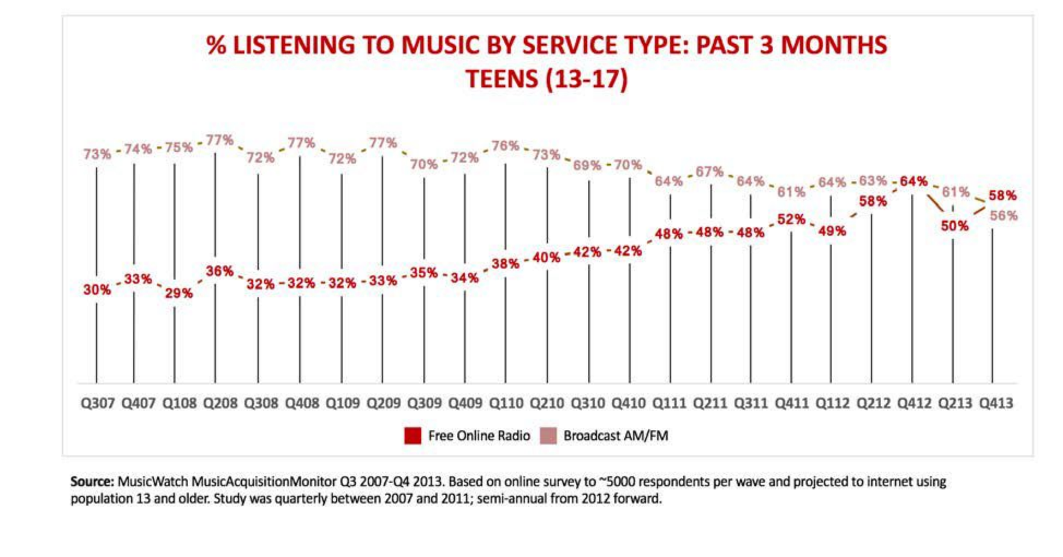
Twitter Killed the Radio Star

Thinkful Supervised Learning Capstone

Conner Brown

23 May 2018

Motivation



How do radio stations react to streaming services?

Common suggestion: get on social media!

Does this have an impact on radio station popularity?

What other factors affect radio station popularity?

Data Scraping

AllAccess.com Data
(Selenium)

Nielsen Markets Table	Market Name, Demographics, State, Market Stations Link

Market Stations Table	Format, Listeners per season, Owner, Station Link

Twitter Data
(tweepy)

Twitter API	Tweets, Followers, Statuses, Favourites, Created At

Data

Rows 2110
Columns 17

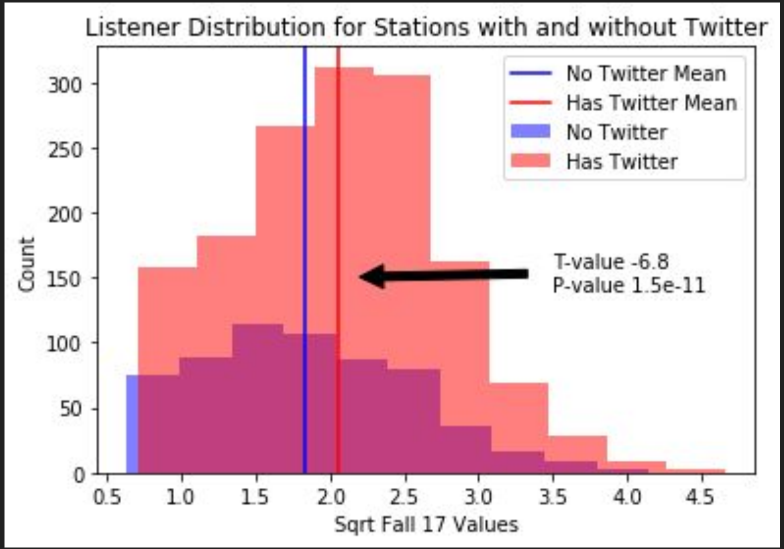
AllAccess.com Data

Twitter Data

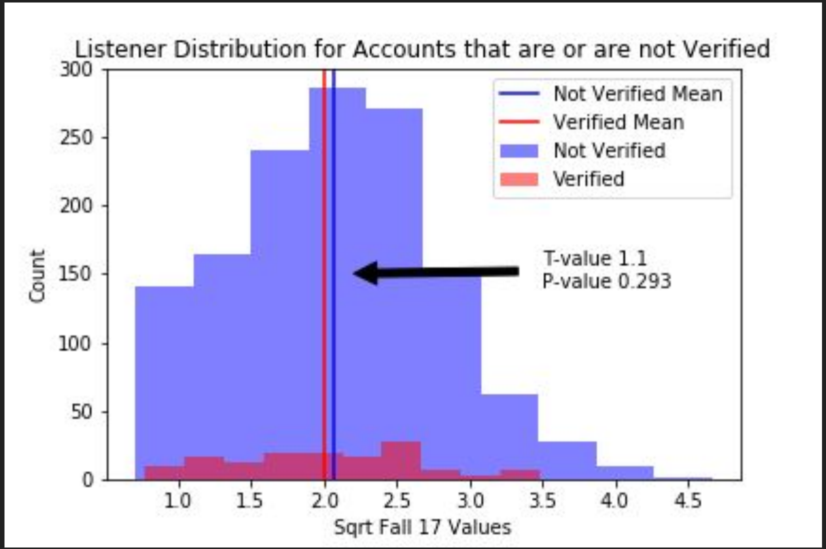
	Format	Population	Spr 17	Spr 16	Black	State	Fall 16	Hispanic	Owner	Fall 17	Followers Count	Friends Count	Listed Count	Created At	Favourites Count	Verified	Statuses Count
0	Country	144700.0	7.5	8.7	11500.0	TX	6.3	31500.0	Townsquare	9.3	2030.0	1754.0	25.0	2010-11-08 20:22:46	191.0	False	10156.0
1	Classic Hits	144700.0	7.5	7.2	11500.0	TX	6.3	31500.0	Townsquare	7.9	578.0	475.0	11.0	2010-12-07 17:00:43	25.0	False	21834.0
2	Top 40/M	144700.0	6.8	7.2	11500.0	TX	7.7	31500.0	Cumulus	6.6	235.0	36.0	12.0	2009-12-17 09:53:19	1.0	False	12.0

Data Type	Columns
Categorical	Format, State, Owner
Numeric	Population, Black, Hispanic, Followers Count, Friends Count, Listed Count, Favourites Count, Statuses Count
Datetime	Created At
Boolean	Verified
Time Series	Spr 16, Fall 16, Spr 17, Fall 17

Data Statistics



- P-value << 0.05
- Difference is most likely NOT due to chance
- Safe to assume stations with twitter accounts have more listeners



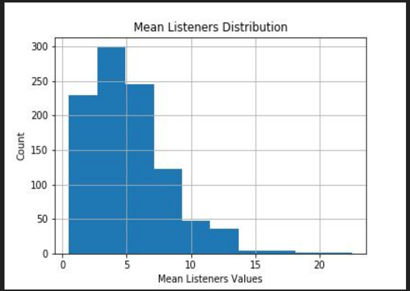
- P-value > 0.05
- Difference may be due to chance
- High variance in ‘verified’ data (what’s going on here?)

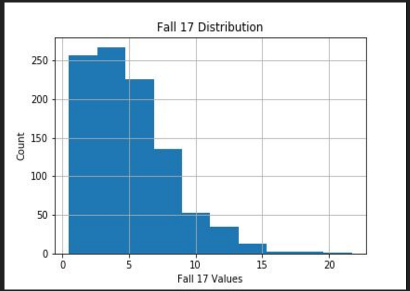
Feature Engineering (derived)

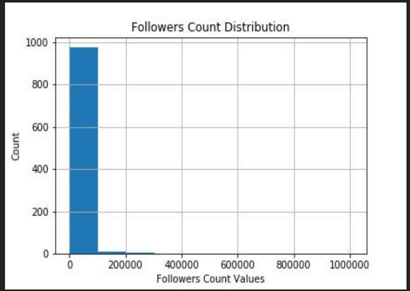
```
# grouped columns based on seasonal listeners
df_listeners = df_feat[['Spr 16','Fall 16','Spr 17']]
av_listeners = df_listeners.mean(axis = 1)
df_feat['Low Listeners'] = np.transpose([av_listeners < 3])
df_feat['Mid Listeners'] = np.transpose([(av_listeners < 7) & (av_listeners >= 3)])
df_feat['High Listeners'] = np.transpose([(av_listeners < 12) & (av_listeners >= 7)])
df_feat['Stellar Listeners'] = np.transpose([av_listeners >= 12])
```

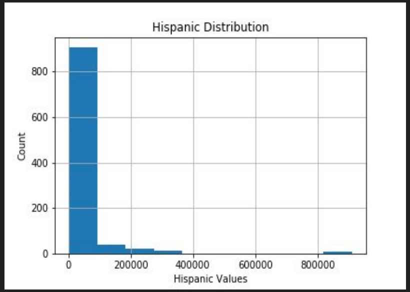
```
# time series stats
df_feat['Mean Listeners'] = av_listeners
df_feat['Std Listeners'] = df_listeners.std(axis = 1)
slopes = []
intercepts = []
for row in df_feat.index:
    slope, intercept,r_value, p_value, std_err = stats.linregress(list(range(1,4,1)),df_listeners.loc[row])
    slopes.append(slope)
    intercepts.append(intercept)
df_feat['Slope Listeners'] = slopes
df_feat['Intercept Listeners'] = intercepts
```

Feature Engineering (continuous)



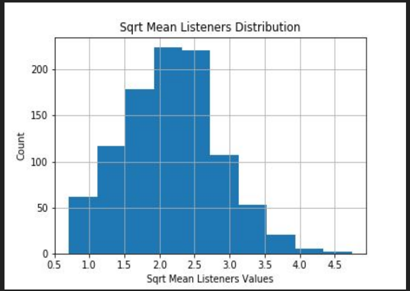


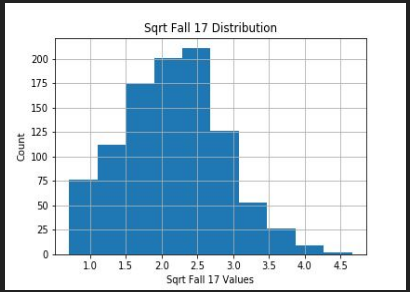


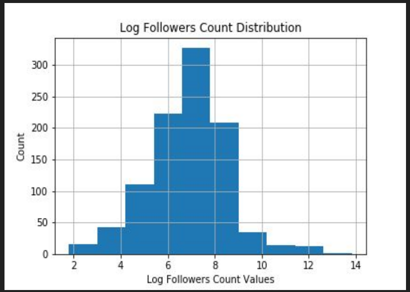


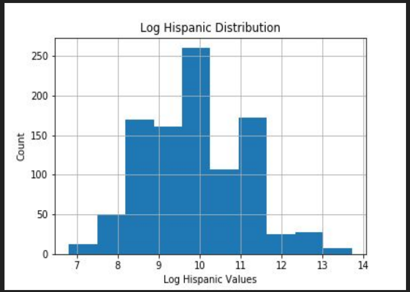
Square Root(X)

Log(X + 1)









More normal!

No more outliers!

Normal-ish

Feature Engineering (categorical)

```
59 unique Formats

Country          580
Top 40/M         363
AC               289
Classic Rock     273
Sports           268
Talk             266
N/T              250
Classic Hits     223
Hot AC           205
Top 40/R         120
Name: Format, dtype: int64
```

```
52 unique States

CA    393
TX    306
FL    278
NY    263
NC    172
PA    165
IL    120
OH    117
MI    111
IA    105
Name: State, dtype: int64
```

```
568 unique Owners

iHeartMedia    890
Cumulus        404
Entercom       302
Townsquare     250
Alpha          119
Cox Radio       67
Midwest         67
Univision       59
Beasley         57
Urban One       50
Name: Owner, dtype: int64
```

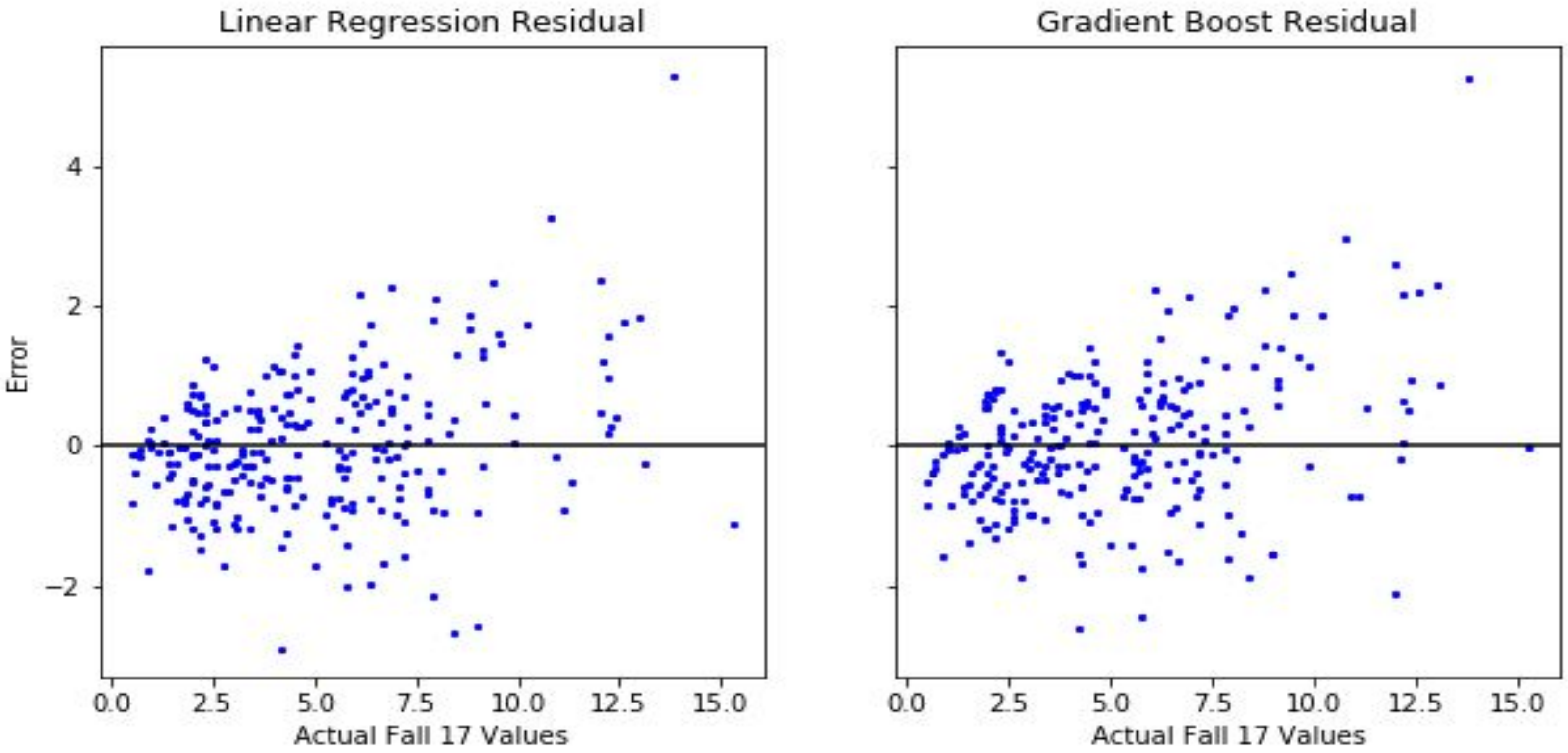
```python
# OHE categorical variables: Format, top 20 Owners, State
# get_dummies Format, State
df_feat = pd.get_dummies(df_feat, columns = ['Format','State'])
# top 20 Owners
top_owners = list(df_feat['Owner'].value_counts()[:20].index)
for owner in top_owners:
    df_feat["Owner_" + owner] = (df_feat['Owner'] == owner).astype(int)
df_feat.drop('Owner',axis = 1, inplace = True)
```

Model Results

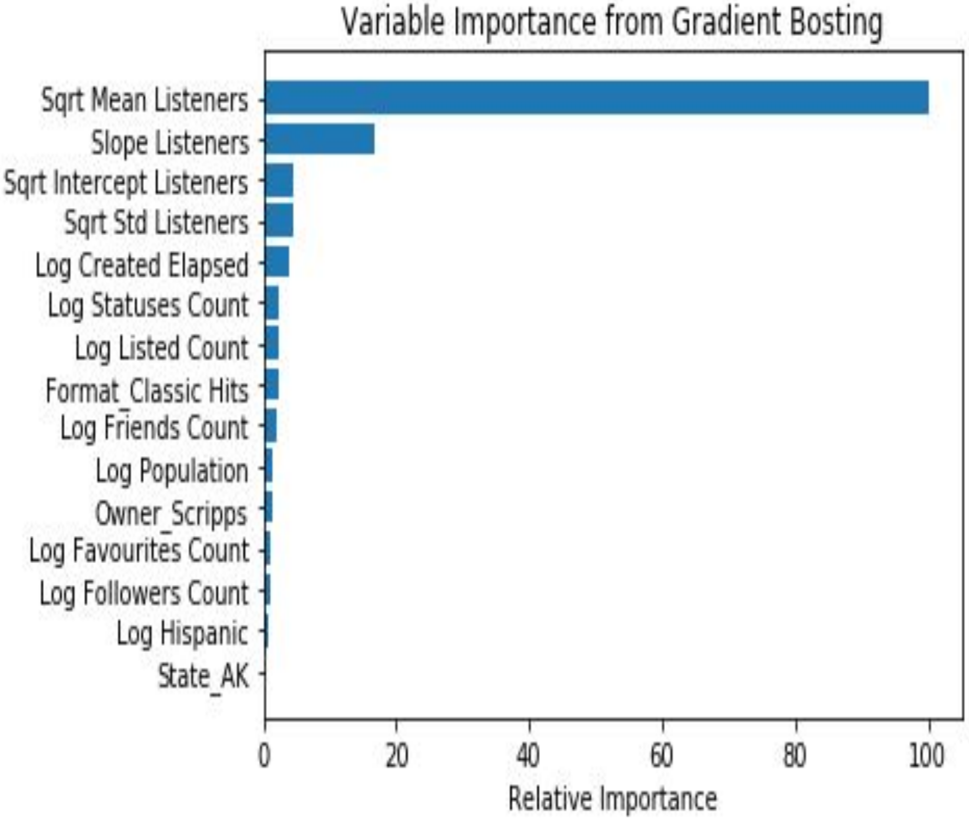
MAE Score = Mean Absolute Error
R-squared = measure of explained variance
Explained Variance = measure of explained variance (if not equal to R-squared then may indicate biased error)

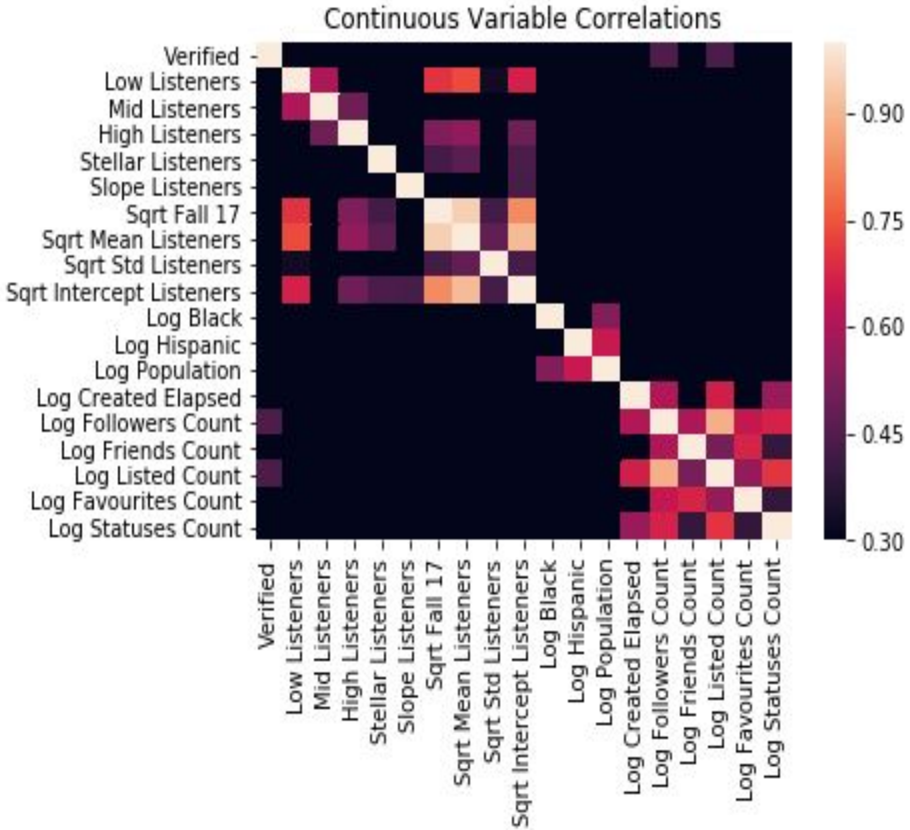
	Baseline	Linear Regression	Random Forest	Gradient Boost
MAE Score	0.782661	0.743576	0.767419	0.749029
R-squared	0.889993	0.898139	0.891722	0.895367
Explained Variance	0.891066	0.898139	0.891722	0.895368

Model Results



Model Results





Further Research

- More Data
- Perform A/B test
    - Step 1: steady state (A/A test)
    - Step 2: make a twitter account
    - Step 3: Profit.
- Analyze using time-series concepts

Acknowledgements

Data Provider:	Nielsen
Mentor:	Dave Samuel
Program Manager:	Matt Shull
Presentation Reviewer:	Demaris and Jerry Brown
Sanity Checker:	Lucia Zhu
Therapy Dog:	Lainey Lou Brown
DJ:	Ultra 2018 Festival Mix
Bed-time Storyteller:	PEP 8
Motivator:	Rent
Tailor:	H&M
Heart Attack Inducer:	Ubuntu 17.10
Main Squeeze:	Ubuntu 16.04 LTS
Lord and Savior:	Grae