Predicting Movie Total Gross after Opening Weekend

Project Luthor Exploration
Subramanian Iyer

Data

Opening Gross

Number of Opening Theaters

Run Time

Average Opening Gross/Theater

Budget

Release Date

Distributor

Genre

Rating

Actor

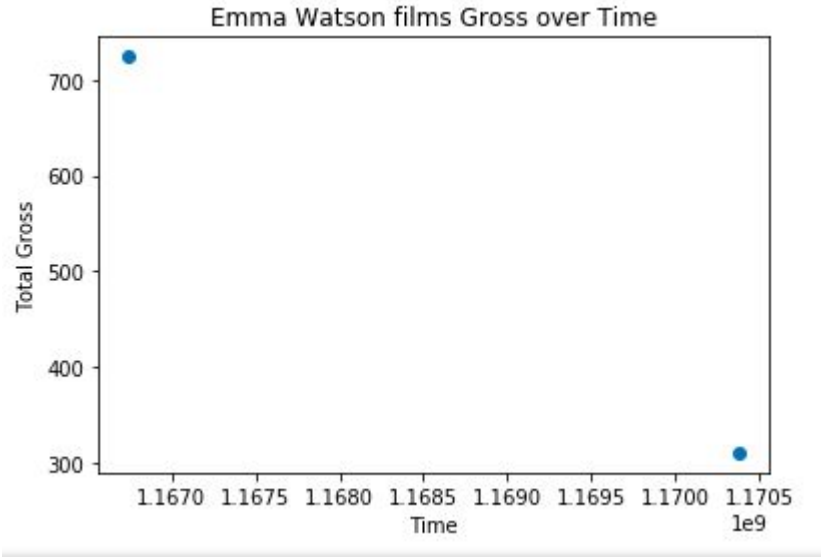
Composer

Director

Writer

Discussion on Dropping

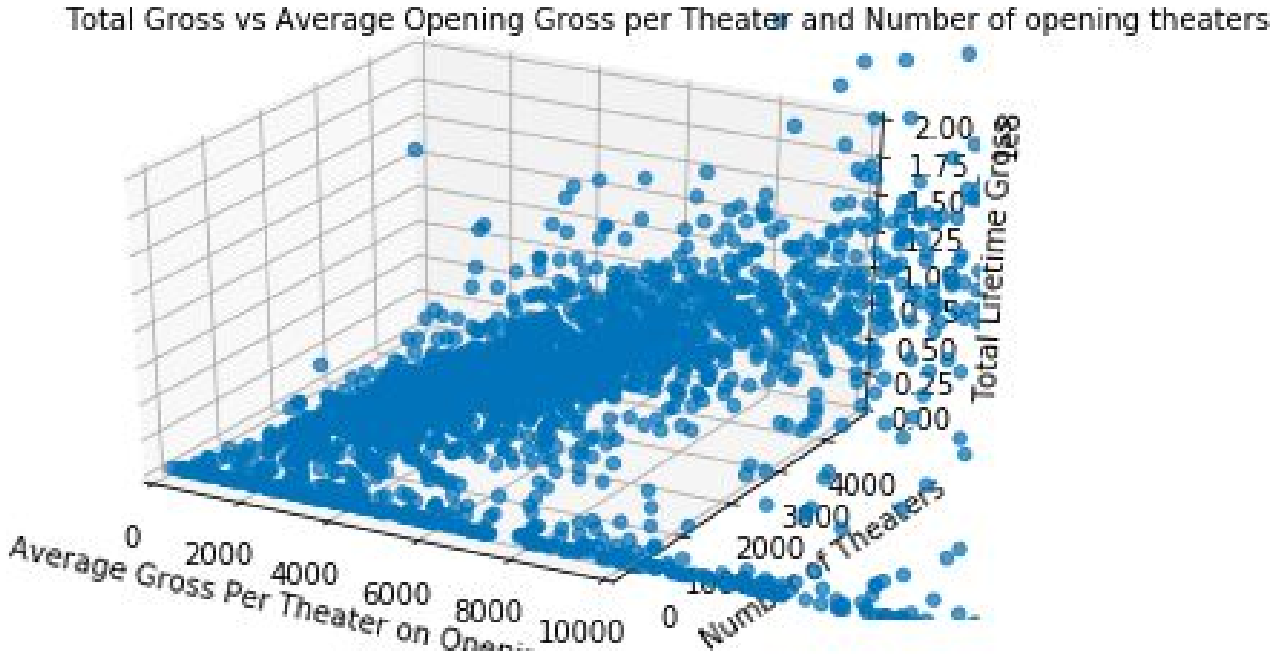
Distributor -170
Genre - 64
Rating - 7
Actor - 790
Composer - 147
Director - 735
Writer - 510



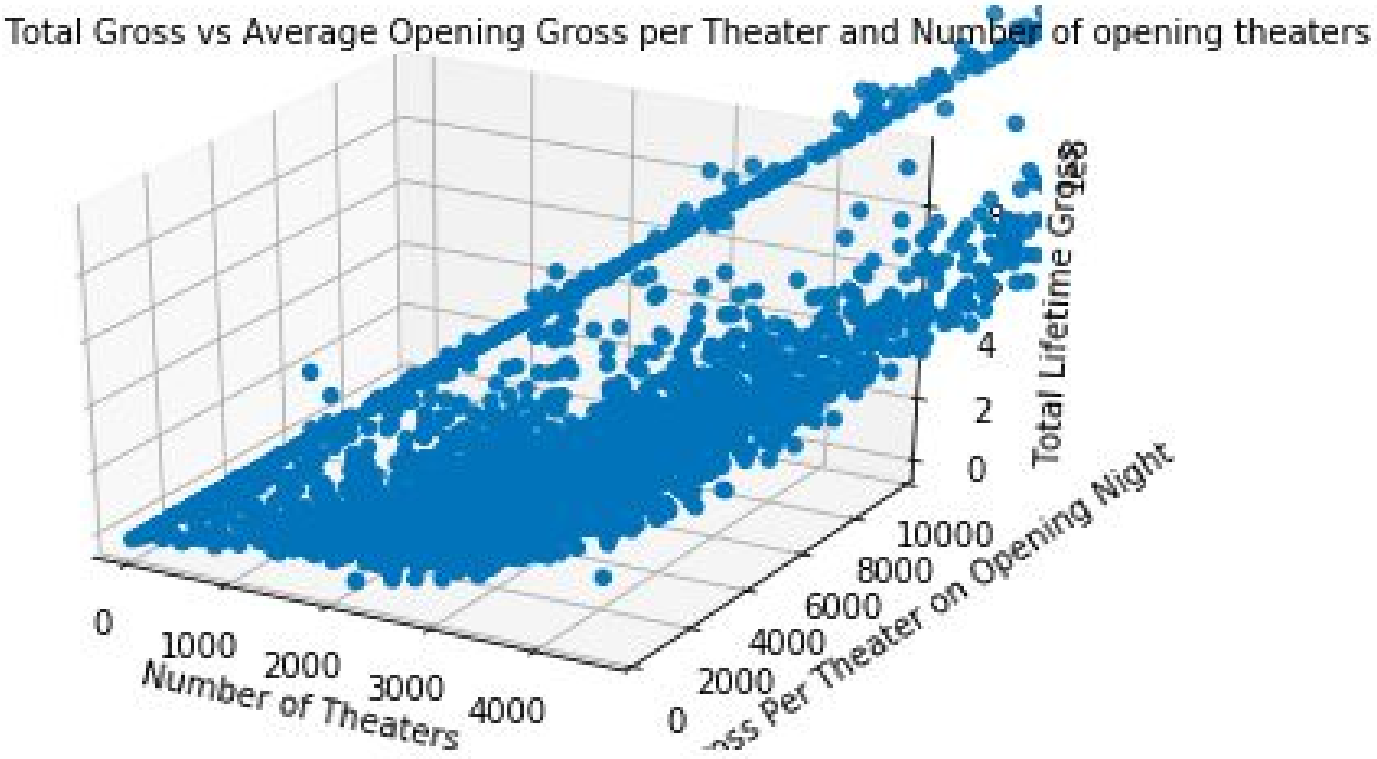
Rating Stats

Rating	count	mean	std
G	65	9.296e+07	8.683e+07
NC-17	4	7.734e+06	8.638e+06
Not Yet Rated	2	2.084e+05	2.181e+05
PG	419	8.317e+07	8.644e+07
PG-13	1038	7.618e+07	9.263e+07
R	1168	3.912e+07	4.597e+07
Unrated	48	1.226e+06	3.314e+06

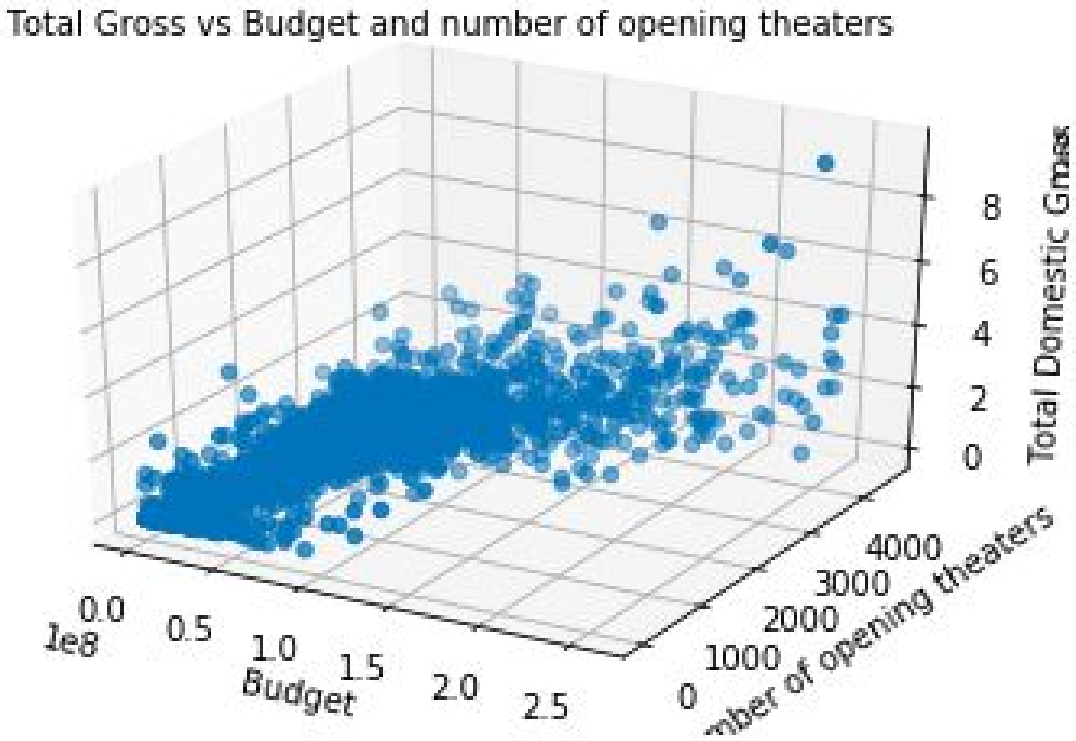
Interactions!



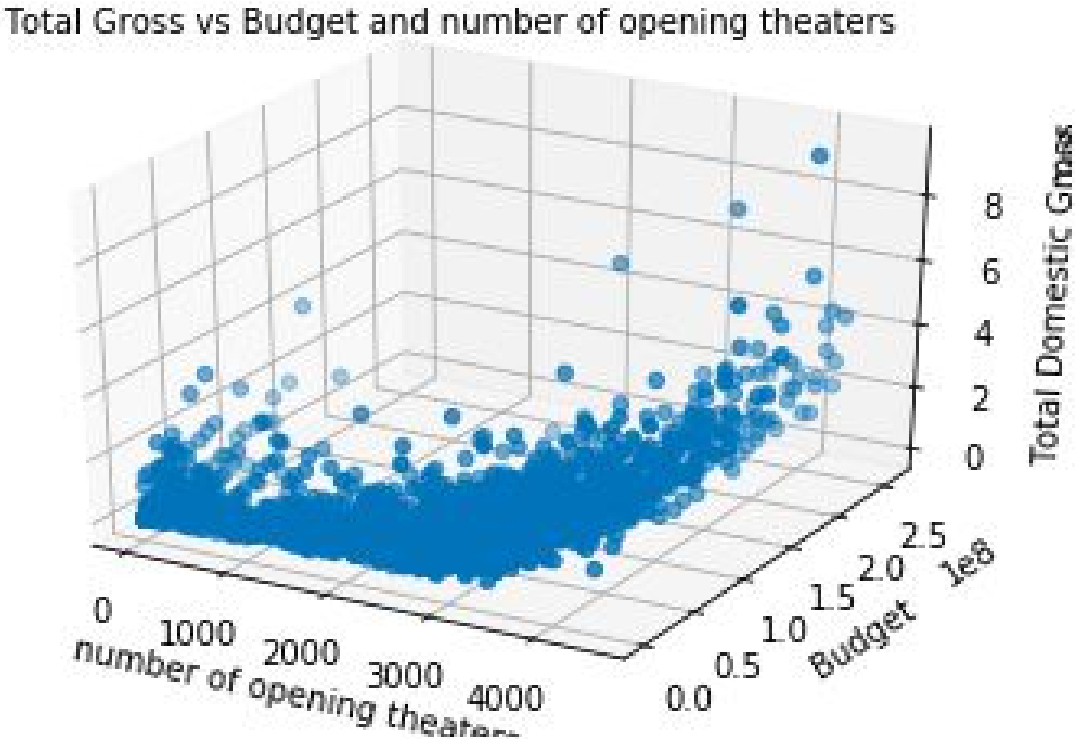
Different Angle



Interactions cont

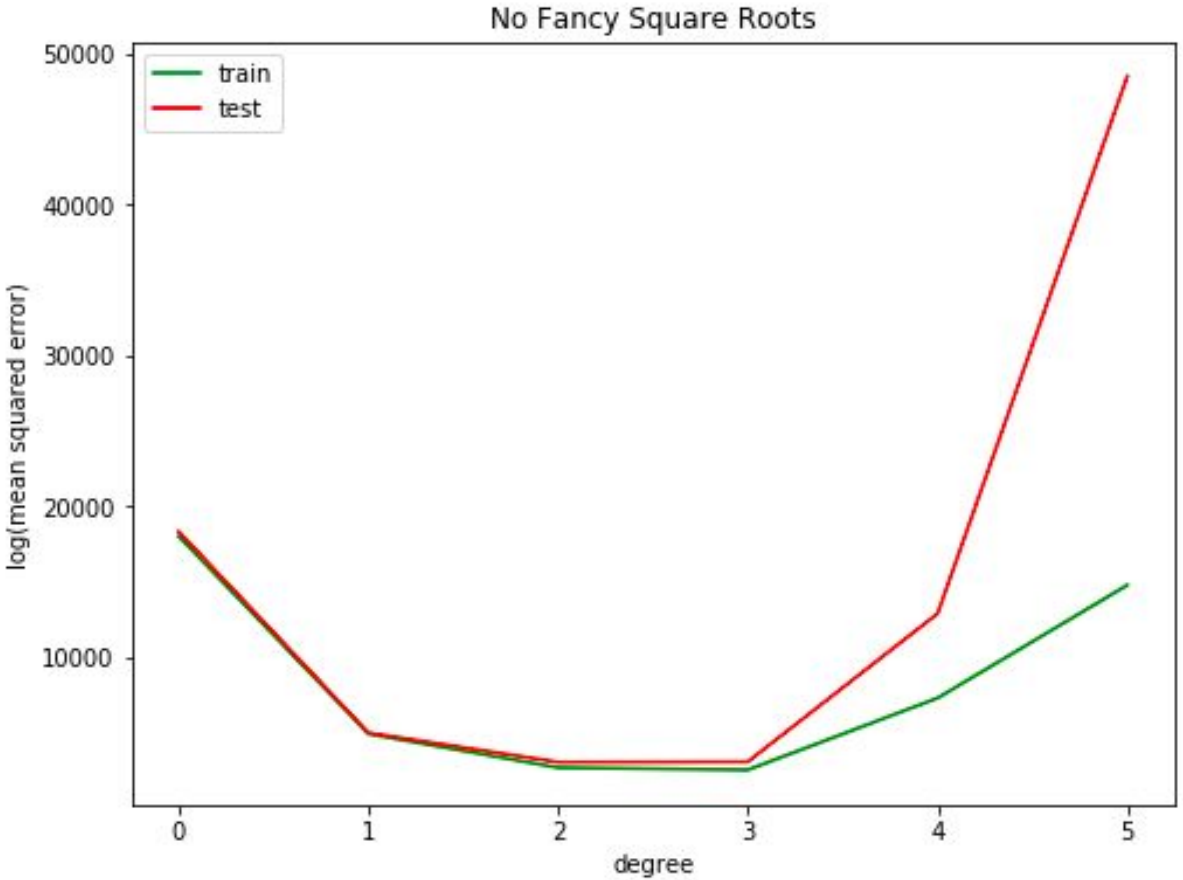


Different Angle

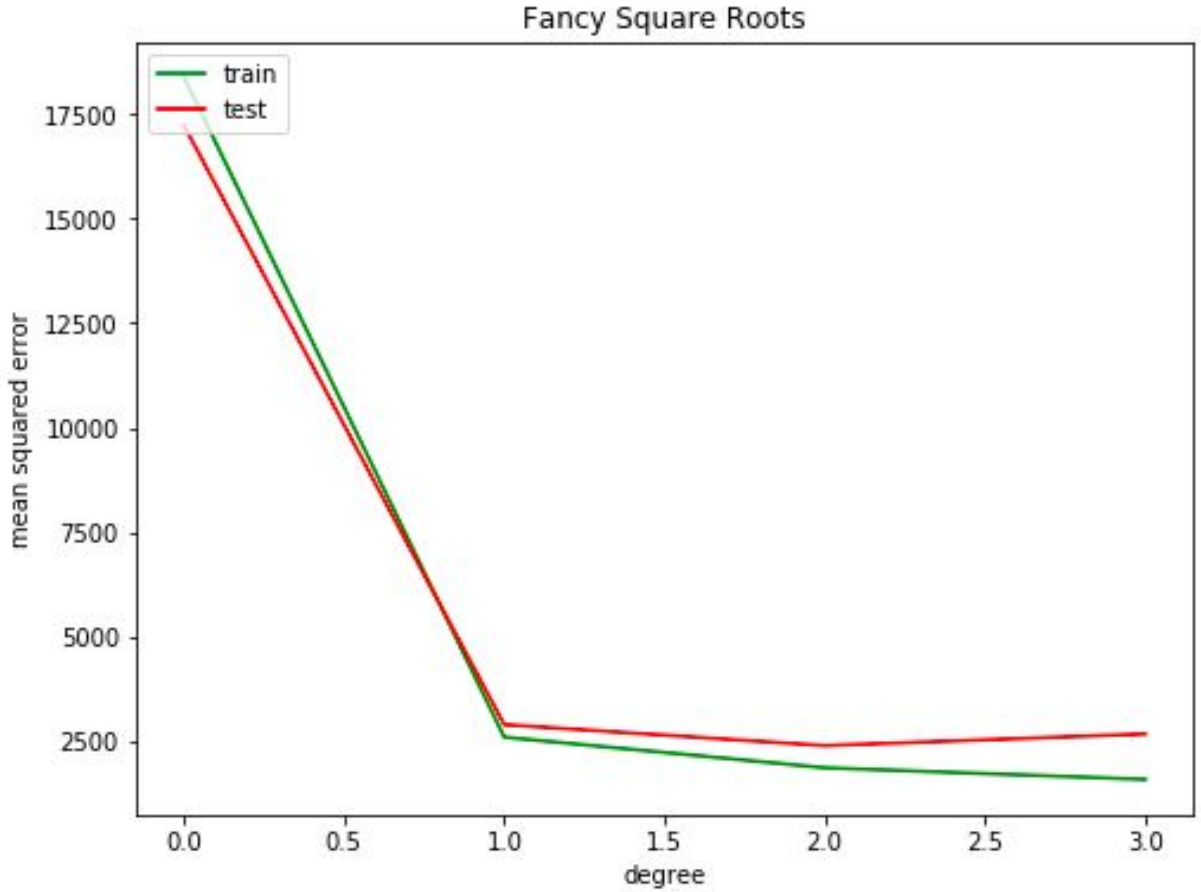


Roots!

3059


No Fancy Square Roots
train
test
log(mean squared error)
50000
40000
30000
20000
10000
0
1
2
3
4
5
degree

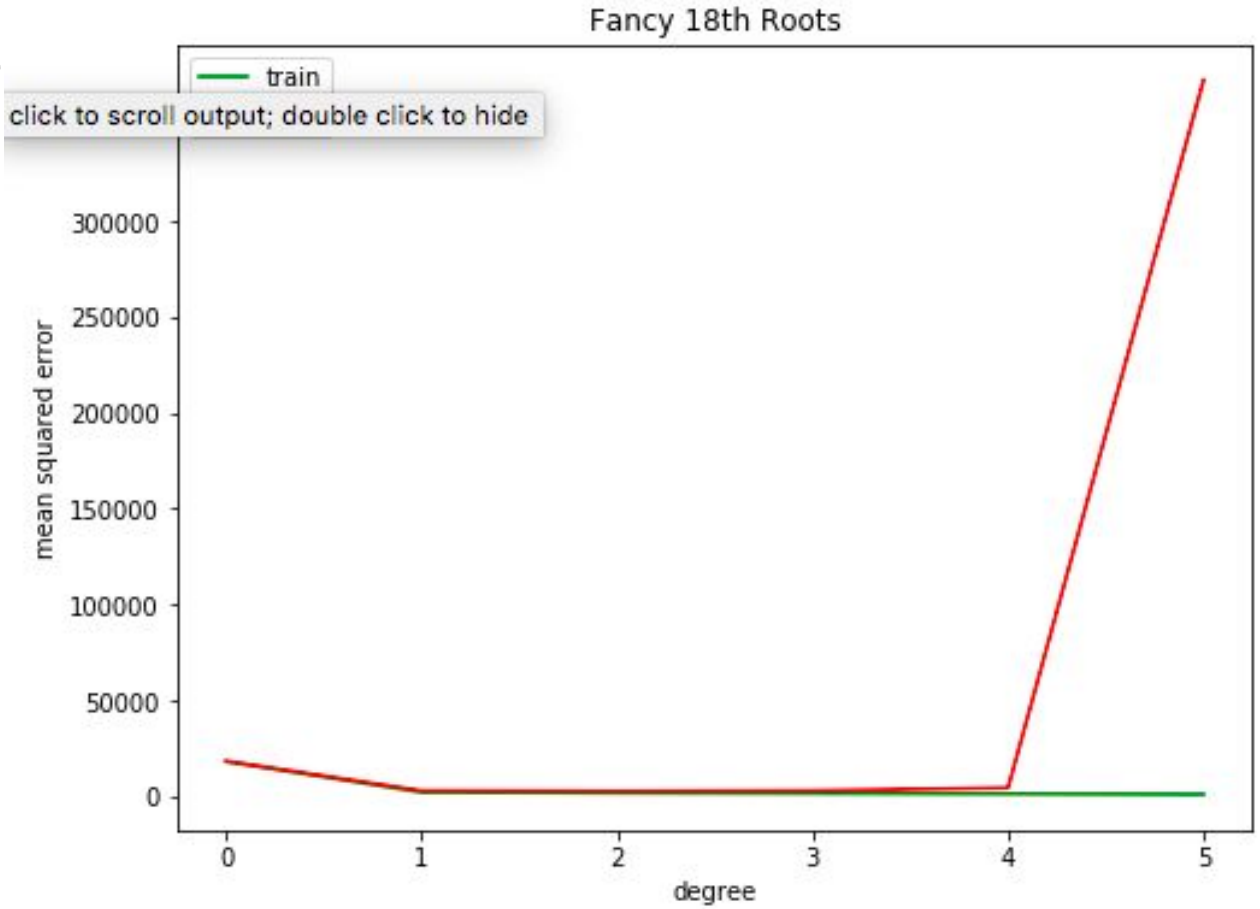
2389


Fancy Square Roots
train
test
17500
15000
12500
10000
7500
5000
2500
mean squared error
0.0
0.5
1.0
1.5
2.0
2.5
3.0
degree

Final Choice

2504

2/18



OLS

-Degrees 1,2,3

-50 trials

-cross_eval score r^2, mse, mae

Best r^2: 0.8901418

Best MSE: 0.3631699

Best MAE: 0.3800509

Best Degree: 1

Ridge

Degrees 1,2,3

15 Trials

Same cross eval deal

Best r^2: 0.892322

Best MSE: 0.356912

Best MAE: 0.377386

Degree 2

Lasso

Degrees 1,2,3,4,5,6

Started at 50, eventually only 1 trial feasible

r^2: 0.876773

MSE: 0.409599

MAE: 0.416719

Degree 6

MIGHT have actually been the best

Elastic Net

Degrees 1,2,3,4,5,6

Started at 50, eventually only 1 trial feasible

r^2: 0.890463

MSE: 0.362286

MAE: 0.381408

Degree 6

MIGHT have actually been the best

Final Model

Features -> ^(1/18)

Dependent -> boxCoxed

Model -> Ridge

Degree -> 2

Performance

r^2, MSE, MAE values before not related to interpretable data

Test/Train Split, built Ridge Model with ^2/18 features, unboxcoxed the predictions to find these values to estimate real world performance:

r^2: 0.76156

MSE: 1.5157 quadrillion dollars squared

MAE: 17.008 million dollars

(Mean TDG in data set was 60.2573 million dollars, with std of 76.8676 million dollars)

Baseline

Vanilla Regression: TDG vs OG

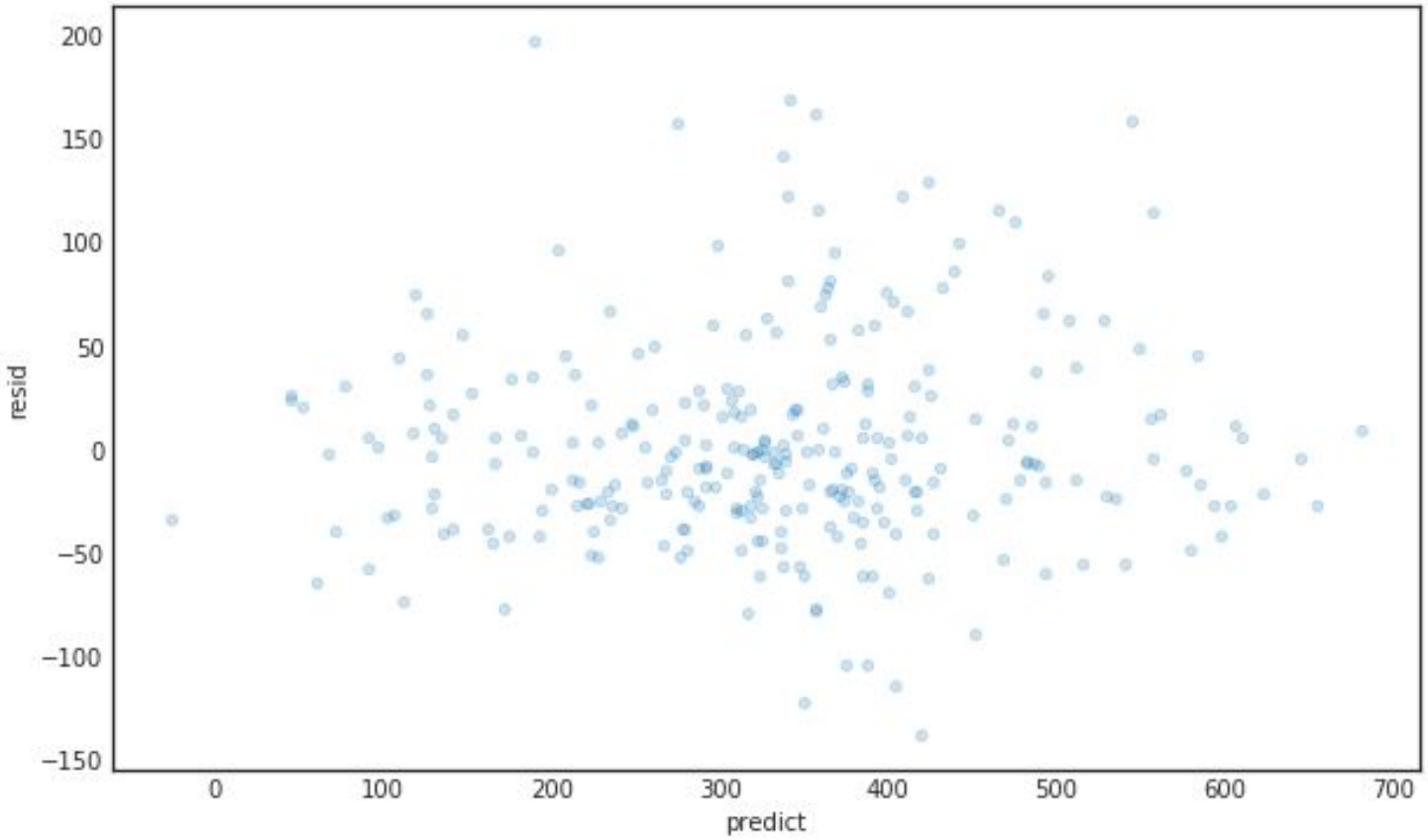
Test r^2: -0.72616

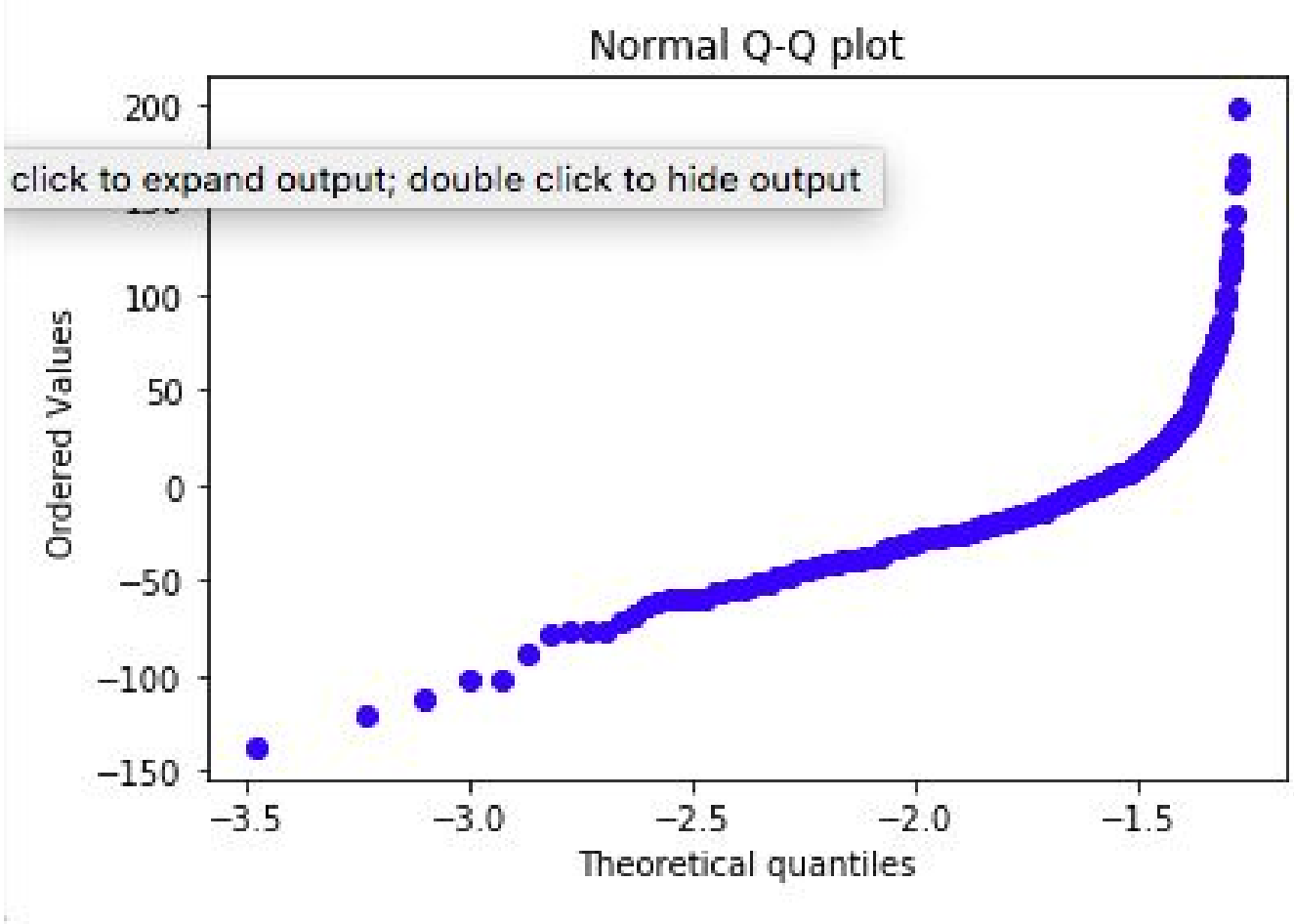
MSE: 1.362196 quadrillion dollars squared

MAE: 20.5 million dollars

Average error is higher, but less extreme mistakes, and of course correlation isawful.

Residual Plot




Normal Q-Q plot
click to expand output; double click to hide output
200
100
50
0
−50
−100
−150
Ordered Values
−3.5
−3.0
−2.5
−2.0
−1.5
Theoretical quantiles

Warnings

Don’t use this when planning production, no causation should be inferred

Potential use: deciding whether to sell the rights and for how much after opening weekend

Potential reworks

-More rigorous process to decide degree

-More trials to pick model (VERY slow, so need lots of time)

-More computing time- explore lasso and elastic net further

-Perhaps can use classification techniques with regards to categorical variables that were dropped

Questions!

```
#OLS
MSEscores = []
Rscores = []
MAEscores = []
trials = 50
for degree in range(1,4):
    MSEscore = 0
    Rscore = 0
    MAEscore = 0
    for i in range(trials):
        est = make_pipeline(PolynomialFeatures(degree), LinearRegression())
        MSEscore +=np.mean(-cross_val_score(est, X, y, cv=10, scoring='mean_squared_error'))
        Rscore += np.mean(cross_val_score(est, X, y, cv = 10, scoring = 'r2'))
        MAEscore += np.mean(-cross_val_score(est, X, y, cv = 10, scoring = 'mean_absolute_error'))
    MSEscore /= trials
    Rscore /= trials
    MAEscore /=trials
    MSEscores.append(MSEscore)
    Rscores.append(Rscore)
    MAEscores.append(MAEscore)

print(MSEscores)
print(Rscores)
print(MAEscores)
```

```
alphs = [1 * 10**e for e in range(-8,3)]
MSEscores = []
Rscores = []
MAEscores = []
trials = 15
for degree in range(1,4):
    MSEscore = 0
    Rscore = 0
    MAEscore = 0
    for i in range(trials):
        print(i)
        est = make_pipeline(PolynomialFeatures(degree), RidgeCV(alphas = alphs, cv = 10))
        MSEscore +=np.mean(-cross_val_score(est, X, y, cv=10, scoring='mean_squared_error'))
        Rscore += np.mean(cross_val_score(est, X, y, cv = 10, scoring = 'r2'))
        MAEscore += np.mean(-cross_val_score(est, X, y, cv = 10, scoring = 'mean_absolute_error'))
    MSEscore /= trials
    Rscore /= trials
    MAEscore /=trials
    MSEscores.append(MSEscore)
    Rscores.append(Rscore)
    MAEscores.append(MAEscore)

print(MSEscores)
print(Rscores)
print(MAEscores)
```