what the heck happened in the 2016 presidential election?

an investigation into the aca

david azaria / dat-nyc-45 / feb 2017

so what was **my** problem statement?

"could **we** have seen the 2016 outcome coming from a mile away?"

and to **whom** does it even matter?

in reality, **anyone** who has some vested interest in predicting and/or better understanding elections and voter behavior

so what **datasets** did i look at?

2012 and 2016 presidential election data / individual on market aca prices across 2015, 2016, and 2017 plan years (all released the previous year)

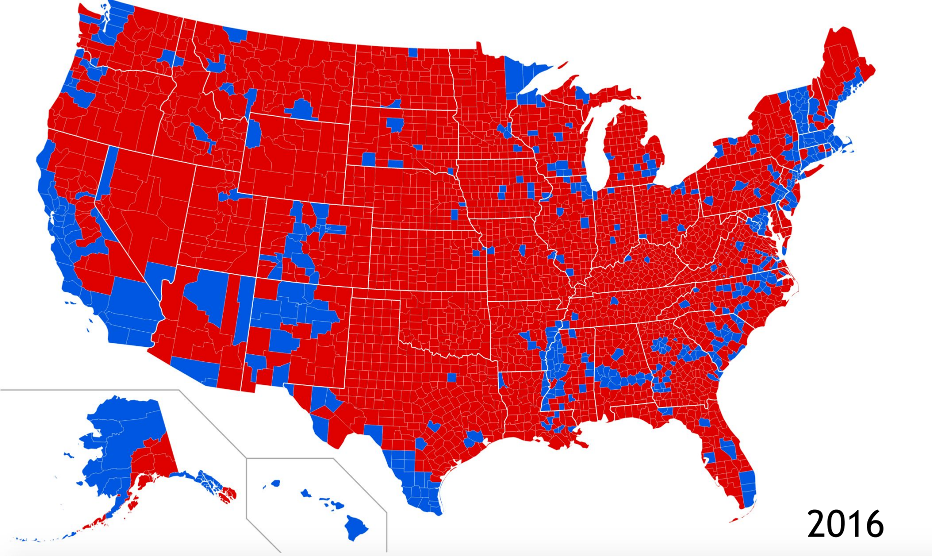
and what will be the **ML** exercise?

could i use a machine learning model to accurately predict if a county will be won by donald trump using aca prices?

but first, a story…

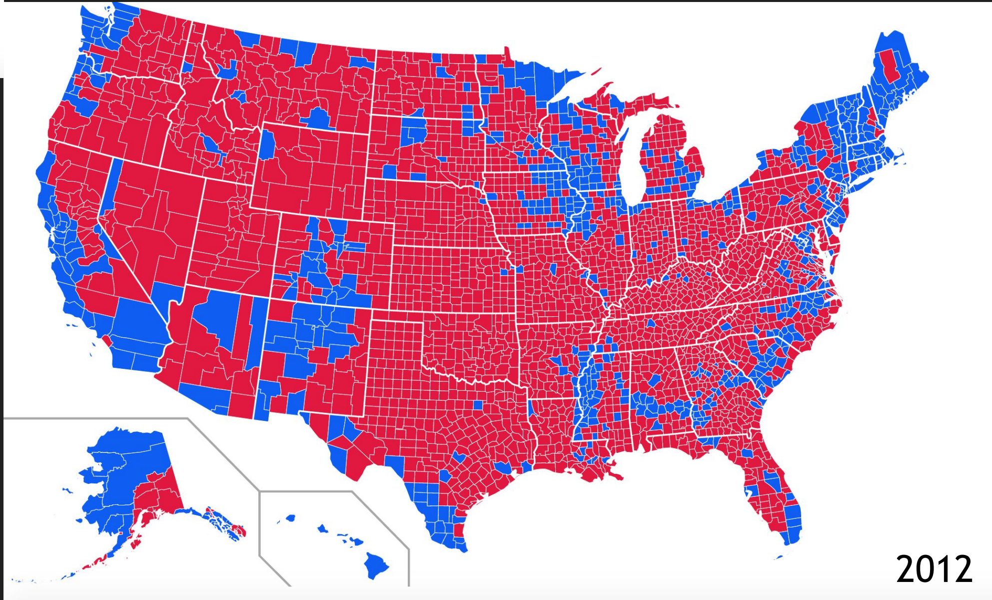
...election night 2016...

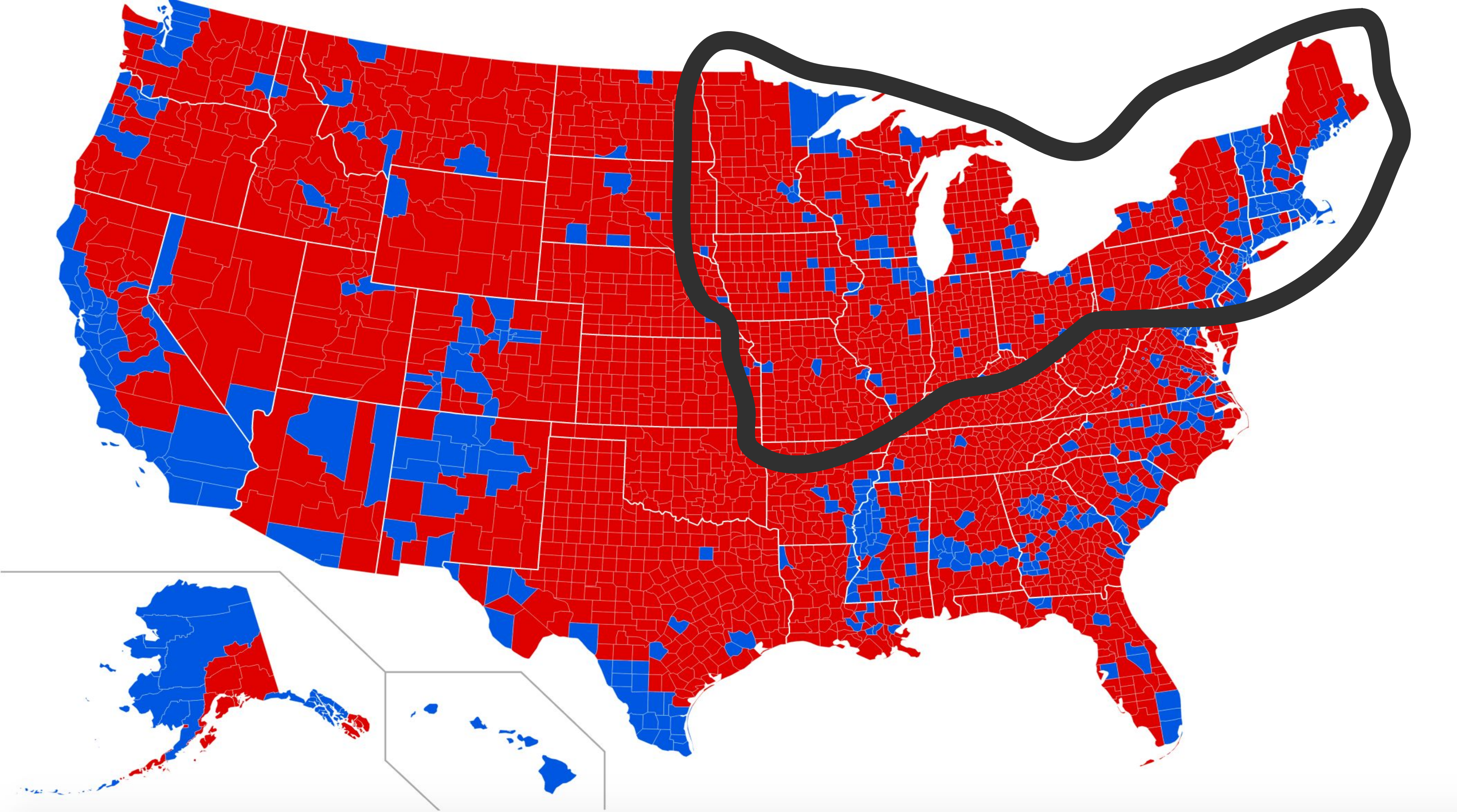
...wtf?



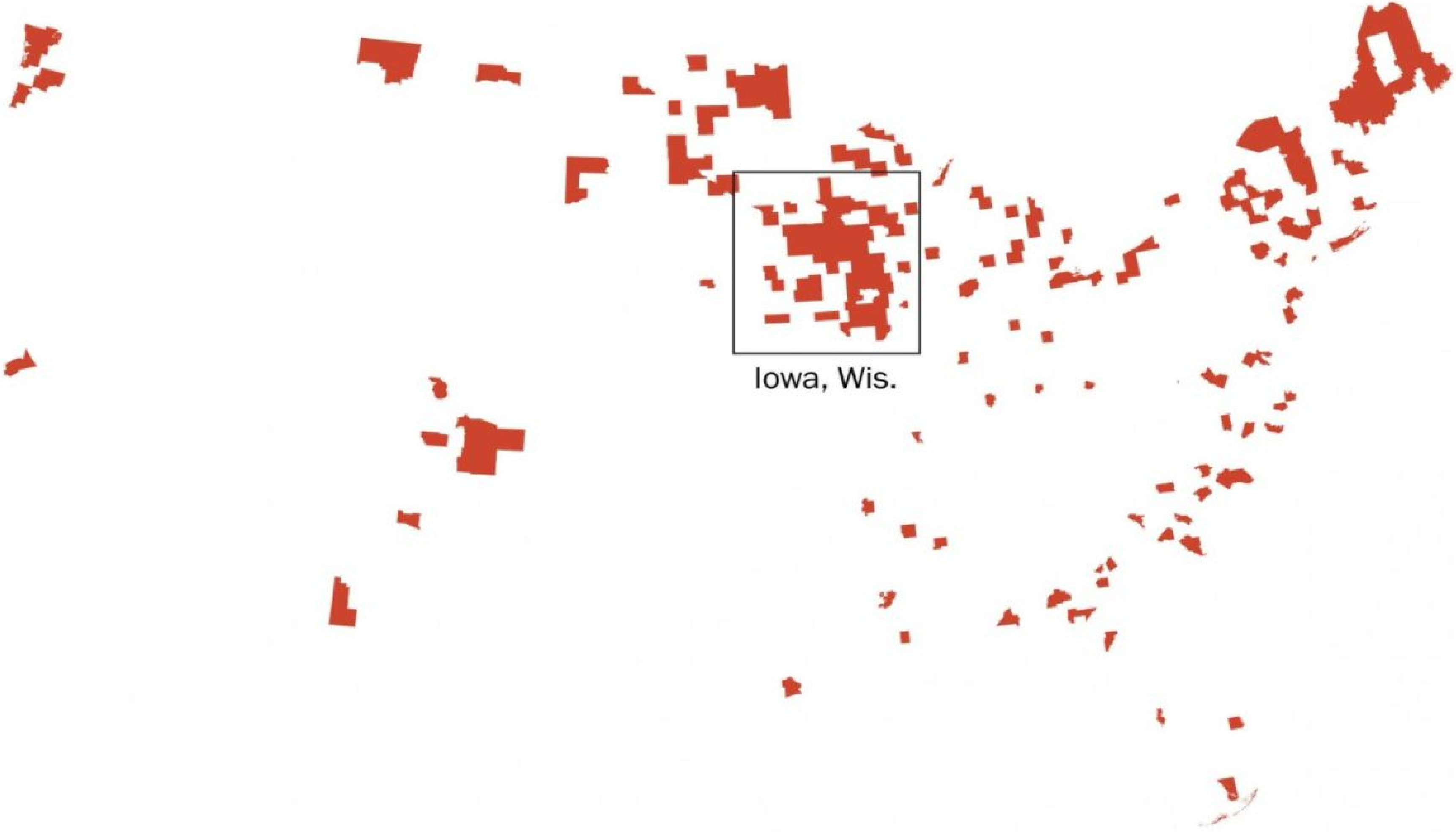
← see all that new red?

why didn’t it look more like this? →

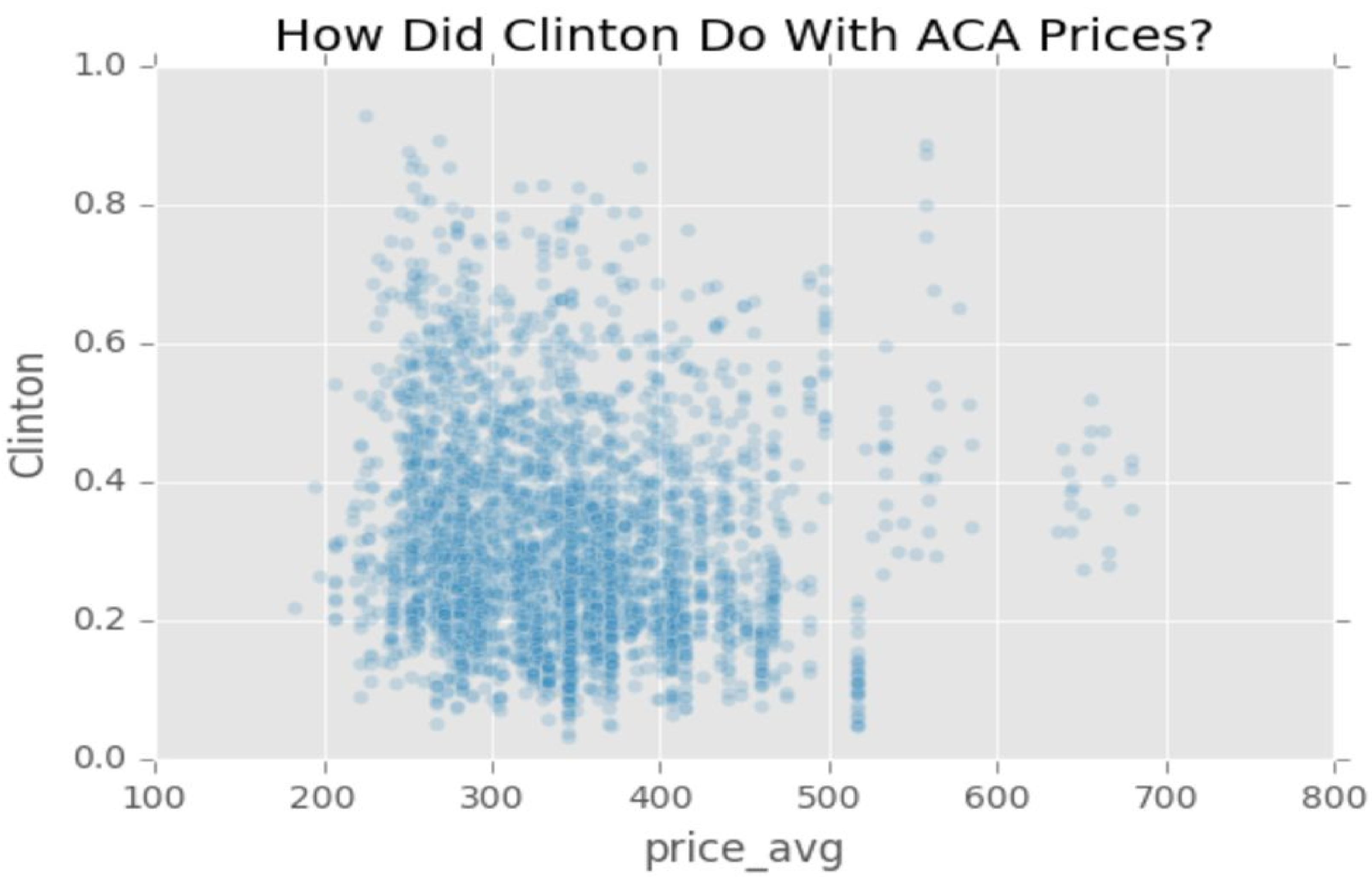


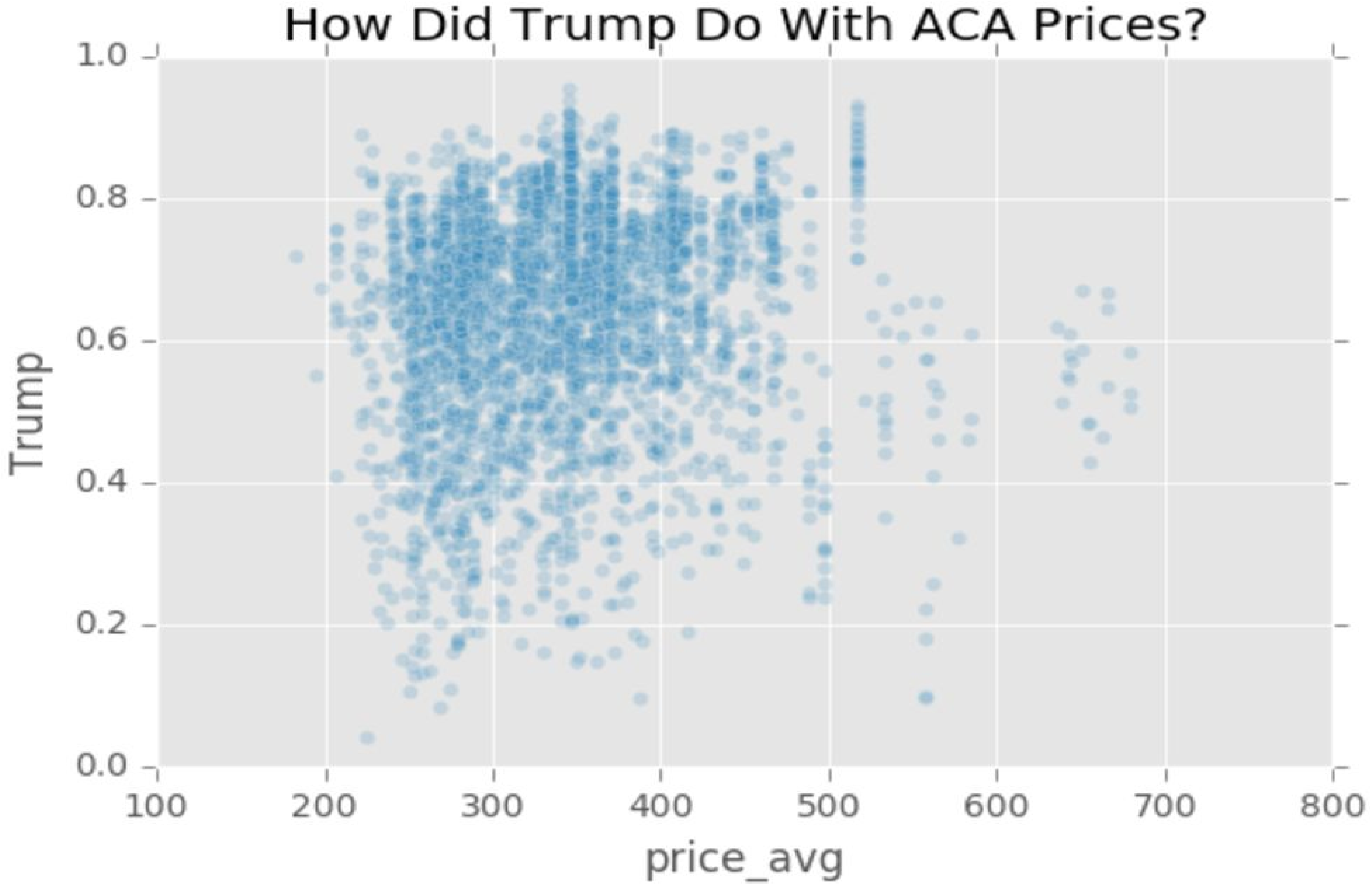


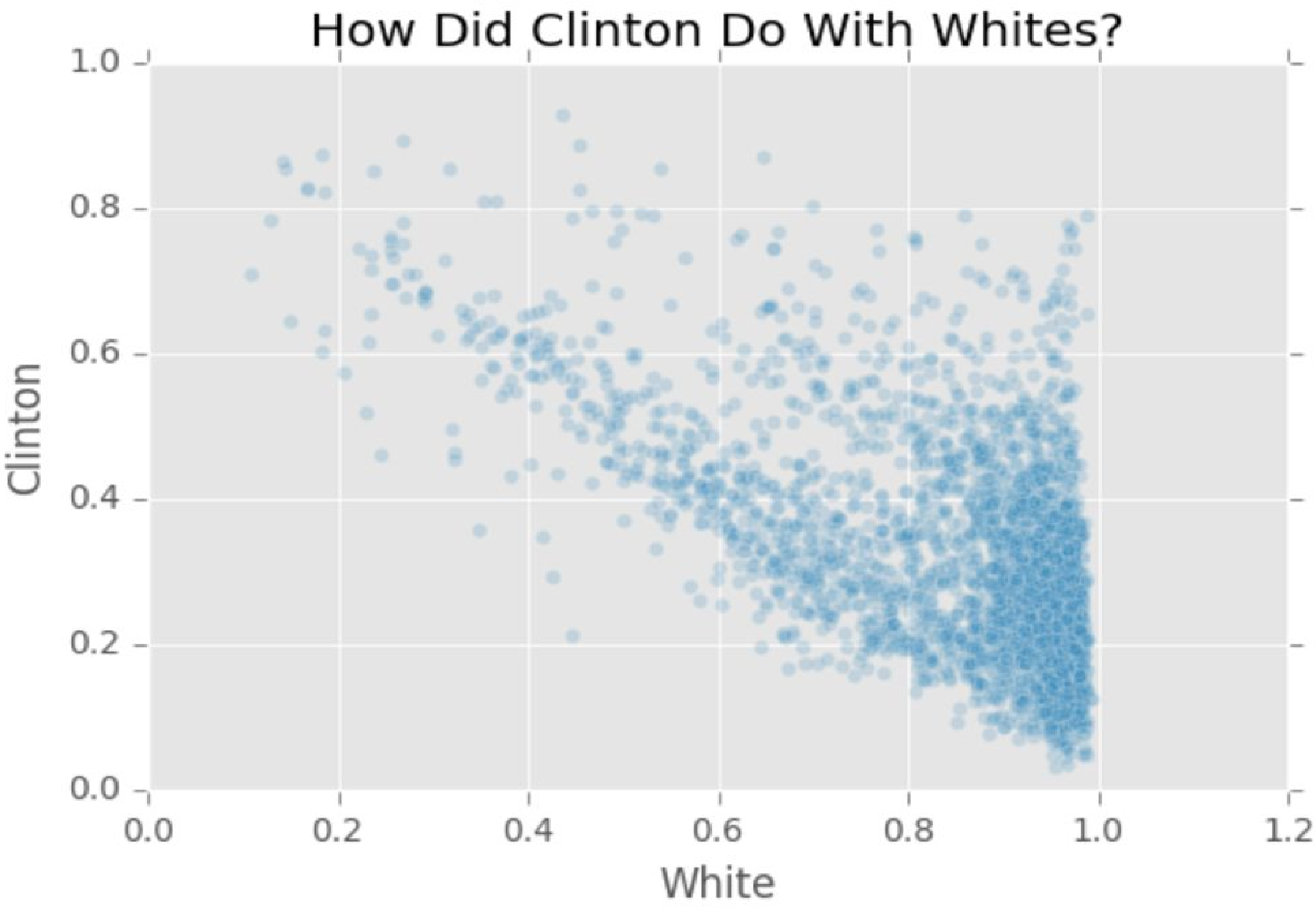
nb: i do not have an art degree



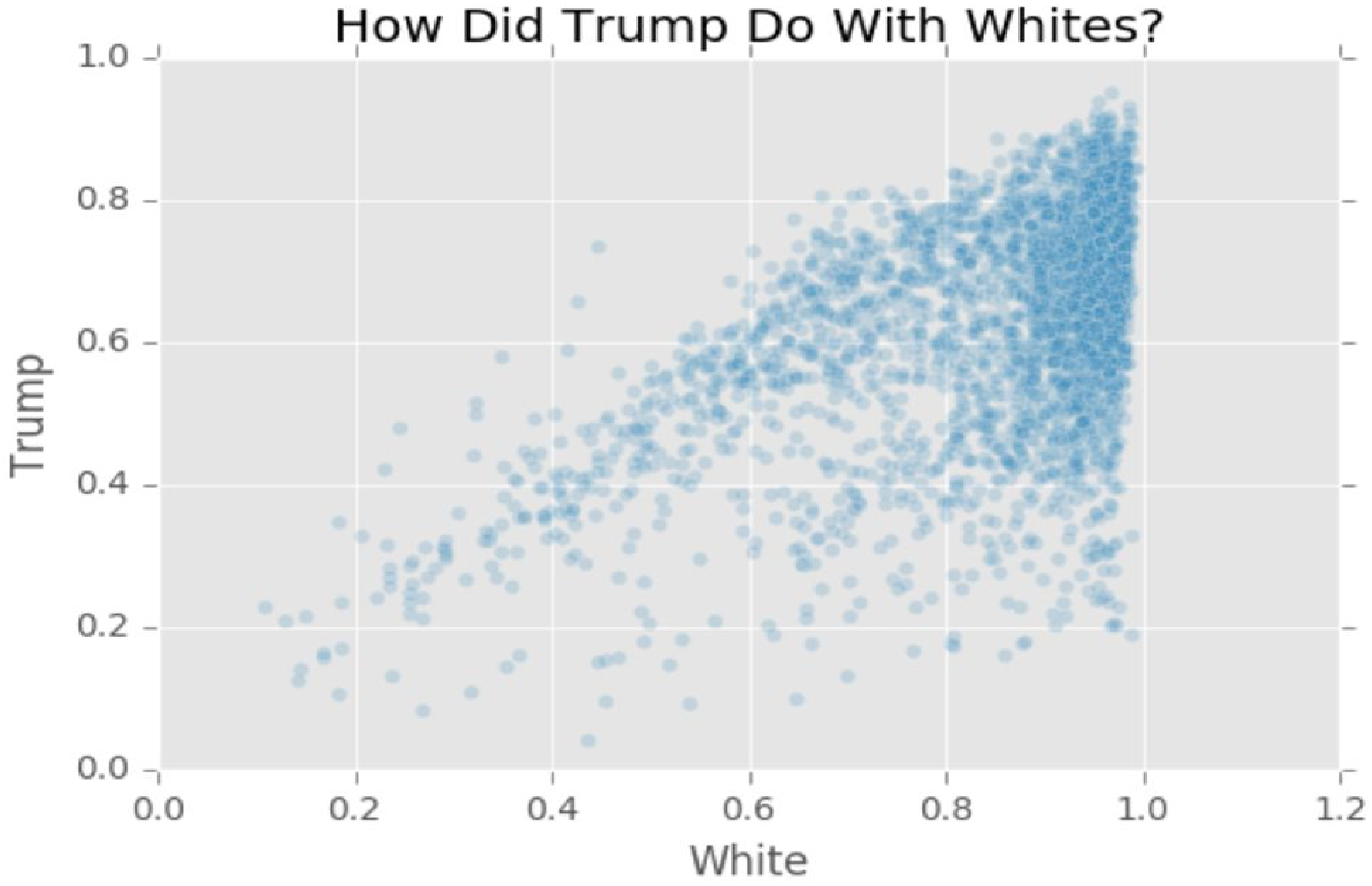
Iowa, Wis.


How Did Clinton Do With ACA Prices?
Clinton
price_avg
1.0
0.8
0.6
0.4
0.2
0.0
100
200
300
400
500
600
700
800


How Did Trump Do With ACA Prices?
Trump
price_avg
0.0
0.2
0.4
0.6
0.8
1.0
100
200
300
400
500
600
700
800


How Did Clinton Do With Whites?
Clinton
White
0.0
0.2
0.4
0.6
0.8
1.0
1.2

awful


How Did Trump Do With Whites?
Trump
White
0.0
0.2
0.4
0.6
0.8
1.0
1.2

tremendous

some normalization and parameter setting...nbd

```
sherlock['Huge_Price_Swing'] = np.where((sherlock.price_pct_ch > .50), 1, 0)
sherlock['election_year'] = np.where(sherlock.year > 2016, 1, 0)
sherlock['Trump_Win'] = np.where(sherlock.Trump > sherlock.Clinton, 1, 0)
sherlock['Obama_Win'] = np.where(sherlock.Obama > sherlock.Romney, 1, 0)
sherlock['Clinton_Win'] = np.where(sherlock.Clinton > sherlock.Trump, 1, 0)
sherlock['Romney_Win'] = np.where(sherlock.Romney > sherlock.Obama, 1, 0)
sherlock['Whiter_County'] = np.where(sherlock.White > .96, 1,0)
sherlock['Blacker_County'] = np.where(sherlock.Black > .11, 1, 0)
sherlock['More_Female_County'] = np.where(sherlock.SEX255214 > 51, 1, 0)
sherlock['More_Hispanic_County'] = np.where(sherlock.Hispanic > .09, 1, 0)
sherlock['Immigrant_County'] = np.where(sherlock.POP645213 > 6, 1, 0)
sherlock['Educated_County'] = np.where(sherlock.Edu_batchelors > 23.2, 1,0)
sherlock['Undereducated_County'] = np.where(sherlock.Edu_highschool < 85, 1, 0)
sherlock['Sig_Population_Change'] = np.where(sherlock.population_change > 2.2, 1, 0)
sherlock['Older_County'] = np.where(sherlock.age65plus > 20, 1, 0)
sherlock['Wealthier_County'] = np.where(sherlock.INC110213 > 51000, 1, 0)
sherlock['Poorer_County'] = np.where(sherlock.Poverty > 20, 1, 0)
sherlock['Denser_County'] = np.where(sherlock.Density > 1000, 1,0)
sherlock['Rural_County'] = np.where(sherlock.Density < 20, 1, 0)
sherlock['County_Name'] = sherlock['county_name_x']
sherlock['State_Code'] = sherlock['state_code']
sherlock['More_Republican'] = np.where(sherlock.votes_gop_2016 > sherlock.votes_gop_2012, 1, 0)
sherlock['More_Democratic'] = np.where(sherlock.votes_dem_2016 > sherlock.votes_dem_2012, 1, 0)
sherlock['Turnout_Increase'] = np.where(sherlock.total_votes_2016 > sherlock.total_votes_2012, 1, 0)
```

	Trump_Win	Clinton_Win	Obama_Win	Romney_Win	More_Republican	More_Democratic	Trump_Flip	Clinton_Flip
Trump_Win	1.000000	-1.000000	-0.764808	0.764808	0.372652	-0.308569	0.116812	-0.185268
Clinton_Win	-1.000000	1.000000	0.764808	-0.764808	-0.372652	0.308569	-0.116812	0.185268
Obama_Win	-0.764808	0.764808	1.000000	-1.000000	-0.251218	0.175042	0.523178	-0.041366
Romney_Win	0.764808	-0.764808	-1.000000	1.000000	0.251218	-0.175042	-0.523178	0.041366
More_Republican	0.372652	-0.372652	-0.251218	0.251218	1.000000	-0.224167	0.080190	-0.127191
More_Democratic	-0.308569	0.308569	0.175042	-0.175042	-0.224167	1.000000	-0.095188	0.184967
Trump_Flip	0.116812	-0.116812	0.523178	-0.523178	0.080190	-0.095188	1.000000	-0.021642
Clinton_Flip	-0.185268	0.185268	-0.041366	0.041366	-0.127191	0.184967	-0.021642	1.000000
Huge_Price_Swing	0.123699	-0.123699	-0.119976	0.119976	0.060100	-0.050002	-0.025855	-0.021116
Turnout_Increase	0.132986	-0.132986	-0.211125	0.211125	0.294265	0.240971	-0.138936	0.046537
Sig_Population_Change	-0.203048	0.203048	0.094041	-0.094041	-0.170768	0.473339	-0.097448	0.119203
Educated_County	-0.338145	0.338145	0.268253	-0.268253	-0.254740	0.437223	-0.010309	0.107434
Undereducated_County	0.008456	-0.008456	-0.067049	0.067049	0.031622	-0.099533	-0.107474	-0.037289
Whiter_County	0.175768	-0.175768	-0.147909	0.147909	0.168342	-0.173623	-0.004953	-0.044933
Blacker_County	-0.262642	0.262642	0.220826	-0.220826	-0.175447	0.058812	-0.012625	0.002538
More_Female_County	-0.180803	0.180803	0.156444	-0.156444	-0.125471	0.073347	-0.013025	-0.035807
More_Hispanic_County	-0.149045	0.149045	0.083517	-0.083517	-0.151540	0.336514	-0.057129	0.058255
Older_County	0.161277	-0.161277	-0.123347	0.123347	0.086891	-0.136847	0.017280	-0.034984
Wealthier_County	-0.153337	0.153337	0.119644	-0.119644	-0.117639	0.292276	0.001739	0.080276
Poorer_County	-0.135471	0.135471	0.091156	-0.091156	-0.070026	-0.062770	-0.040612	0.009593
Denser_County	-0.409308	0.409308	0.323132	-0.323132	-0.268227	0.310654	-0.023480	0.102315
FIPS	0.023924	-0.023924	-0.012385	0.012385	0.036422	0.014163	0.013993	0.001064
Rural_County	0.106740	-0.106740	-0.114863	0.114863	-0.052036	-0.120267	-0.044042	-0.029612
Immigrant_County	-0.306069	0.306069	0.196577	-0.196577	-0.258915	0.468364	-0.076941	0.119634

check out “huge price swing”...

what else are you noticing?

```
feature_cols = ['Huge_Price_Swing', 'Obama_Win']
X = sherlock_holmes[feature_cols]
y = sherlock_holmes['Trump_Win']
```

my original features

my coefficients

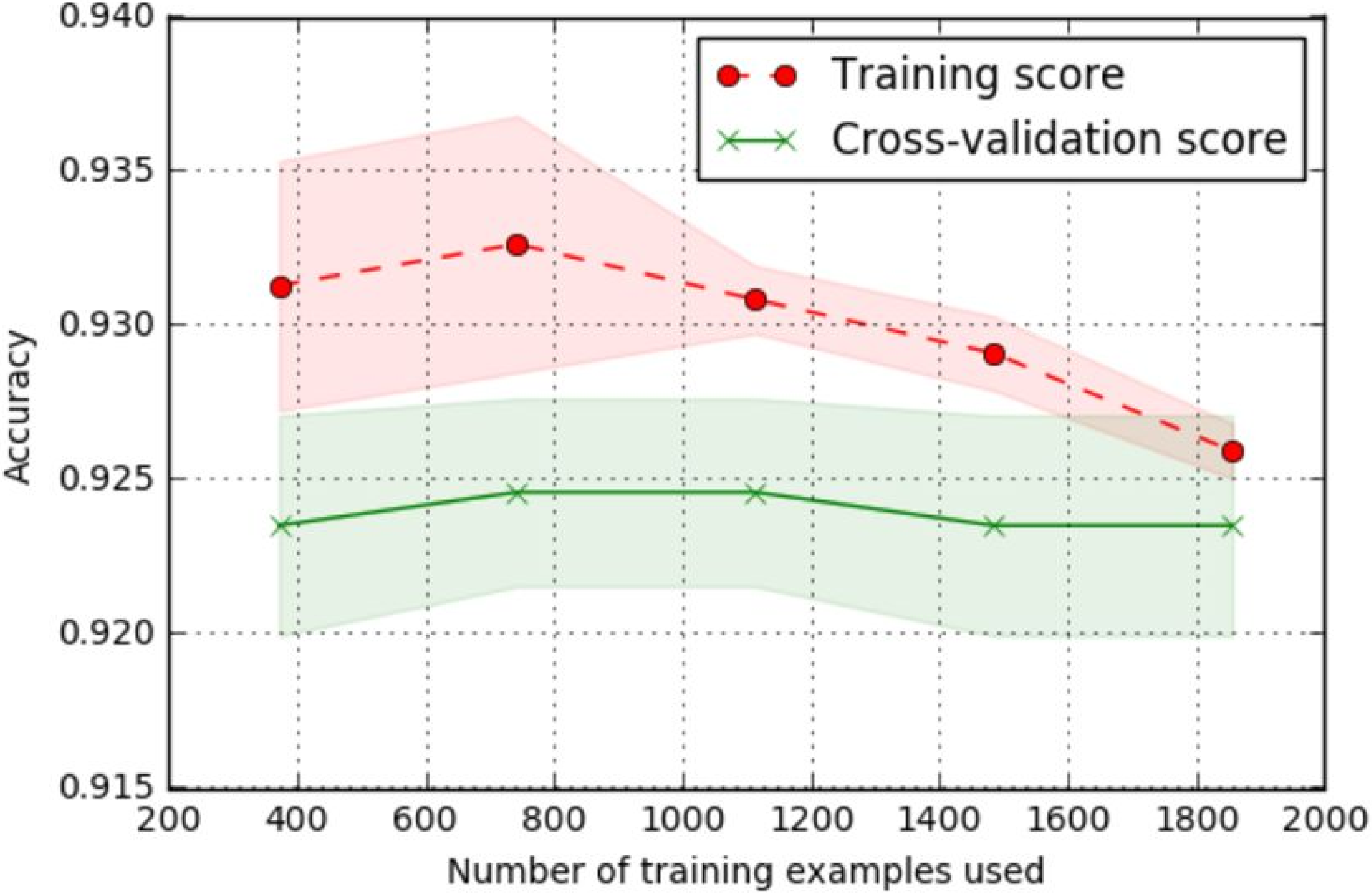
```
logreg.fit(X_train_std, y_train)
zip(feature_cols, logreg.coef_[0])
```

```
[('Huge_Price_Swing', 0.36036884062608088), ('Obama_Win', -2.279381374374152 9)]
```

my accuracy score

```
from sklearn import metrics
print metrics.accuracy_score(y_test, y_pred_class)
```

```
0.923772609819
```



Training score
Cross-validation score
Accuracy
0.940
0.935
0.930
0.925
0.920
0.915
200
400
600
800
1000
1200
1400
1600
1800
2000
Number of training examples used

but i wasn’t exactly satisfied...

...nor did i know whether that was outcome was significant

```python
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

X = sherlock_holmes[['Huge_Price_Swing', 'Obama_Win', 'Trump_Win',
                     'Whiter_County', 'Educated_County', 'Turnout_Increase']].dropna()
y = X['Trump_Win']
X.drop('Trump_Win', axis=1, inplace=True)

model.fit(X, y)

from sklearn.tree import export_graphviz
from os import system
def build_tree_image(model):
    dotfile = open("tree.dot", 'w')
    export_graphviz(model,
                            out_file = dotfile,
                            feature_names = X.columns)
    dotfile.close()
    system("dot -Tpng tree.dot -o tree.png")

build_tree_image(model)
```

1

```python
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

X = sherlock_holmes[['Huge_Price_Swing', 'Trump_Flip', 'Turnout_Increase']].dropna()
y = X['Trump_Flip']
X.drop('Trump_Flip', axis=1, inplace=True)

model.fit(X, y)

from sklearn.tree import export_graphviz
from os import system
def build_tree_image(model):
    dotfile = open("tree.dot", 'w')
    export_graphviz(model,
                            out_file = dotfile,
                            feature_names = X.columns)
    dotfile.close()
    system("dot -Tpng tree.dot -o tree.png")

build_tree_image(model)
```

2

	Features	Importance Score
1	Turnout_Increase	0.959275
0	Huge_Price_Swing	0.040725

rf not particularly conclusive or reassuring...

	Features	Importance Score
1	Obama_Win	0.815060
3	Educated_County	0.114795
2	Whiter_County	0.033616
4	Turnout_Increase	0.019695
0	Huge_Price_Swing	0.016834

¯_(ツ)_/¯