Game Popularity: Predicting Peak Concurrent Users for Steam Games with Linear Regression

Jack Etheredge
04-27-2018

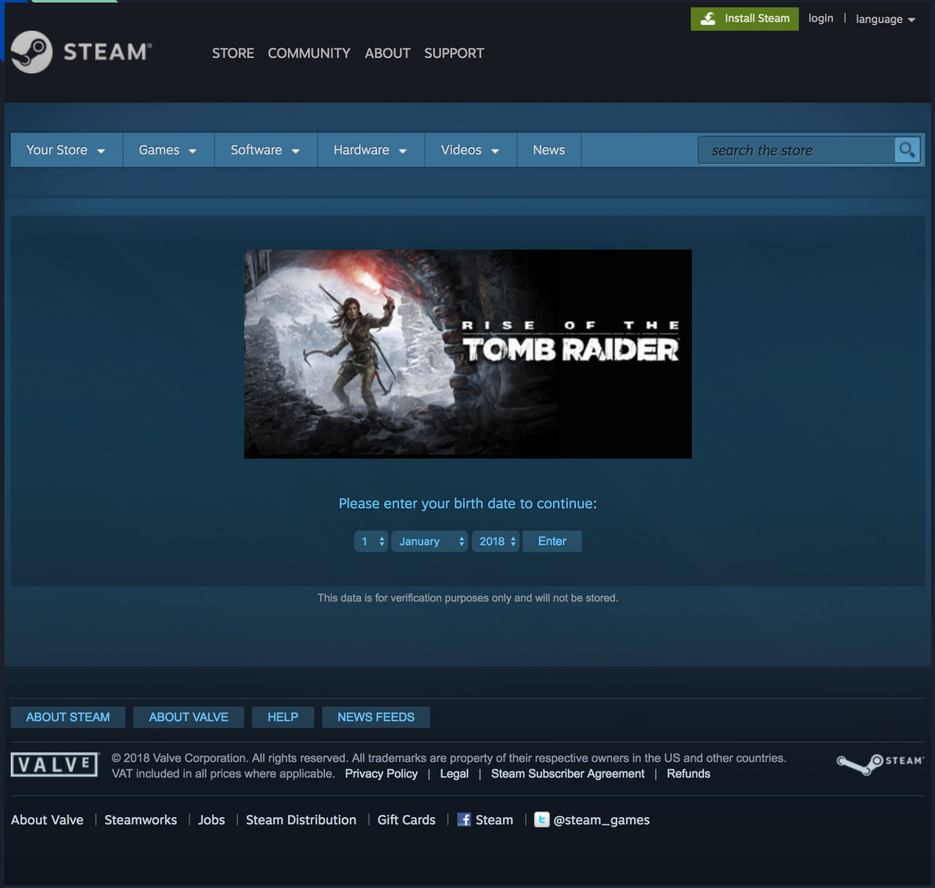
Steam: an online videogame store

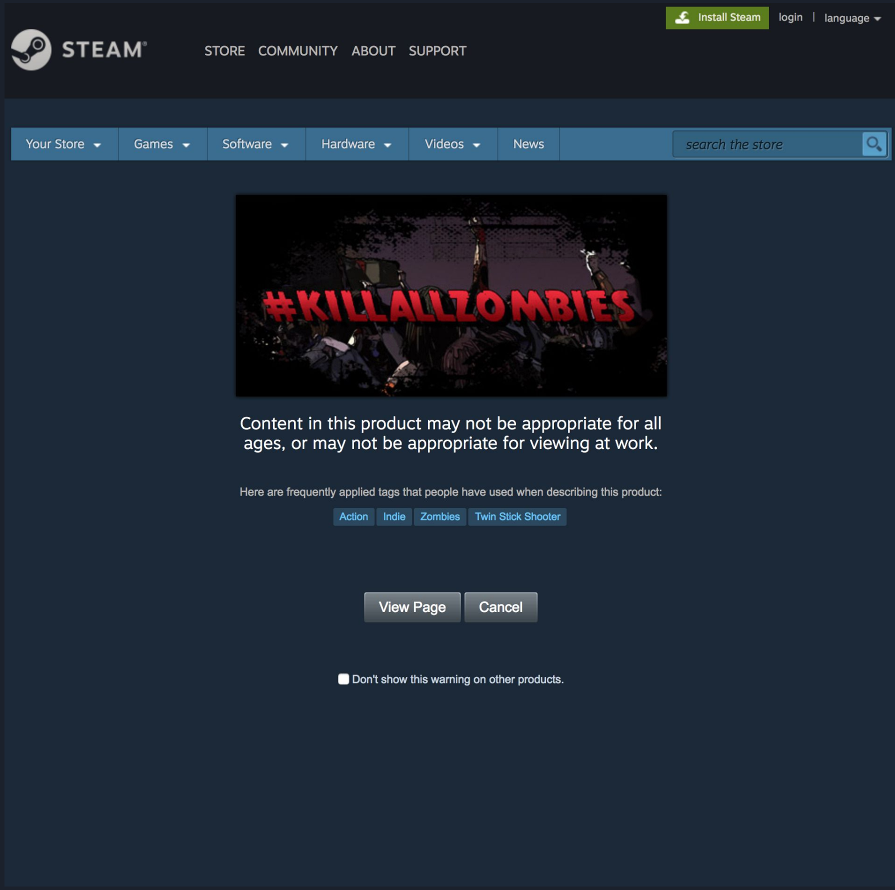
Steam is an online videogame store.

Some data you can gather:

- Price
- Percentage of positive user reviews
- Number of user reviews
- User-defined tags

Hurdles: Including different form screens





Wanted to predict ownership, but couldn’t get that value

Wanted to predict ownership, but couldn't get that value

But stubbornness prevailed.

Let's find additional sources.

Steam vs SteamDB

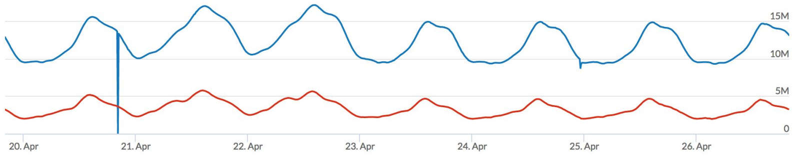
Two sources: fuzzy matching of names

Apps ▾ Packages ▾ Sales Calculator Patch notes Stats ▾ Search... Sign In

Steam Online Charts

All

Concurrent Steam Users



Top games by current player count

Compare 0 Click on plus symbol on apps you want to compare

Show All entries. Hold Shift to sort multiple columns. Search:

	APPID	NAME	CURRENT	24H PEAK	ALL-TIME PEAK
+	753	Steam	13,075,055	14,624,869	18,537,490
+	578080	PLAYERUNKNOWN'S BATTLEGROUNDS	594,535	1,857,855	3,257,248
+	570	Dota 2	379,676	614,264	1,295,114
+	730	Counter-Strike: Global Offensive	379,358	429,449	854,801
+	218620	PAYDAY 2	60,309	72,080	247,709
+	359550	Tom Clancy's Rainbow Six Siege	52,915	94,136	178,953
+	230410	Warframe	51,154	67,252	121,862
+	624090	Football Manager 2018	49,129	52,403	71,575
+	252950	Rocket League	46,073	46,440	103,129
+	440	Team Fortress 2	40,490	42,764	120,436
+	346110	ARK: Survival Evolved	37,150	38,720	101,035
+	271590	Grand Theft Auto V	33,579	52,308	364,548

For ~23500 games, ~8000 have concurrent user data

Retained ~7000 values of ~8000 with concurrent user data using fuzzy name matching

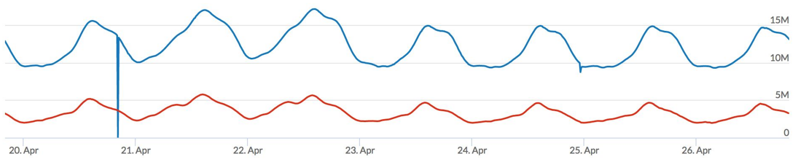
Predicting max concurrent users

Apps Packages Sales Calculator Patch notes Stats Search... Sign In

Steam Online Charts

All

Concurrent Steam Users



Top games by current player count

Compare 0 Click on plus symbol on apps you want to compare

Show All entries. Hold Shift to sort multiple columns. Search:

	APPID	NAME	CURRENT	24H PEAK	ALL-TIME PEAK
+	753	Steam	13,075,055	14,624,869	18,537,490
+	578080	PLAYERUNKNOWN'S BATTLEGROUNDS	594,535	1,857,855	3,257,248
+	570	Dota 2	379,676	614,264	1,295,114
+	730	Counter-Strike: Global Offensive	379,358	429,449	854,801
+	218620	PAYDAY 2	60,309	72,080	247,709
+	359550	Tom Clancy's Rainbow Six Siege	52,915	94,136	178,953
+	230410	Warframe	51,154	67,252	121,862
+	624090	Football Manager 2018	49,129	52,403	71,575
+	252950	Rocket League	46,073	46,440	103,129
+	440	Team Fortress 2	40,490	42,764	120,436
+	346110	ARK: Survival Evolved	37,150	38,720	101,035
+	271590	Grand Theft Auto V	33,579	52,308	364,548

Treating this as a proxy for popularity of the game

Independent variables

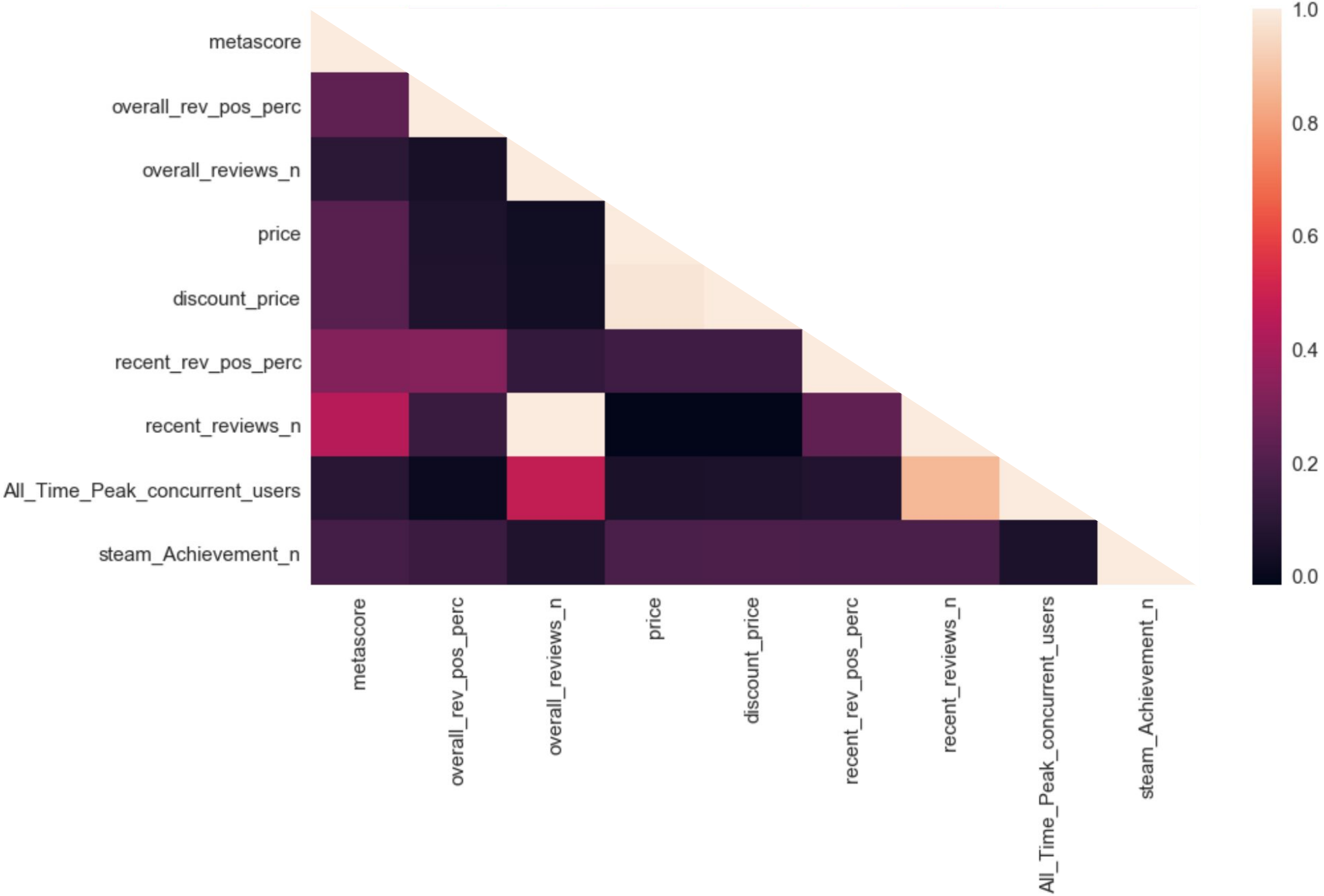
Numerical values:

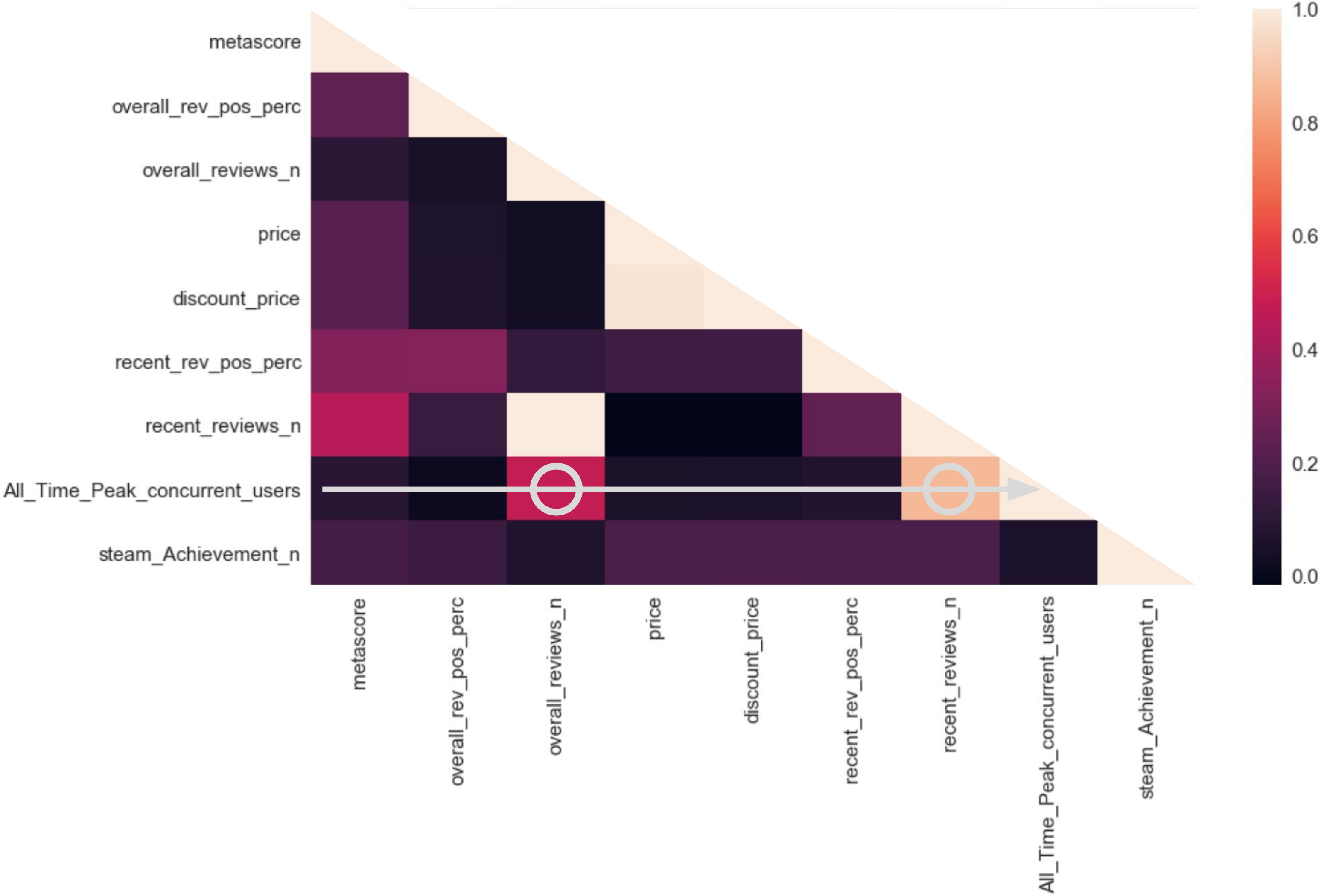
- Price
- Discounted Price
- Number of overall reviews (and Number of recent reviews)
- Percentage of positive overall reviews (and Percentage of recent positive reviews)
- Metacritic score
- Number of Steam Achievements

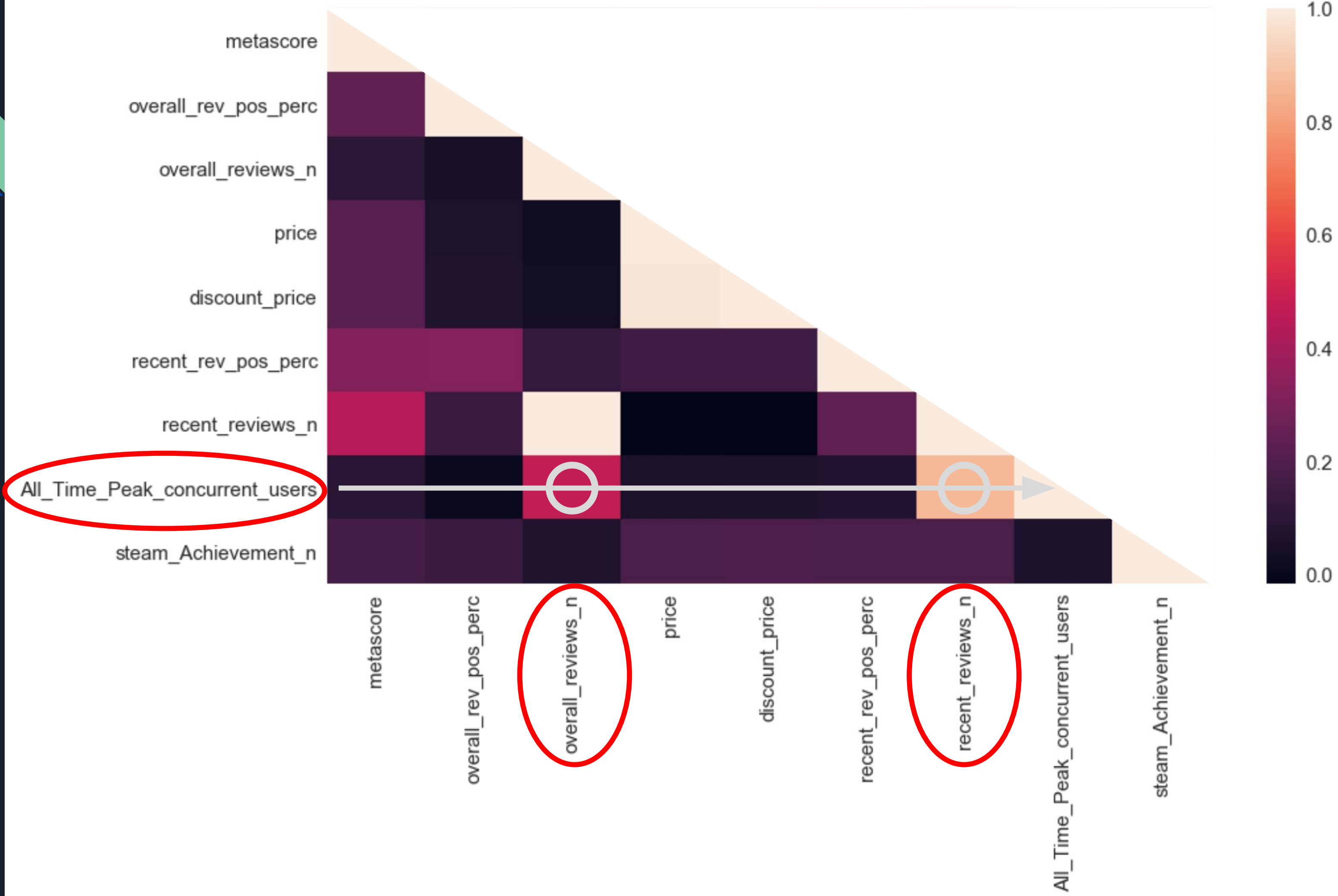
Categorical values:

- ESRB Rating
- Reasons for Rating
- Specs (multi-player, full controller support, etc)
- Genre
- User-defined tags

Release date (currently tabled)


metascore
overall_rev_pos_perc
overall_reviews_n
price
discount_price
recent_rev_pos_perc
recent_reviews_n
All_Time_Peak_concurrent_users
steam_Achievement_n
1.0
0.8
0.6
0.4
0.2
0.0


metascore
overall_rev_pos_perc
overall_reviews_n
price
discount_price
recent_rev_pos_perc
recent_reviews_n
All_Time_Peak_concurrent_users
steam_Achievement_n
1.0
0.8
0.6
0.4
0.2
0.0


metascore
overall_rev_pos_perc
overall_reviews_n
price
discount_price
recent_rev_pos_perc
recent_reviews_n
All_Time_Peak_concurrent_users
steam_Achievement_n
1.0
0.8
0.6
0.4
0.2
0.0

Dep. Variable:	All_Time_Peak_concurrent_users	**R-squared:**	0.377
Model:	OLS	**Adj. R-squared:**	0.376
Method:	Least Squares	**F-statistic:**	354.7
Date:	Tue, 24 Apr 2018	**Prob (F-statistic):**	0.00
Time:	21:23:04	**Log-Likelihood:**	-77030.
No. Observations:	6454	**AIC:**	1.541e+05
Df Residuals:	6442	**BIC:**	1.542e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	1704.9610	1.66e+04	0.103	0.918	-3.07e+04	3.42e+04
early_access[T.True]	-559.0148	1645.771	-0.340	0.734	-3785.272	2667.242
esrb[T.e]	-134.3428	1.66e+04	-0.008	0.994	-3.27e+04	3.24e+04
esrb[T.m]	1276.6759	1.66e+04	0.077	0.939	-3.13e+04	3.39e+04
esrb[T.nr]	60.6402	1.65e+04	0.004	0.997	-3.24e+04	3.25e+04
esrb[T.r]	2615.2252	1.81e+04	0.145	0.885	-3.28e+04	3.8e+04
esrb[T.t]	4641.9969	1.66e+04	0.280	0.780	-2.79e+04	3.72e+04
overall_reviews_n	0.9711	0.016	61.563	0.000	0.940	1.002
price	-7.2259	174.066	-0.042	0.967	-348.454	334.002
overall_rev_pos_perc	-42.9889	18.244	-2.356	0.018	-78.753	-7.225
discount_price	89.4572	175.799	0.509	0.611	-255.167	434.082
metascore	24.5742	15.863	1.549	0.121	-6.522	55.670

Omnibus:	21069.868	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4608422136.414
Skew:	54.706	**Prob(JB):**	0.00
Kurtosis:	4141.239	**Cond. No.**	2.60e+06

Dep. Variable:	All_Time_Peak_concurrent_users	**R-squared:**	0.377
Model:	OLS	**Adj. R-squared:**	0.376
Method:	Least Squares	**F-statistic:**	354.7
Date:	Tue, 24 Apr 2018	**Prob (F-statistic):**	0.00
Time:	21:23:04	**Log-Likelihood:**	-77030.
No. Observations:	6454	**AIC:**	1.541e+05
Df Residuals:	6442	**BIC:**	1.542e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	1704.9610	1.66e+04	0.103	0.918	-3.07e+04	3.42e+04
early_access[T.True]	-559.0148	1645.771	-0.340	0.734	-3785.272	2667.242
esrb[T.e]	-134.3428	1.66e+04	-0.008	0.994	-3.27e+04	3.24e+04
esrb[T.m]	1276.6759	1.66e+04	0.077	0.939	-3.13e+04	3.39e+04
esrb[T.nr]	60.6402	1.65e+04	0.004	0.997	-3.24e+04	3.25e+04
esrb[T.r]	2615.2252	1.81e+04	0.145	0.885	-3.28e+04	3.8e+04
esrb[T.t]	4641.9969	1.66e+04	0.280	0.780	-2.79e+04	3.72e+04
overall_reviews_n	0.9711	0.016	61.563	0.000	0.940	1.002
price	-7.2259	174.066	-0.042	0.967	-348.454	334.002
overall_rev_pos_perc	-42.9889	18.244	-2.356	0.018	-78.753	-7.225
discount_price	89.4572	175.799	0.509	0.611	-255.167	434.082
metascore	24.5742	15.863	1.549	0.121	-6.522	55.670

Omnibus:	21069.868	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4608422136.414
Skew:	54.706	**Prob(JB):**	0.00
Kurtosis:	4141.239	**Cond. No.**	2.60e+06

Including a small number of “first pass” variables, R^2 is low

Dep. Variable:	All_Time_Peak_concurrent_users	**R-squared:**	0.377
Model:	OLS	**Adj. R-squared:**	0.376
Method:	Least Squares	**F-statistic:**	354.7
Date:	Tue, 24 Apr 2018	**Prob (F-statistic):**	0.00
Time:	21:23:04	**Log-Likelihood:**	-77030.
No. Observations:	6454	**AIC:**	1.541e+05
Df Residuals:	6442	**BIC:**	1.542e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	1704.9610	1.66e+04	0.103	0.918	-3.07e+04	3.42e+04
early_access[T.True]	-559.0148	1645.771	-0.340	0.734	-3785.272	2667.242
esrb[T.e]	-134.3428	1.66e+04	-0.008	0.994	-3.27e+04	3.24e+04
esrb[T.m]	1276.6759	1.66e+04	0.077	0.939	-3.13e+04	3.39e+04
esrb[T.nr]	60.6402	1.65e+04	0.004	0.997	-3.24e+04	3.25e+04
esrb[T.r]	2615.2252	1.81e+04	0.145	0.885	-3.28e+04	3.8e+04
esrb[T.t]	4641.9969	1.66e+04	0.280	0.780	-2.79e+04	3.72e+04
$\boxed{\textbf{overall\_reviews\_n}}$	0.9711	0.016	61.563	$\boxed{0.000}$	0.940	1.002
price	-7.2259	174.066	-0.042	0.967	-348.454	334.002
$\boxed{\textbf{overall\_rev\_pos\_perc}}$	-42.9889	18.244	-2.356	$\boxed{0.018}$	-78.753	-7.225
discount_price	89.4572	175.799	0.509	0.611	-255.167	434.082
$\boxed{\textbf{metascore}}$	24.5742	15.863	1.549	$\boxed{0.121}$	-6.522	55.670

Omnibus:	21069.868	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4608422136.414
Skew:	54.706	**Prob(JB):**	0.00
Kurtosis:	4141.239	**Cond. No.**	2.60e+06

Including a small number of “first pass” variables, R^2 is low

We already knew review number and positivity would show up here

Let’s feed in more data

Dep. Variable:	All_Time_Peak_concurrent_users	**R-squared:**	0.377
Model:	OLS	**Adj. R-squared:**	0.376
Method:	Least Squares	**F-statistic:**	354.7
Date:	Tue, 24 Apr 2018	**Prob (F-statistic):**	0.00
Time:	21:23:04	**Log-Likelihood:**	-77030.
No. Observations:	6454	**AIC:**	1.541e+05
Df Residuals:	6442	**BIC:**	1.542e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	1704.9610	1.66e+04	0.103	0.918	-3.07e+04	3.42e+04
early_access[T.True]	-559.0148	1645.771	-0.340	0.734	-3785.272	2667.242
esrb[T.e]	-134.3428	1.66e+04	-0.008	0.994	-3.27e+04	3.24e+04
esrb[T.m]	1276.6759	1.66e+04	0.077	0.939	-3.13e+04	3.39e+04
esrb[T.nr]	60.6402	1.65e+04	0.004	0.997	-3.24e+04	3.25e+04
esrb[T.r]	2615.2252	1.81e+04	0.145	0.885	-3.28e+04	3.8e+04
esrb[T.t]	4641.9969	1.66e+04	0.280	0.780	-2.79e+04	3.72e+04
overall_reviews_n	0.9711	0.016	61.563	0.000	0.940	1.002
price	-7.2259	174.066	-0.042	0.967	-348.454	334.002
overall_rev_pos_perc	-42.9889	18.244	-2.356	0.018	-78.753	-7.225
discount_price	89.4572	175.799	0.509	0.611	-255.167	434.082
metascore	24.5742	15.863	1.549	0.121	-6.522	55.670

Omnibus:	21069.868	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4608422136.414
Skew:	54.706	**Prob(JB):**	0.00
Kurtosis:	4141.239	**Cond. No.**	2.60e+06

Including a small number of “first pass” variables, R^2 is low

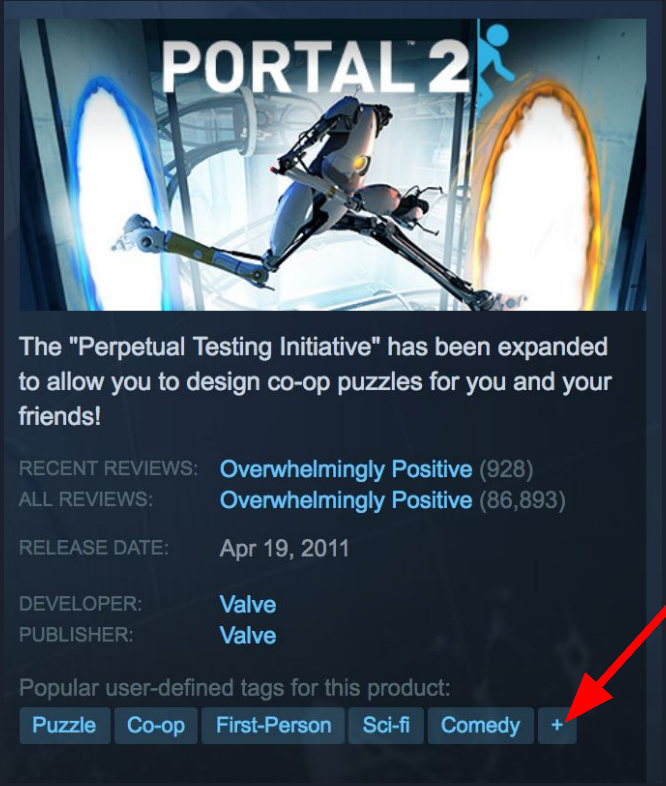
We already knew review number and positivity would show up here

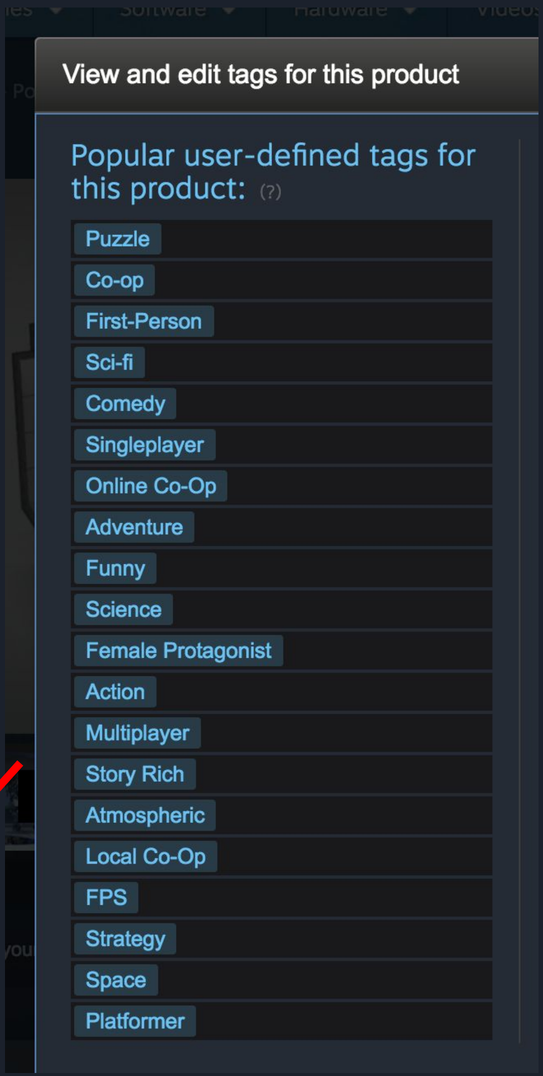
Let’s feed in more data

There are also huge problems with skew I’ll deal with later

Categorical variables

Lots of categoricals, due to several tags, genres, etc





Independent variables

Lots of categoricals, due to several tags, genres, etc all need to be “unpacked” since each game can have multiple tags, genres, specs, and even multiple developers and publishers:

Independent variables: (9888 of them!)

Lots of categoricals, due to several tags, genres, etc all need to be “unpacked” since each game can have multiple tags, genres, specs, and even multiple developers and publishers:

Independent variables: (9888 of them!)

Lots of categoricals, due to several tags, genres, etc all need to be "unpacked" since each game can have multiple tags, genres, specs, and even multiple developers and publishers:

(9888)


IT'S OVER
9000!!!!

Independent variables

Lots of categoricals, due to several tags, genres, etc

(9888 of them! -> reduced to 845 by removing features with very low counts (<10))

Scoring throughout the rest of the talk:

R-squared is test R-squared

All train and test fit and predictions are performed with 10-fold cross-validation

All the data is standardized

Select k-best features

All features (standard ordinary least squares regression):

R^2 train = 0.412, R^2 test = -1.43E25

10 features (select k-best):

R^2 train = 0.329, R^2 test = -2.11

Select k-best features

All features (standard ordinary least squares regression):

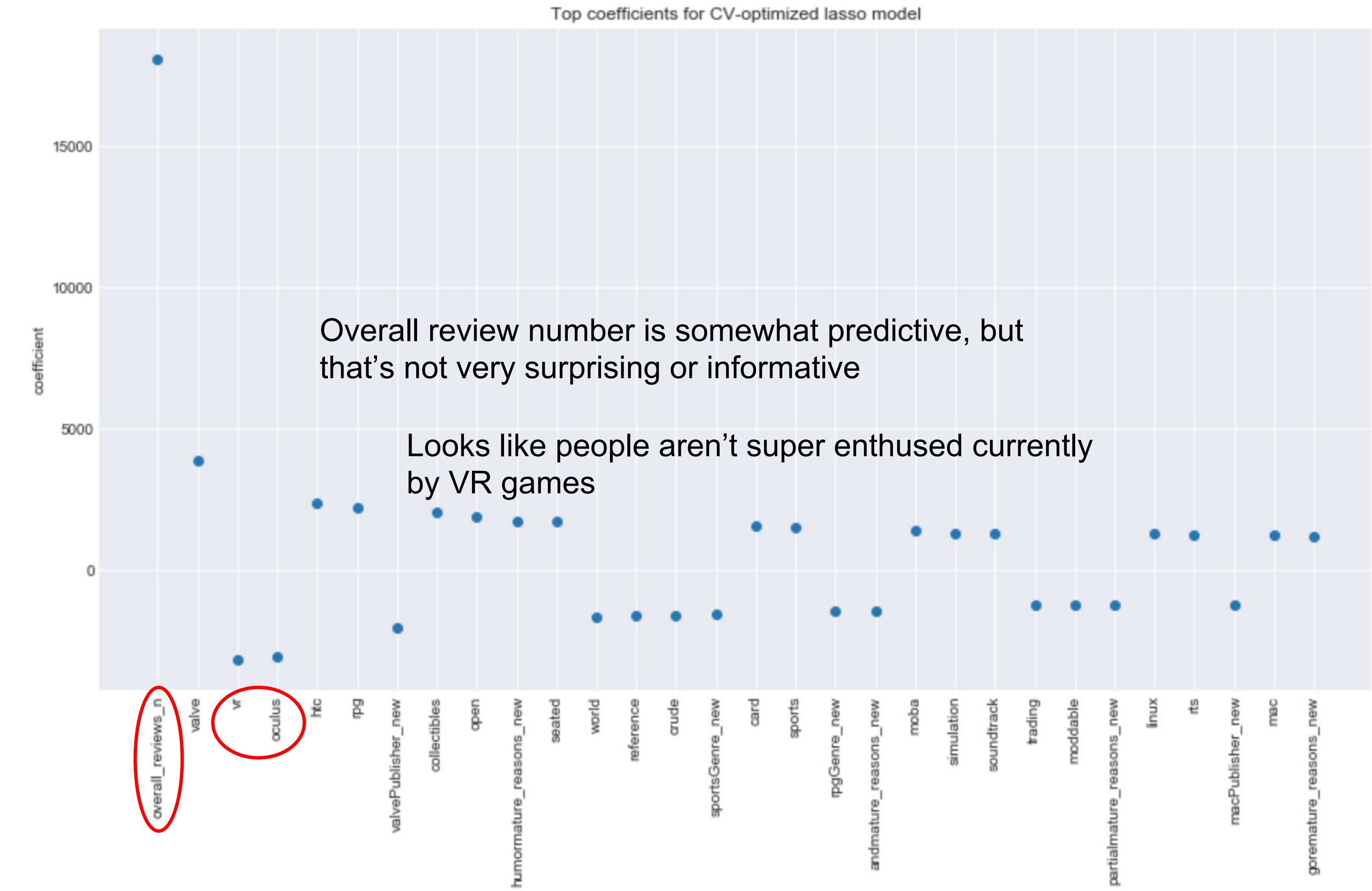
R^2 train = 0.412, R^2 test = -1.43E25

The model is overfit, performing far better on test and train, even after selecting only the 10 best features

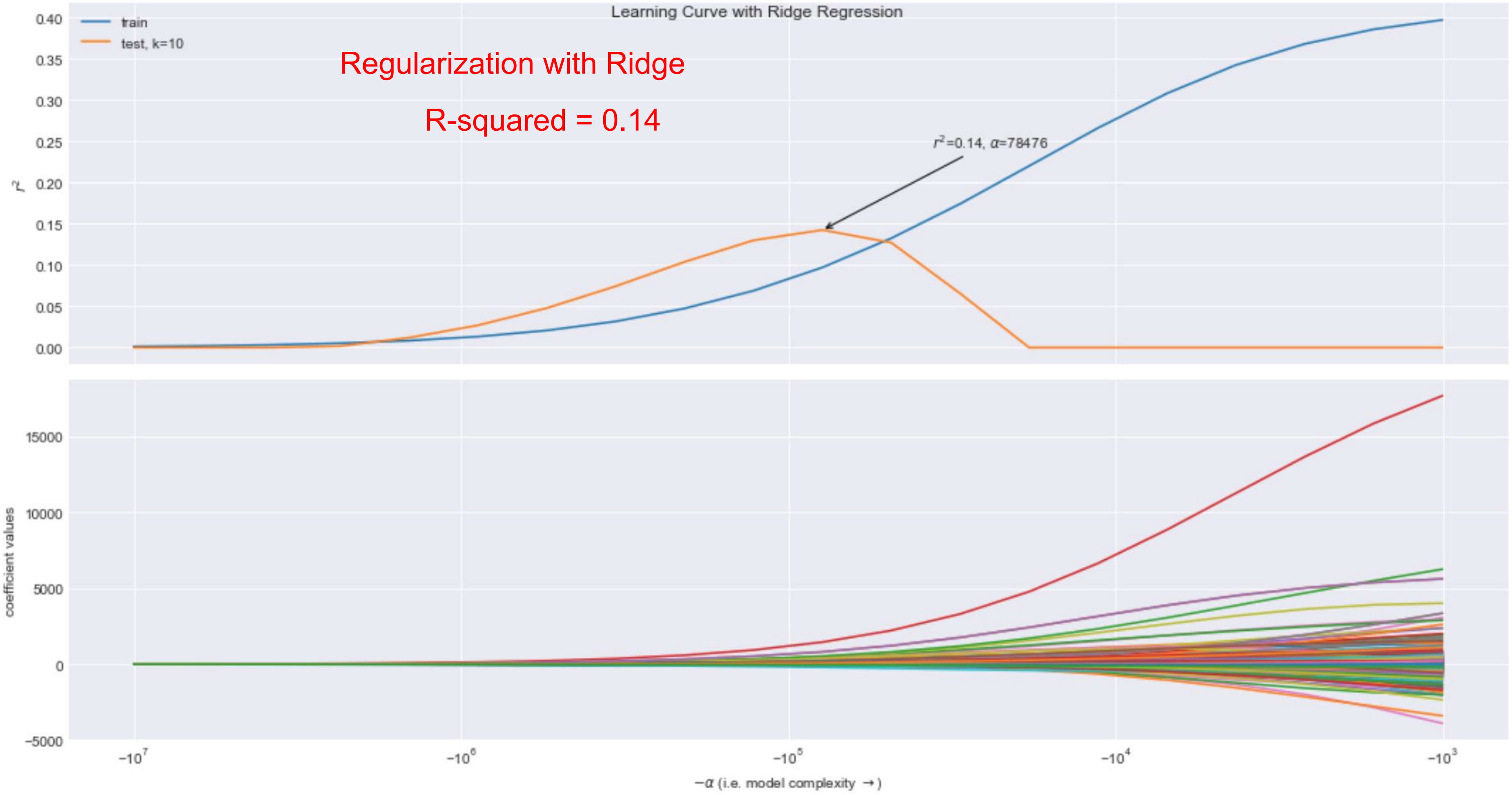
10 features (select k-best):

R^2 train = 0.329, R^2 test = -2.11

Learning Curve with Lasso Regression
train
test, k=10
Regularization with Lasso
R-squared = 0.024
r^2=0.024, α=26366.5
0.08
0.06
0.04
0.02
0.00
r^2
0.04
0.02
0.00
−0.02
−0.04
coefficient values
$-\alpha$ (i.e. model complexity →)



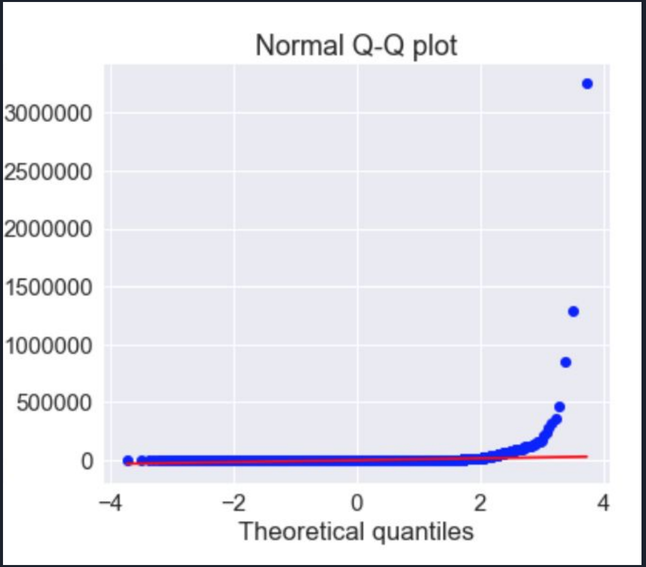
Top coefficients for CV-optimized lasso model
coefficient
15000
10000
5000
0
overall_reviews_n
valve
vr
oculus
htc
rpg
valvePublisher_new
collectibles
open
humormature_reasons_new
seated
world
reference
crude
sportsGenre_new
card
sports
rpgGenre_new
andmature_reasons_new
moba
simulation
soundtrack
trading
moddable
partialmature_reasons_new
linux
rts
macPublisher_new
mac
gorematume_reasons_new
Overall review number is somewhat predictive, but that's not very surprising or informative
Looks like people aren't super enthused currently by VR games



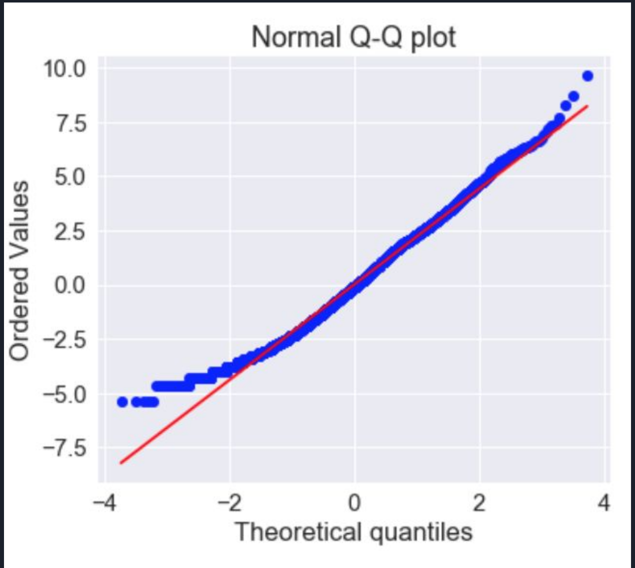
Learning Curve with Ridge Regression
train
test, k=10
Regularization with Ridge
R-squared = 0.14
r^2=0.14, α=78476
r^2
coefficient values
-10^7
-10^6
-10^5
-10^4
-10^3
$-\alpha$ (i.e. model complexity →)

Perhaps I should try taking the log(y)

y



log(y)



Select k-best features

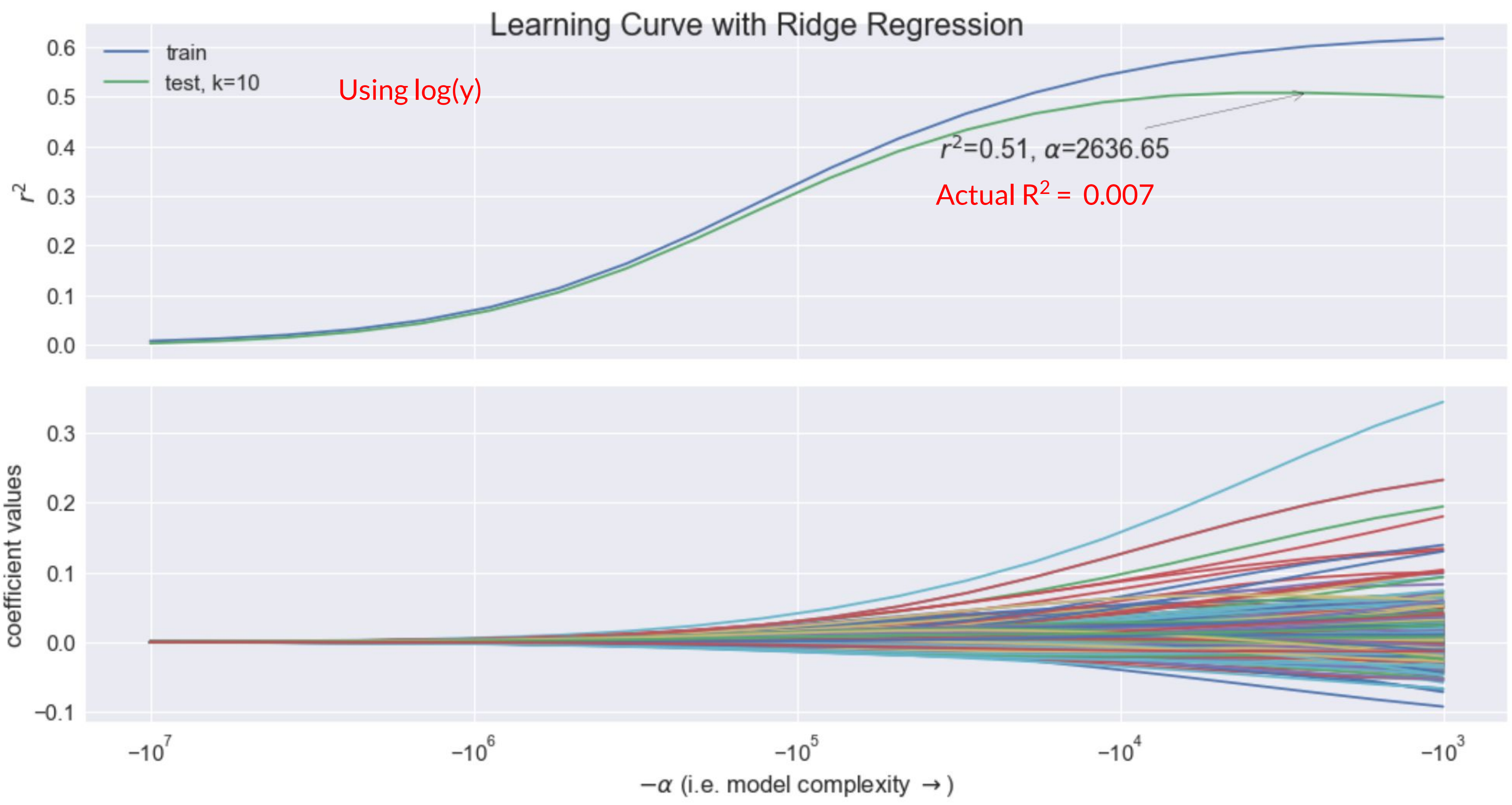
10 features (select k-best):

R^2 train = 0.329, R^2 test = -2.11

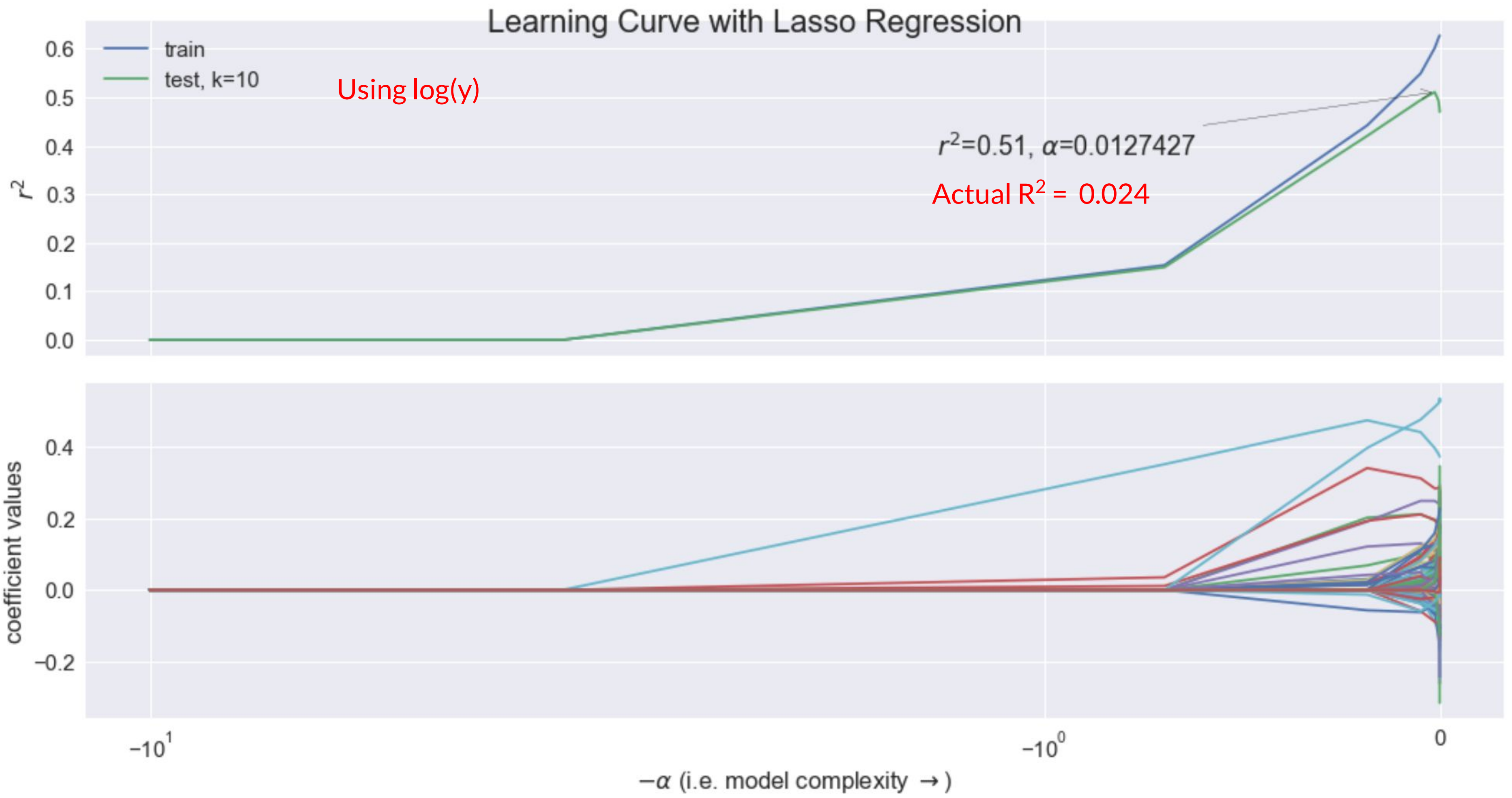
10 features (select k-best) with a log-transformed y:

Prematurely exciting R^2 train = 0.446, R^2 test = 0.442

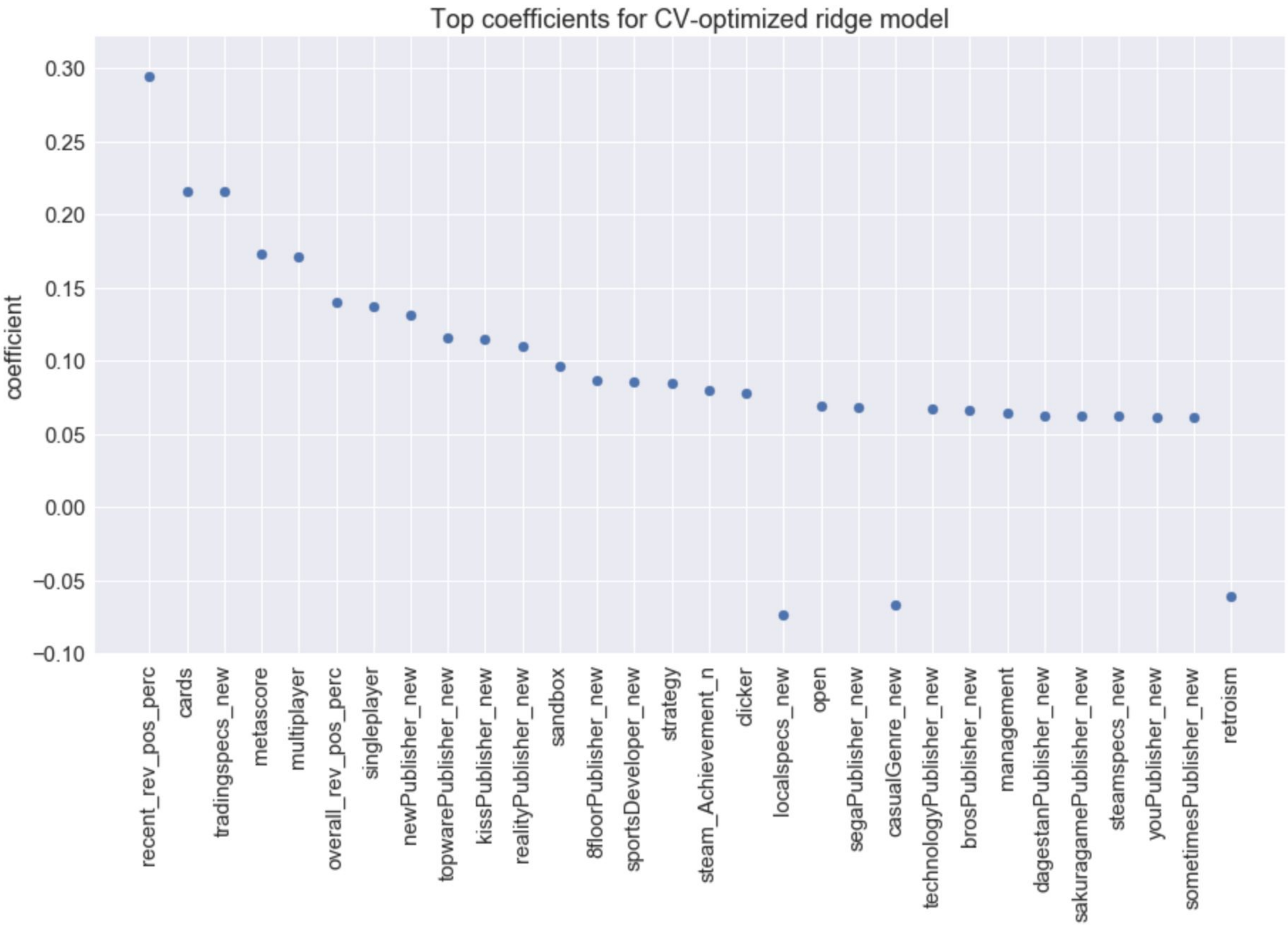
Actual (accounting for log-transform): R^2 test =0.0012



Learning Curve with Ridge Regression
train
test, k=10
Using log(y)
r^2=0.51, α=2636.65
Actual R^2 = 0.007
0.6
0.5
0.4
0.3
0.2
0.1
0.0
r^2
0.3
0.2
0.1
0.0
−0.1
coefficient values
-10^7
-10^6
-10^5
-10^4
-10^3
$-\alpha$ (i.e. model complexity →)



Learning Curve with Lasso Regression
train
test, k=10
Using log(y)
r^2=0.51, α=0.0127427
Actual R^2 = 0.024
r^2
coefficient values
-10^1
-10^0
0
$-\alpha$ (i.e. model complexity $\rightarrow$)


Top coefficients for CV-optimized ridge model
coefficient
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
recent_rev_pos_perc
cards
tradingspecs_new
metascore
multiplayer
overall_rev_pos_perc
singleplayer
newPublisher_new
topwarePublisher_new
kissPublisher_new
realityPublisher_new
sandbox
8floorPublisher_new
sportsDeveloper_new
strategy
steam_Achievement_n
dicker
localspecs_new
open
segaPublisher_new
casualGenre_new
technologyPublisher_new
brosPublisher_new
management
dagestanPublisher_new
sakuragamePublisher_new
steamspecs_new
youPublisher_new
sometimesPublisher_new
retroism

Conclusion:

The Steam data I acquired is insufficient to meaningfully predict the peak concurrent users

There is a correlation between the number of reviews and the peak concurrent users

There were some other weak predictors that were consistent across models:

“Card” and “multiplayer” tags are positively correlated

“VR” and “oculus” tags are negatively correlated

By and large, though, these models weaken our belief that there is a combination of Steam user tags that strongly predict the the peak concurrent users

Future Directions (in order of importance?):

Try again with a Poisson regression.

Future Directions (in order of importance?):

Try again with a Poisson regression.

Try predicting the number of reviews?

Try predicting the price?

Try predicting whether something will be on sale or the percentage discount?

Additional supporting data sets

Walk before I run - Learn what I can from simpler datasets that “play nice”

Keep learning how to deal with difficult datasets

Learn how to avoid (infrequent) timeout errors, possibly by not loading images and videos

To add to talk later:

Plot predicted (y_pred) vs actual (y)

X is observations (index)

Do early positive reviews predict the future popularity of a game?

- If the first few reviews are positive, does the game have more reviews or more

Thanks

Scraping example

Add gif

???

???

But...

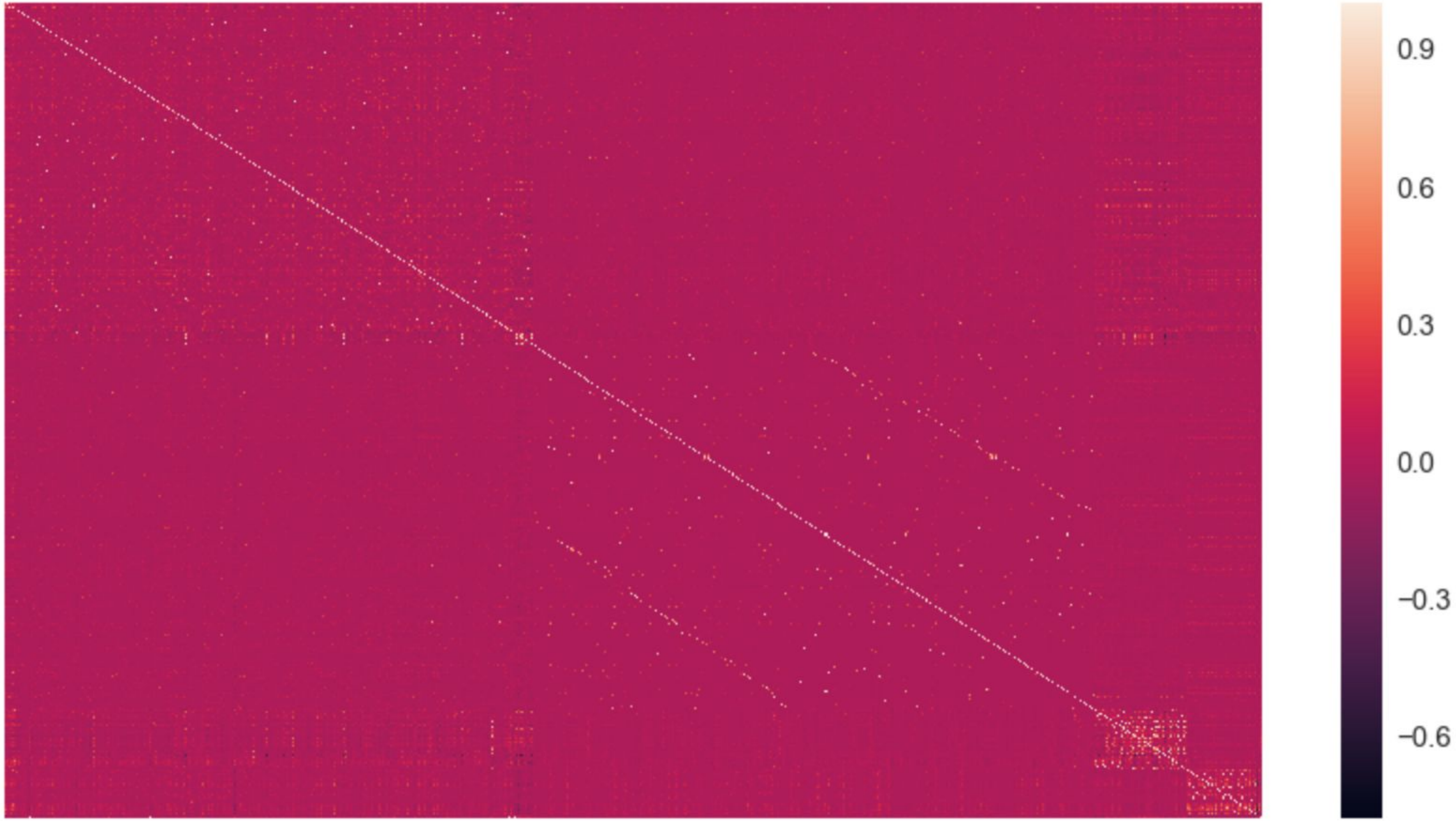
R-squared accounting for log:

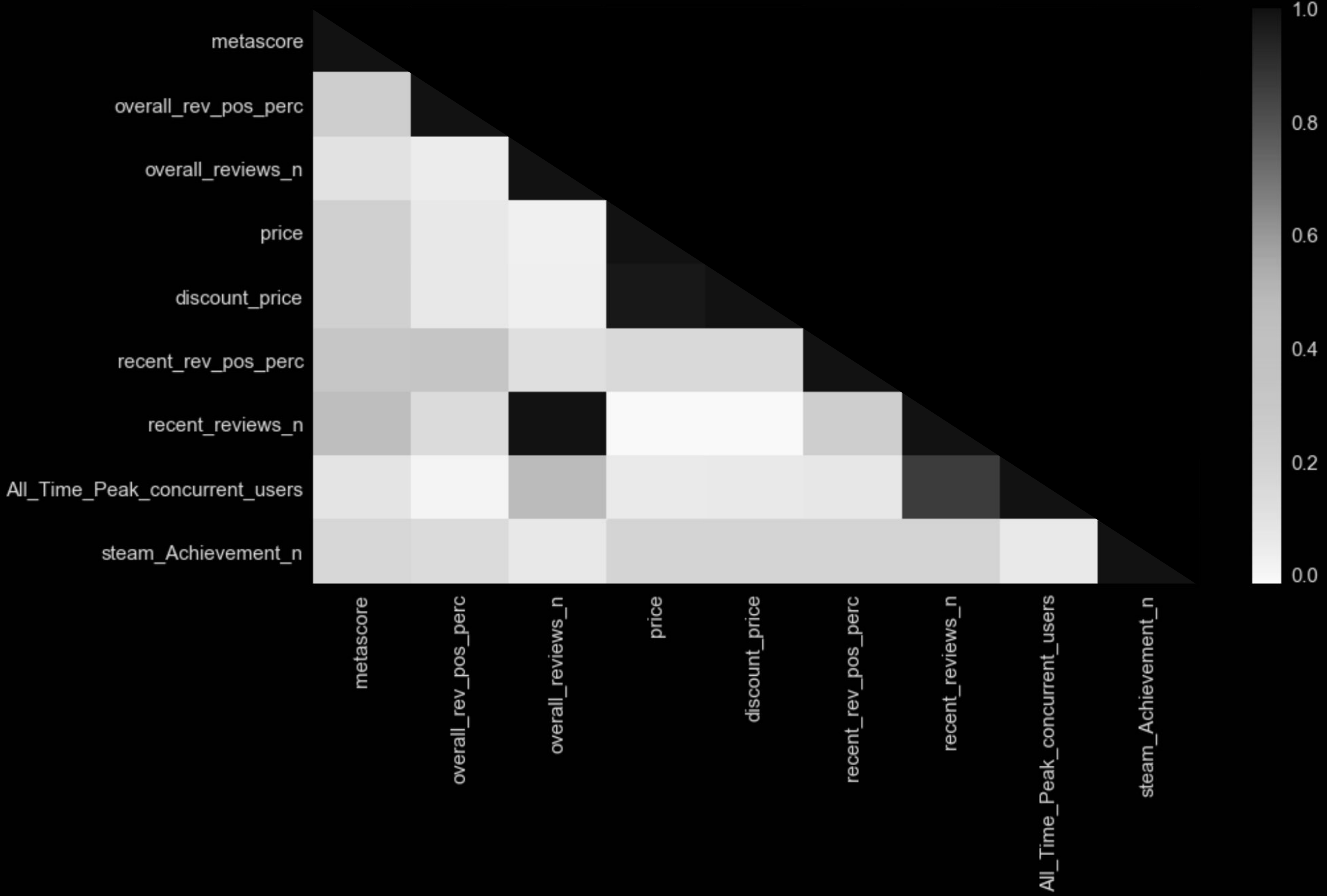
w/ lasso: 0.024

w/ ridge: 0.00736

Heatmap correlation between tags

Plot residuals


0.9
0.6
0.3
0.0
−0.3
−0.6



metascore
overall_rev_pos_perc
overall_reviews_n
price
discount_price
recent_rev_pos_perc
recent_reviews_n
All_Time_Peak_concurrent_users
steam_Achievement_n
1.0
0.8
0.6
0.4
0.2
0.0

Coefficients for k select-best

'Discount_price'	0.35191321764886785
'Metascore'	0.5427651876621874
'Overall_rev_pos_perc'	0.06471729618599727
'Overall_reviews_n'	0.44041114779194246
'Price'	0.06471729618599707
'Recent_rev_pos_perc'	0.20732552893908276
'Recent_reviews_n'	0.12486045021661757
'steam_Achievement_n'	0.2207162479411279
'1980s'	0.30148250975401025
'1990'	0.3014825097540103

Coefficients for k select-best with log(y)

'Discount_price'	9999.114731628768
'Metascore'	1278.0476087302811
'Overall_rev_pos_perc'	1222.9442277864596
'Overall_reviews_n'	2245.3153201385626
'Price'	496.5336479490818
'Recent_rev_pos_perc'	677.2985247433261
'Recent_reviews_n'	4312.236707314003
'steam_Achievement_n'	-1784.0977483984502
'1980s'	4447.772258636796
'1990'	4447.772258636796

OLS improved by Select K-Best Features:

All features:

```
cv_y = y

cv_result = model_selection.cross_validate(
        OLS_pipe, X=x, y=cv_y, cv=10, return_train_score=True)
print(f"train: {np.mean(cv_result['train_score']):.3}, test: {np.mean(cv_result['test_score']):.3}")
```

executed in 7.29s, finished 10:30:14 2018-04-27

```
train: 0.412, test: -1.43e+25
```

Only top 10 features:

```
cv_y = y

cv_result = model_selection.cross_validate(
        select_pipe, X=x, y=cv_y, cv=10, return_train_score=True)
print(f"train: {np.mean(cv_result['train_score']):.3}, test: {np.mean(cv_result['test_score']):.3}")
```

executed in 3.97s, finished 10:23:25 2018-04-27

```
train: 0.329, test: -2.11
```

OLS improved by Select K-Best Features:

All features:

```
cv_y = y

cv_result = model_selection.cross_validate(
        OLS_pipe, X=x, y=cv_y, cv=10, return_train_score=True)
print(f"train: {np.mean(cv_result['train_score']):.3}, test: {np.mean(cv_result['test_score']):.3}")
```

executed in 7.29s, finished 10:30:14 2018-04-27

```
train: 0.412, test: -1.43e+25
```

Only top 10 features:

```
cv_y = y

cv_result = model_selection.cross_validate(
        select_pipe, X=x, y=cv_y, cv=10, return_train_score=True)
print(f"train: {np.mean(cv_result['train_score']):.3}, test: {np.mean(cv_result['test_score']):.3}")
```

executed in 3.97s, finished 10:23:25 2018-04-27

```
train: 0.329, test: -2.11
```

Select k-best features

10 features:

```
cv_y = y

cv_result = model_selection.cross_validate(
        select_pipe, X=x, y=cv_y, cv=10, return_train_score=True)
print(f"train: {np.mean(cv_result['train_score']):.3}, test: {np.mean(cv_result['test_score']):.3}")
```

executed in 3.97s, finished 10:23:25 2018-04-27

```
train: 0.329, test: -2.11
```

10 features with a log-transformed y:

```
cv_y = log(y)

cv_result = model_selection.cross_validate(
        select_pipe, X=x, y=cv_y, cv=10, return_train_score=True)
print(f"train: {np.mean(cv_result['train_score']):.3}, test: {np.mean(cv_result['test_score']):.3}")
```

executed in 3.97s, finished 06:11:57 2018-04-27

```
train: 0.446, test: 0.442
```