
[170]   train-merror:0.001598
[171]   train-merror:0.001399
[172]   train-merror:0.001399
[173]   train-merror:0.001399
[174]   train-merror:0.001399
[175]   train-merror:0.001199
[176]   train-merror:0.001199
[177]   train-merror:0.001199
[178]   train-merror:0.001199
```

```
[195]   train-merror:0.000799
[196]   train-merror:0.000799
[197]   train-merror:0.000599
[198]   train-merror:0.000599
[199]   train-merror:0.000400
[200]   train-merror:0.000400
[201]   train-merror:0.000400
[202]   train-merror:0.000400
[203]   train-merror:0.000400
[204]   train-merror:0.000400
[205]   train-merror:0.000400
[206]   train-merror:0.000400
```

분석 기법 – xgboost

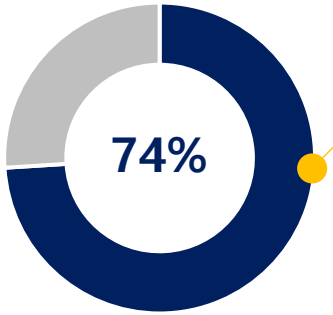
- Confusion Matrix and statistics

		Predicted Values	
		Flop(0)	Hit(1)
Actual Values	Flop(0)	461	179
	Hit(1)	106	506

- **Accuracy = 0.772**
- Sensitivity = 0.813
- Specificity = 0.739

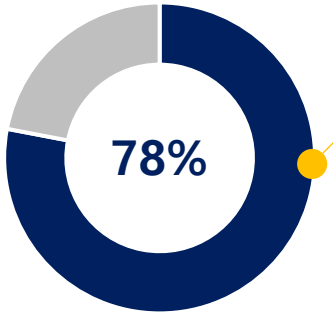
```
> confusionMatrix(f_y_test, f_prd0)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 461 179
         1 106 506

               Accuracy : 0.7724
                 95% CI : (0.7481, 0.7953)
    No Information Rate : 0.5471
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5457

 Mcnemar's Test P-Value : 2e-05

            Sensitivity : 0.8131
            Specificity : 0.7387
         Pos Pred Value : 0.7203
         Neg Pred Value : 0.8268
             Prevalence : 0.4529
         Detection Rate : 0.3682
   Detection Prevalence : 0.5112
      Balanced Accuracy : 0.7759

       'Positive' Class : 0
```

분석 비교 – Rpart, Random Forest, xgboost



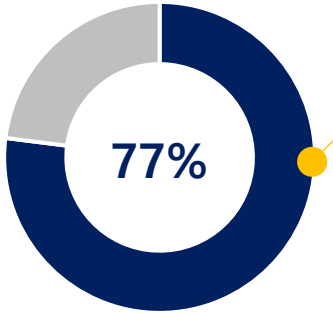
Rpart

```
> table(spotify_dt_table$target,spotify_dt_table$target_pred)

      Flop Hit
  Flop  458 182
  Hit   143 469
```



Random Forest

```
> table(spotify_forest_table$target,spotify_forest_table$target_pred)

      Flop Hit
  Flop  449 191
  Hit    85 527
```



xgboost

```
> table(y_test, prd0)
      prd0
y_test   0   1
     0 461 179
     1 106 506
```

- **Random Forest**
-실제 Hit한 값을 가장 정확하게 예측

- **xgboost**
- 실제 Flop한 값을 가장 정확하게 예측

분석 비교 – Rpart, Random Forest, xgboost

세개의 분석결과를 비교해 본 결과 어떤 하나의 분석기법이 뛰어난 예측력을 보여주지는 못했지만 전반적인 성공한 음원들의 공통된 요소를 파악하고자 하는 목적이 있기 때문에 넓은 관점에서 이해할 필요가 있음

활용 가능성	• 음원 수익 창출을 위해 해당 모델을 기반으로 나온 장르나 특징을 활용하여 대중들이 선호할 수 있는 음반 작업을 데 도움을 줄 수 있음
보완점	• 음원은 미디어의 일부이기 때문에 각 시기의 트렌드에 굉장히 밀접하게 변화함 • 10년간 음원 차트 분석을 통해 나온 결과값 역시 각 시기별 트렌드를 세분화 하여 반영되기 어려움이 있음 • 전반적인 성공한 음원들의 공통된 요소를 파악하고자 하는 목적이 있기 때문에 넓은 관점에서 이해할 필요가 있음