IOD: Project 2

Popularity predictor with Spotify's audio features

Wong Kar Mun

Problem:

Using features in tracks to predict popularity using Regression

Features
(17 features reduced to 6 features)

1. Year
2. Acousticness
3. Speechiness
4. Danceability
5. Instrumentalness
6. Liveness

- Dataset: 166k tracks
- Ranges from 1921 - 2020
- Normalized with MinMaxScaler

Equation with Linear Regression

$$Popularity = -0.06 + 0.68 year + -0.04 acousticness + -0.07 speechiness + 0.04 danceability + -0.03 instrumentalness + -0.03 liveness$$

Questions?

- What other Models to use to improve score?
- How can we choose better parameters for the different models?
- Are there other methods besides regression to solve problem?

Linear Regression

Forward Feature Selection (Equation)

$$Popularity = -0.06 + 0.68 year + -0.04 acousticness + -0.07 speechiness + 0.04 danceability + -0.03 instrumentalness + -0.03 liveness$$

1 Test size VS CV

Test Size

CV	0.1	0.2	0.3	0.4	0.5
5	0.7842	0.7833	0.7834	0.7833	0.7833
10	0.7842	0.7833	0.7834	0.7833	0.7833

2 Metrics

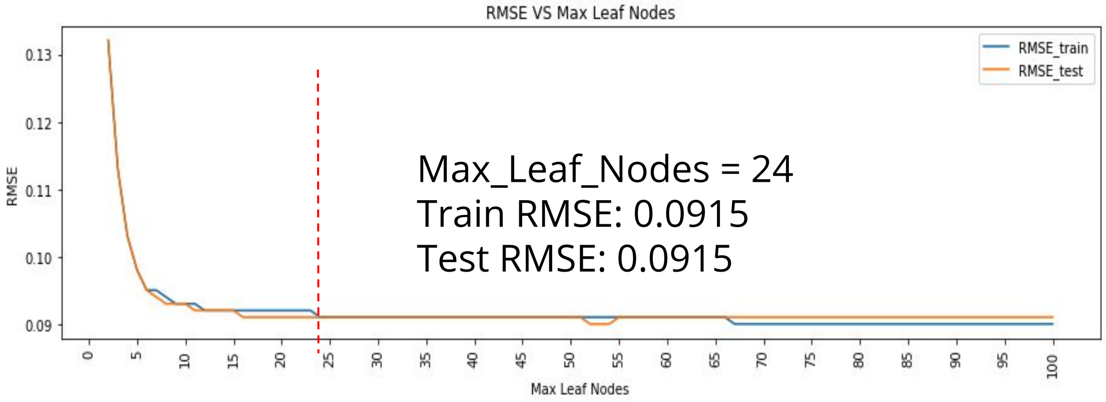
Model	Param	CV score cv=10	Train RMSE	Test RMSE	Test - Train RMSE (VAR)
Linear - FF	test = 0.1	0.7842	0.0998	0.0998	0.0000
Linear - FF	test = 0.3	0.7834	0.0998	0.0998	0.0000

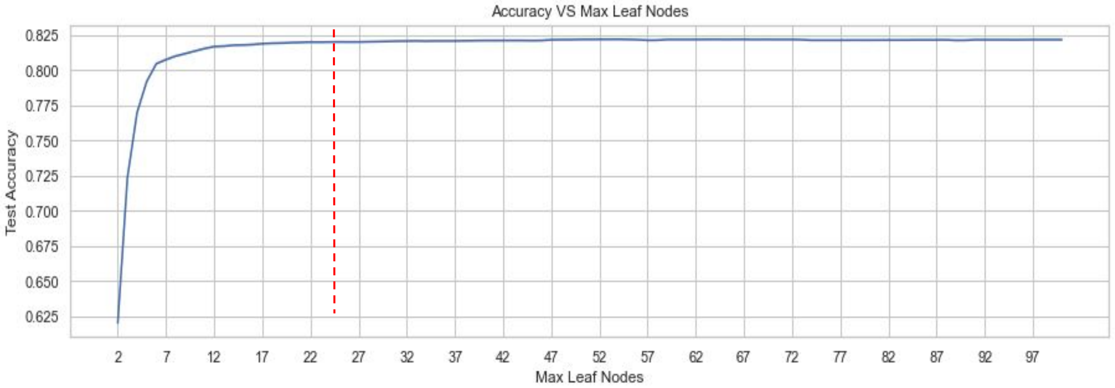
3 Consistency of CV = 10

CV=10

	0.3 Linear Reg	0.1 Linear Reg
0	0.7865	0.7866
1	0.7829	0.7832
2	0.7899	0.7795
3	0.7827	0.7838
4	0.7794	0.7818
5	0.7857	0.7842
6	0.7822	0.7847
7	0.7853	0.7906
8	0.7849	0.7826
9	0.7849	0.7858

Decision Tree





1

Test Size

CV	0.1	0.2	0.3	0.4	0.5
5	0.8184	0.8179	0.8175	0.8173	0.8172
10	0.8184	0.8179	0.8176	0.8173	0.8174

2

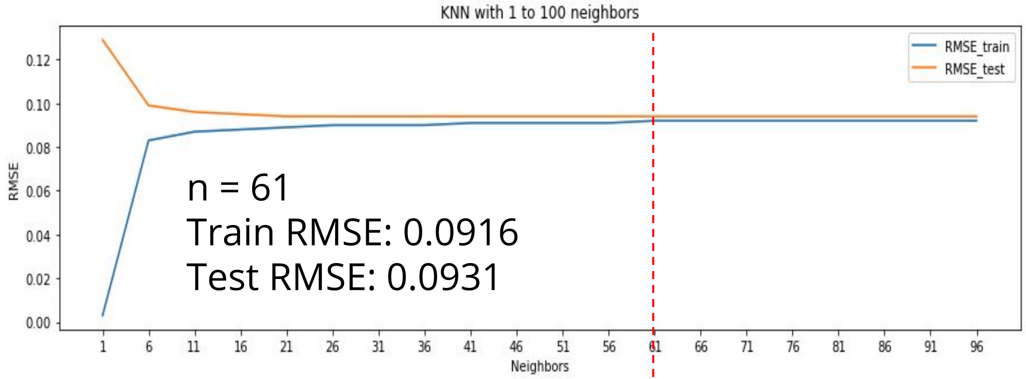
Model	Param	CV score cv=10	Train RMSE	Test RMSE	Test - Train RMSE (VAR)
DT Reg	max_leaf=24 test = 0.1	0.8187	0.0915	0.0915	0.0000
DT Reg	max_leaf=24 test = 0.3	0.8173	0.0915	0.0915	0.0000
DT Reg	max_leaf=12 test = 0.3	0.8153	0.0923	0.0924	0.0001

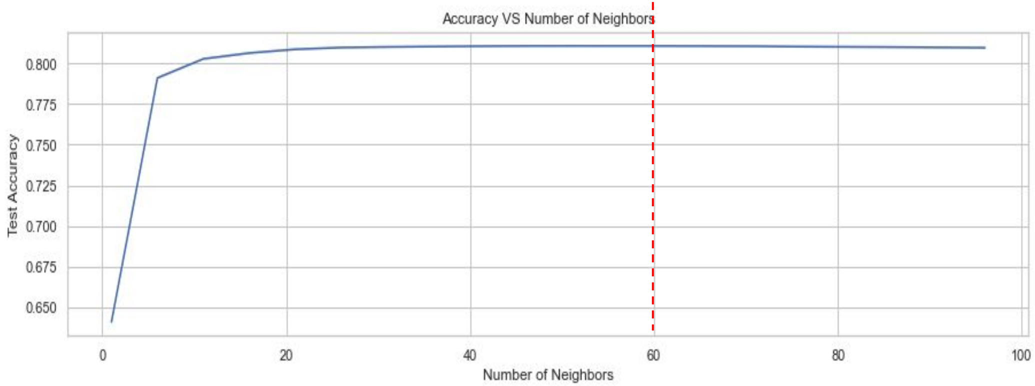
3

CV=10

	0.3 DT Reg		0.1 DT Reg
0	0.8162	0	0.8183
1	0.8239	1	0.8235
2	0.8202	2	0.8173
3	0.8177	3	0.8207
4	0.8135	4	0.8233
5	0.8124	5	0.8119
6	0.8193	6	0.8144
7	0.8173	7	0.8155
8	0.8185	8	0.8202
9	0.8138	9	0.8214

KNN Regression





1

Test Size

CV	0.1	0.2	0.3	0.4	0.5
5	0.8108	0.8094	0.8088	0.8078	0.8069
10	0.8114	0.8102	0.8094	0.8085	0.8075

2

Model	Param	CV score cv=10	Train RMSE	Test RMSE	Test - Train RMSE (VAR)
KNN Reg	n=61, test=0.1	0.8114	0.0916	0.0931	0.0015
KNN Reg	n=61, test=0.3	0.8075	0.0916	0.0931	0.0015
KNN Reg	n=21, test=0.3	0.8102	0.0892	0.0936	0.0044

3

CV=10

0.3

	KNN Reg
0	0.8022
1	0.8041
2	0.8004
3	0.8094
4	0.8190
5	0.8046
6	0.8106
7	0.8044
8	0.8091
9	0.8110

0.1

	KNN Reg
0	0.8135
1	0.8041
2	0.8153
3	0.8131
4	0.8056
5	0.8123
6	0.8116
7	0.8096
8	0.8162
9	0.8132

Random Forest Regression

2.2.1 Find the best parameter

```
# Perform Grid-Search
gsc = GridSearchCV(
estimator=RandomForestRegressor(),
param_grid={
        'max_depth': range(3,7),
        'n_estimators': (10, 50, 100,500,1000),},cv=5,
        scoring='neg_root_mean_squared_error', verbose=0,n_jobs=-1)

grid_result = gsc.fit(X_train, y_train)
best_params = grid_result.best_params_
```

```
best_params
```

```
{'max_depth': 6, 'n_estimators': 100}
```

Train RMSE: 0.0906
Test RMSE: 0.0909

1

Test Size

CV	0.1	0.2	0.3	0.4	0.5
5	0.8211	0.8205	0.8202	0.8202	0.8203
10	0.8211	0.8204	0.8202	0.8202	0.8203

2

Model	Param	CV score cv=10	Train RMSE	Test RMSE	Test - Train RMSE (VAR)
Random Forest	n=100, test=0.1	0.8211	0.0906	0.0909	0.0003
Random Forest	n=100, test=0.3	0.8210	0.0906	0.0909	0.0003
Random Forest	n=500, test=0.3	0.8208	0.0906	0.0909	0.0003

3

CV=10

	0.3 RF Reg		0.1 RF Reg
0	0.8216	0	0.8222
1	0.8240	1	0.8135
2	0.8243	2	0.8241
3	0.8128	3	0.8218
4	0.8215	4	0.8156
5	0.8208	5	0.8237
6	0.8194	6	0.8208
7	0.8208	7	0.8194
8	0.8239	8	0.8261
9	0.8208	9	0.8237

Comparison of all Models

CV=10

	Linear Reg	KNN Reg	DT Reg	RF Reg	Stacking Reg
0	0.7865	0.8135	0.8183	0.8222	0.8236
1	0.7829	0.8041	0.8235	0.8135	0.8145
2	0.7899	0.8153	0.8173	0.8241	0.8253
3	0.7827	0.8131	0.8207	0.8218	0.8231
4	0.7794	0.8056	0.8233	0.8156	0.8167
5	0.7857	0.8123	0.8119	0.8237	0.8244
6	0.7822	0.8116	0.8144	0.8208	0.8220
7	0.7853	0.8096	0.8155	0.8194	0.8204
8	0.7849	0.8162	0.8202	0.8261	0.8270
9	0.7849	0.8132	0.8214	0.8237	0.8246

Mean CV score and RMSE

Model	Param	CV score cv=10	Train RMSE	Test RMSE	Test - Train RMSE (VAR)
Stacking	test = 0.1	0.8221	0.0901	0.0906	0.0005
Random Forest	n=100, test=0.1	0.8211	0.0906	0.0909	0.0003
DT Reg	max_leaf=24 test = 0.1	0.8187	0.0915	0.0915	0.0000
KNN Reg	n=61, test=0.1	0.8114	0.0916	0.0931	0.0015
Linear - FF	test = 0.1	0.7842	0.0998	0.0998	0.0000

Actual vs Predictions of all Models

Target

Predictions

id	year	acousticness	speechiness	danceability	instrumentalness	liveness	popularity	LR prediction	KNN prediction	Random Forest prediction	Stacking prediction
RxglqC79gXUMI8jPMVc	0.69697	0.397590	0.045098	0.468623	0.000013	0.0712	0.44	0.41	0.42	0.39	0.40
gAdATsfINRIljv4LbrKX2	0.69697	0.094779	0.025593	0.498988	0.004670	0.1000	0.35	0.43	0.42	0.41	0.41
pRsuHfZ8YcdNaPZVl8u	0.69697	0.529116	0.031166	0.706478	0.000069	0.1420	0.32	0.42	0.41	0.41	0.40
JOum4NudWQvWXXwU	0.69697	0.012651	0.039009	0.559717	0.000139	0.0876	0.31	0.43	0.40	0.39	0.39
53rV8ViOYJ02LtjgEihele	0.69697	0.027209	0.271414	0.809717	0.000000	0.0610	0.35	0.43	0.40	0.40	0.40

Conclusion

Implemented ensembling method, stacking. Accuracy Score and Test RMSE observed to have improvements.

Con: Generating predictions is slower and more computationally expensive.

THE END