Linear regression and classification in NLP

Kevin Foley
PhD Candidate (Government)
Cornell University
kpf32@cornell.edu

My personal interests in NLP

- Modeling discourse and political rhetoric
- Bringing statistics into NLP (probability distributions, quantifying uncertainty)
- Working with difficult data sources (multilingual, archival / historical / printed materials, non-Latin scripts and non Indo-European languages)
- Most of my deep learning and machine learning work goes into OCR, word boundary segmentation and other NLP pipeline tasks

Linear Regression

- Fitting a line to points
- Points can be survey responses, income, democracy scores
- Points can also be pixel data, word labels, part of speech tags
- Disconnect between how computer science / machine learning field and social sciences talk about regression

- Categorical vs. continuous data
- One dimension vs. many dimensions

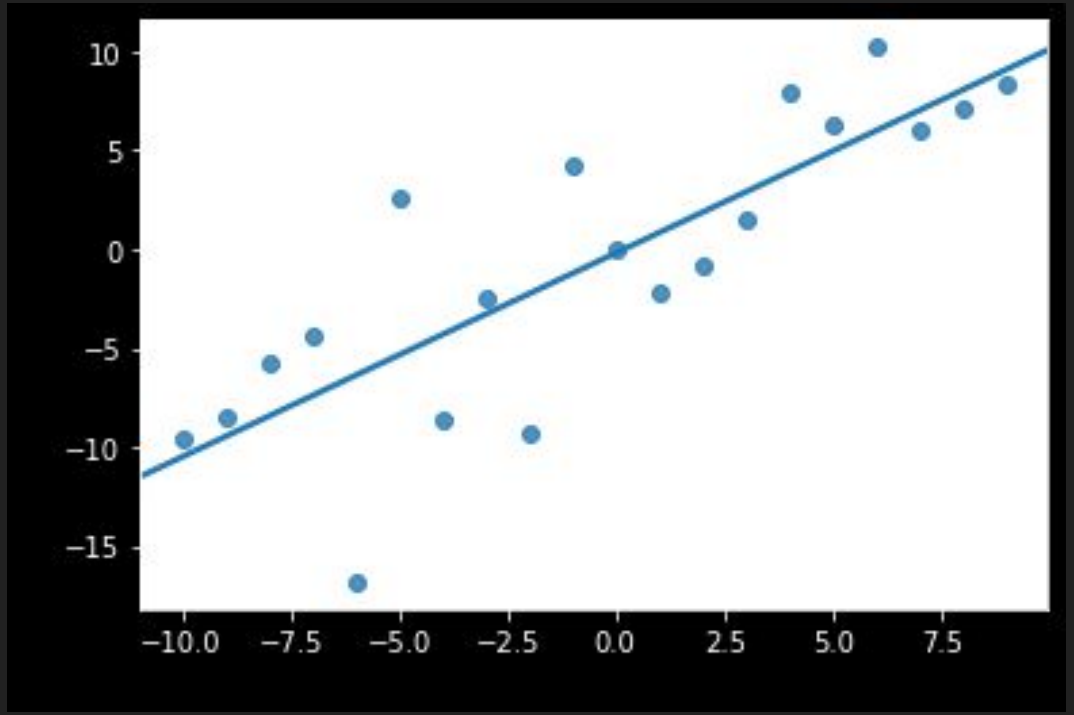
- **weights** (ML) a.k.a. **coefficients** (statistics) are a vector of values that map X to Y
- **bias** (ML) a.k.a. **intercept** (stats) is a constant value that shifts the regression line up or down.
- Bias can be incorporated into the model by adding a constant vector of 1's to the data

Translating between machine learning and statistics

features	variables
inputs	independent variables
outputs	dependent variables / outcome variables
weights	coefficients / parameter estimates
bias	intercept
hidden	unobserved

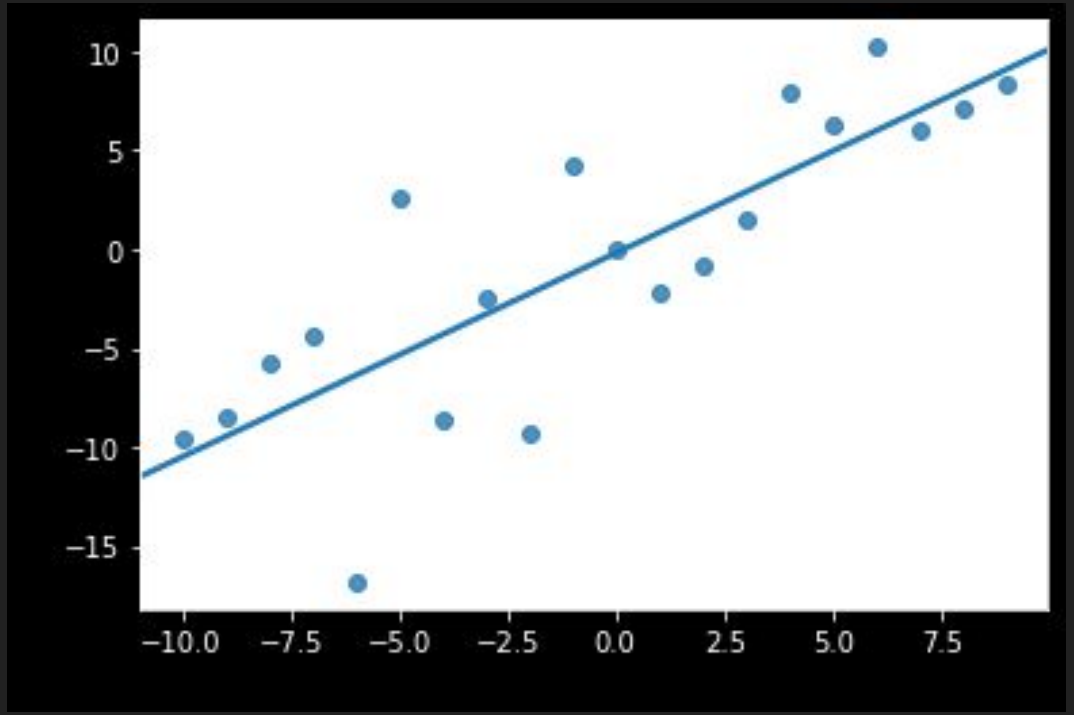
Linear Regression

X	Y	X	Y
-10.0	-9.6	0.0	0.0
-9.0	-8.6	1.0	-2.1
-8.0	-5.7	2.0	-0.9
-7.0	-4.4	3.0	1.5
-6.0	-16.8	4.0	7.9
-5.0	2.6	5.0	6.3
-4.0	-8.6	6.0	10.2
-3.0	-2.4	7.0	6.1
-2.0	-9.3	8.0	7.2
-1.0	4.3	9.0	8.4



Ordinary Least Squares (OLS) estimator

- Fits continuous data with outcomes ranging from (-inf,inf)
- Minimizes mean squared error
- First model taught in most statistics classes
- Widely used in social and natural science
- Convenient closed-form estimator



Ordinary Least Squares (OLS) estimator

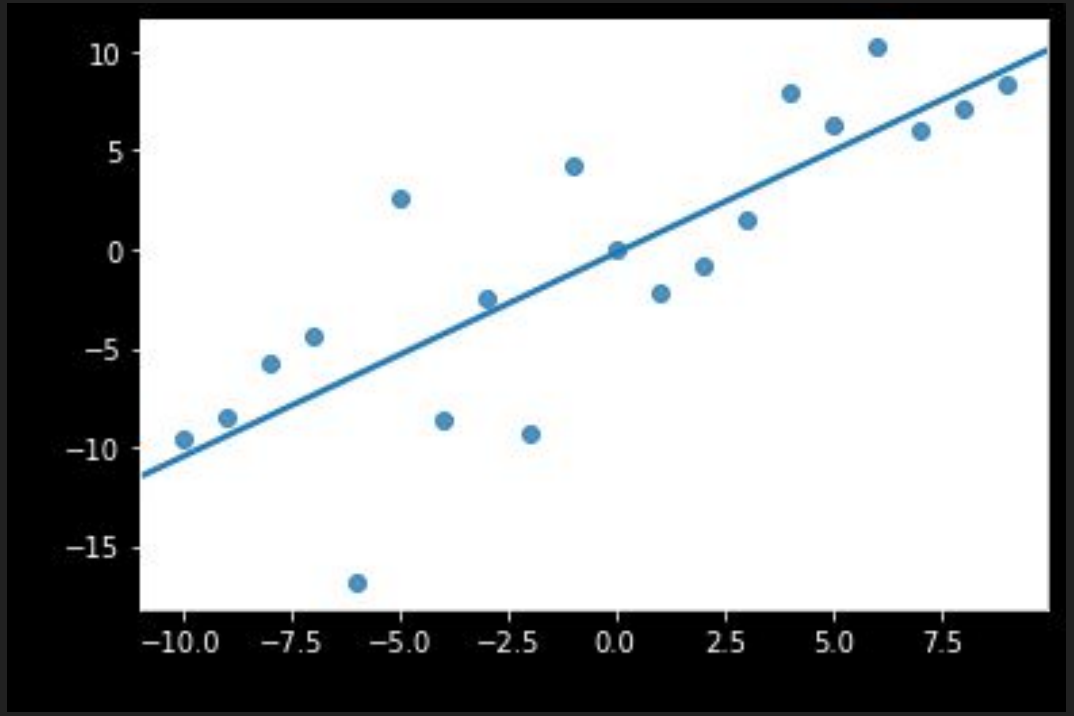
- Convenient closed-form estimator:

$$\hat{\beta} = (X'X)^{-1} X'y$$

(where β is a vector of weights and y is the labeled output data)

Prediction: guessing value of new data

$$\hat{Y} = \beta X$$



Ordinary Least Squares (OLS) estimator

- Better suited to regression problems with continuous output
- Not the best choice for classification problems
- but FWIW there is a closed form solution for "multiclass" problems; just replace the output vector **y** with a matrix **Y.**

  (But you wouldn't really want to use this for classification problems!)

- Estimator:

$$\hat{\beta} = (X'X)^{-1} X'y$$

- For 2+ outcome variables:

$$\hat{B} = (X'X)^{-1} X'Y$$

OLS: social science example

TABLE 5. Frequency of Female Speech

	Dependent variable:					
	Female speech					
	Citizens (1)	Citizens (2)	Admin. (3)	Admin. (4)	Politicians (5)	Politicians (6)
Female president	0.10	0.11	0.21***	0.19***	0.90***	0.88***
	(0.08)	(0.09)	(0.09)	(0.08)	(0.07)	(0.08)
Female attendance	0.001**	0.001**	0.0003	0.0000	−0.0002	−0.0002
	(0.0005)	(0.001)	(0.001)	(0.001)	(0.0004)	(0.0004)
Female literacy	0.34	0.42	0.06	−0.14	−0.11	−0.34
	(0.40)	(0.41)	(0.66)	(0.81)	(0.43)	(0.46)
District FE	✓	✓	✓	✓	✓	✓
Backwardness score control		✓		✓		✓
Observations	913	913	473	473	322	322

Note: $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$. Robust Standard Errors, clustered at the district, in parenthesis. The Backwardness Score is a measure of village-level development, calculating using demographic and infrastructural variables, including the share of population belonging to the Scheduled Castes and Tribes, as well as indicators for the presence of a primary or secondary school, hospital or medical clinic, and bank.

Parthasarathy, Ramya, Vijayendra Rao, and Nethra Palaniswamy. "Deliberative Democracy in an Unequal World: A Text-As-Data Study of South India's Village Assemblies." *American Political Science Review* 113, no. 3 (2019): 623-640.

But what if your outcome data is categorical?

- example 1: age in years ⇒ worried about climate change
- yes = 1, no = 0

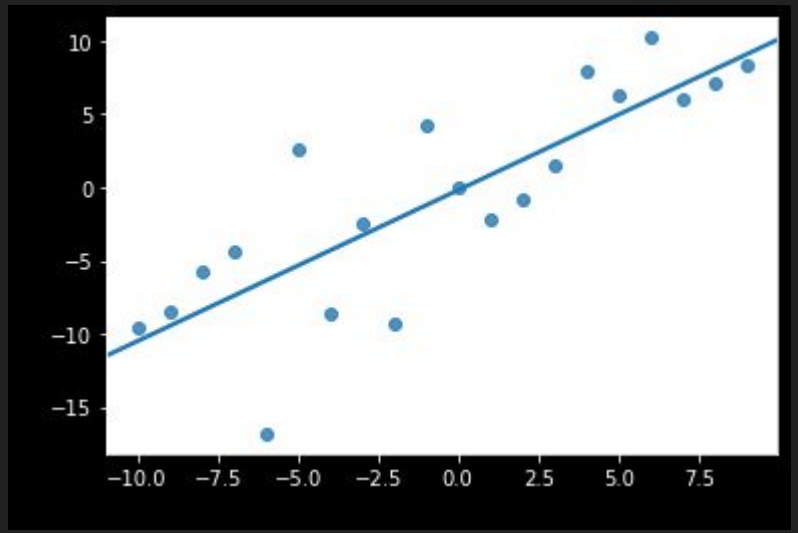
- example 2: pixel values ⇒ "is this a photo of a dog"
- yes = 1, no = 0

- example 3: word counts ⇒ "is this novel / news article / email / tweet"
- novel = {1,0,0}, article = {0,1,0}, tweet = {0,0,1}

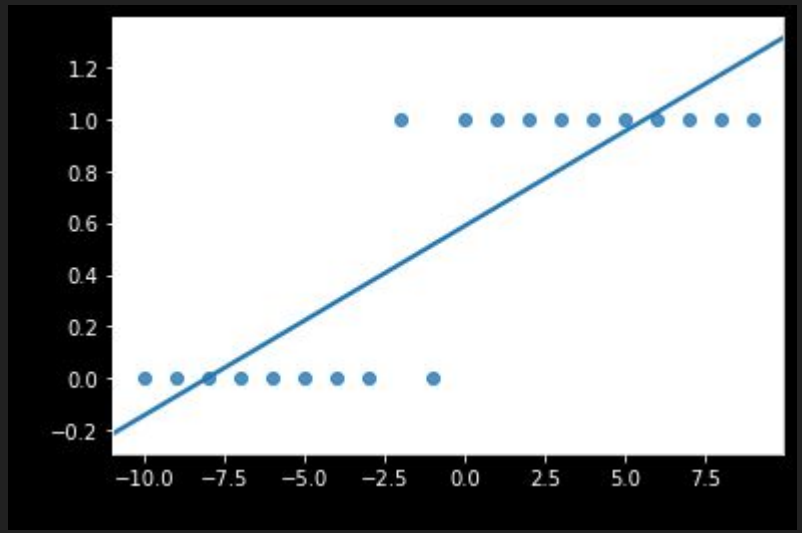
(last two are classification problems)

But what if your outcome data is categorical?

- OLS (the blue regression line) doesn't fit the data so well when there are only two possible outcomes in a given dimension.



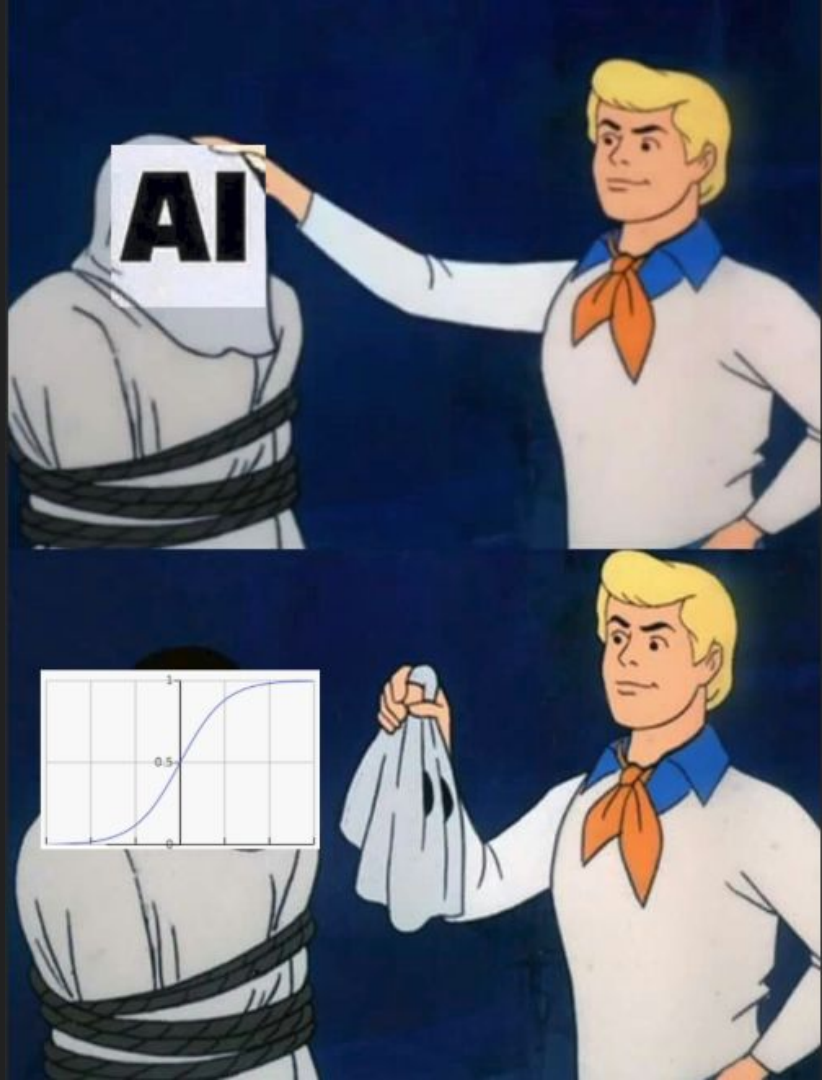
continuous output



categorical (discrete) output

Logistic Regression

- Fits continuous data with categorical outcomes
- Used in social science for categorical outcomes, used in ML classification tasks
- No convenient closed-form solution, estimated with optimizers (stochastic gradient descent, etc.)
- ***but don't forget - we're still just fitting lines to data points***



Logistic Regression

- Fits continuous data with categorical outcomes
- Used in social science for categorical outcomes, used in ML classification tasks
- No convenient closed-form solution, estimated with optimizers (stochastic gradient descent, etc.)
- ***this isn't magic - we're still just fitting lines to data points!***

$$
\begin{aligned}
P(y=1) &= \sigma(w \cdot x + b) \\
&= \frac{1}{1+e^{-(w \cdot x+b)}} \\
P(y=0) &= 1-\sigma(w \cdot x+b) \\
&= 1-\frac{1}{1+e^{-(w \cdot x+b)}} \\
&= \frac{e^{-(w \cdot x+b)}}{1+e^{-(w \cdot x+b)}}
\end{aligned}
$$

Logistic regression: social science example

Table 3: **Early Protests and Probability of Promotion for Party Secretaries**

	(3.1)	(3.2)	(3.3)
Early protests (before 15 Sept)	**−0.079****	**−0.081****	**−0.071***
	(0.035)	(0.036)	(0.039)
Peak protests (on or after 15 Sept)	−0.010	−0.0091	−0.012
	(0.036)	(0.036)	(0.038)
GDP growth (pct, 1y lag)	0.0055	0.0059	0.0082
	(0.0075)	(0.0079)	(0.0081)
Observations	526	502	449

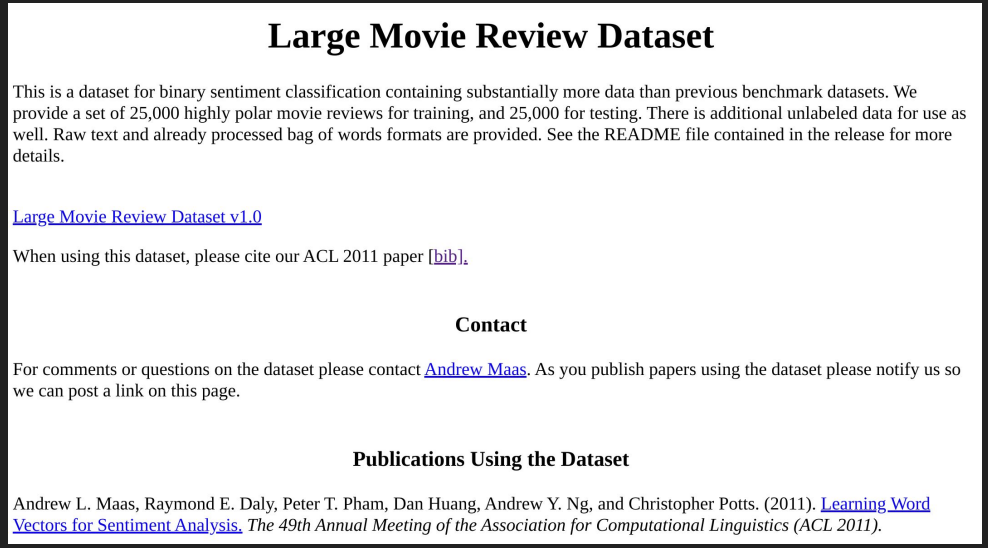
Notes:

Marginal effects on promotion probability at mean values. Model 3.1 includes all observations. Model 3.2 excludes changes made as the result of anti-corruption investigations. Model 3.3 excludes all individual–year observations associated with officials implicated in anti-corruption investigations. Clustered standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$. See also Appendix Table 3.

Foley, Kevin, Jeremy Wallace, and Jessica Chen Weiss. "The Political and Economic Consequences of Nationalist Protest in China: Repercussions of the 2012 Anti-Japanese Demonstrations." *China Quarterly* 236. (2018).

Regression classifiers: movie reviews example

- Dataset: Stanford Large Movie Review Dataset (Maas et al., 2011, "Learning word vectors for sentiment analysis")
- 25,000 positive reviews and 25,000 negative reviews
- split 50/50 into training and test sets



Large Movie Review Dataset

This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided. See the README file contained in the release for more details.

Large Movie Review Dataset v1.0

When using this dataset, please cite our ACL 2011 paper [bib].

Contact

For comments or questions on the dataset please contact Andrew Maas. As you publish papers using the dataset please notify us so we can post a link on this page.

Publications Using the Dataset

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*.

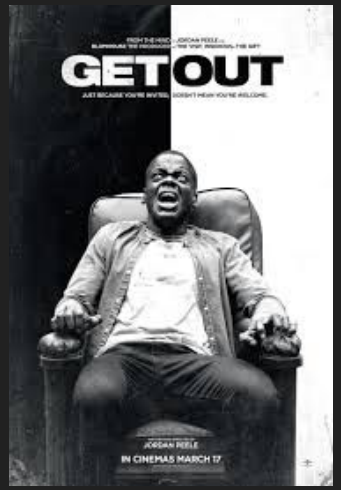
https://ai.stanford.edu/~amaas/data/sentiment/

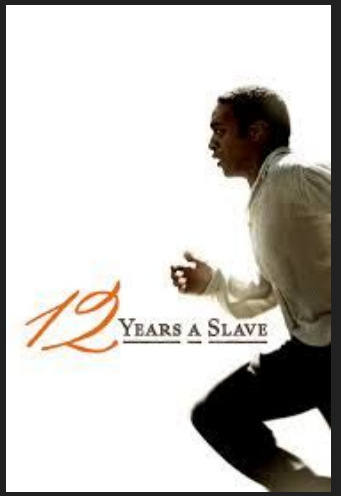
Three (out-of-sample) review snippets

"No one expects greatness from a movie called Dumb and Dumberer -- it would be dumb to expect much by way of humor or plot or character or energy -- but even so, this manages to be disappointing."

"This is one of the most searingly intense portraits of slavery ever committed to film, exercising onscreen brutality to bludgeon slavery's grim, cruel and conscience-less degradation."

"Peele has a solid script and a knack for storytelling in an appealing way which makes Get Out more enjoyable than many of the other suspense/thriller/horror movies being made today."





Regression classifiers: data processing steps

1. count words
2. maybe get rid of less common words
3. map each unique word to a column
4. Create a zero-valued vector for each document, setting each column to 1 if a word is present
5. stack the document vectors and label vectors

```
for val in ["neg", "pos"]:
    file_paths[val] = ["%s/train/%s/%s" % (data_root, val, x) for x in \
                       os.listdir("%s/train/%s/" % (data_root, val))]
    for file_path in tqdm(file_paths[val]):
        with open(file_path, "r") as f:
            doc = f.read().strip()
            vocab.update(split(doc))
            idx_doc[idx]  = doc
            idx_sent[idx] = -1 if val == "neg" else 1
            idx += 1

vocab_idx = {x[0]: i for i, x in enumerate(vocab.most_common()[0:V])}
idx_vocab = {value: key for key, value in vocab_idx.items()}

for doc_idx in tqdm(range(idx)):
    doc = idx_doc[doc_idx]
    idx_vec[doc_idx] = np.zeros(V, dtype=np.float32)
    doc_words = np.int32([vocab_idx[x] for x in list(set(split(doc))) if x in vocab_idx])
    idx_vec[doc_idx][doc_words] = 1

X = np.stack([idx_vec[i] for i in range(idx)])
Y = np.stack([idx_sent[i] for i in range(idx)])
```

Regression classifiers: estimating and fit

1. Fit a logistic model (here w/ sklearn)
2. Predict on new data
3. That's it!

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(random_state=0, solver='lbfgs').fit(X, Y)

for test in [test1, test2, test3]:
    test_vec = np.zeros((1,V))
    indexes  = np.int32([vocab_idx[x] for x in split(test) if x in vocab_idx])
    indexes  = np.expand_dims(indexes,0)
    test_vec[0,indexes] = 1
    pred = clf.predict(test_vec)[0]
    prob = clf.predict_proba(test_vec)[0][0]
    print("Rating: %0.2f (%d%%)\nReview: %s\n" % (pred, prob*100, test))
```

```
Rating: -1.00 (77%)
Review: No one expects greatness from a movie called Dumb and Dumberer -- it would be dumb to expect much by way of humor or plot or character or energy -- but even so, this manages to be disappointing.

Rating: 1.00 (37%)
Review: This is one of the most searingly intense portraits of slavery ever committed to film, exercising onscreen brutality to bludgeon slavery's grim, cruel and conscience-less degradation.

Rating: 1.00 (8%)
Review: Peele has a solid script and a knack for storytelling in an appealing way which makes Get Out more enjoyable than many of the other suspense/thriller/horror movies being made today.
```

Regression classifiers: inspecting the model

We can easily check which words have the strongest effect on our classifier:

```
beta = clf.coef_.squeeze()
best_words  = ", ".join([idx_vocab[x] for x in np.argsort(beta)[::-1][0:50]])
worst_words = ", ".join([idx_vocab[x] for x in np.argsort(beta)[0:50]])
print("Most positive words: \n%s\n\nMost negative words: \n%s\n" % (best_words, worst_words))
```

```
Most positive words:
excellent, superb, perfect, perfectly, amazing, fantastic, atmosphere, wonderful, enjoyable, simple,
highly, favorite, today, great, loved, powerful, expecting, brilliant, surprised, jack, enjoyed, best,
unique, realistic, fun, episodes, manages, strong, easy, liked, beautiful, job, beauty, hilarious,
moving, bit, definitely, believable, buy, plenty, return, greatest, performances, unlike, present,
entertaining, love, worth, deal, tells

Most negative words:
waste, worst, poorly, awful, dull, mess, fails, badly, boring, annoying, avoid, lame, worse, poor,
terrible, horrible, save, unless, supposed, unfortunately, christmas, ridiculous, weak, disappointed,
bad, cheap, predictable, stupid, nothing, sorry, effort, attempt, silly, oh, crap, wonder, 4, premise,
1, basically, instead, money, okay, looks, bunch, minutes, dumb, nudity, what's, guess
```

Regression classifiers: inspecting the model

We can also check the effect of individual words in our sample review snippets:

```
word_effects = "\n\n".join([" ".join(["%s (%0.2f)" % (x, idx_beta[vocab_idx[x]]) if x in \
  vocab_idx else "%s (?)" % x for x in split(z) ]) for z in [test1, test2, test3]] )
print(word_effects)
```

```
no (-0.09) one (0.01) expects (?) greatness (?) from (0.00) a (-0.00) movie (-0.04) called (0.00) dumb
(-0.09) and (0.02) dumberer (?) it (0.04) would (-0.04) be (-0.02) dumb (-0.09) to (-0.06) expect (0.04)
much (-0.05) by (-0.01) way (0.01) of (-0.00) humor (0.05) or (-0.03) plot (-0.08) or (-0.03) character
(-0.00) or (-0.03) energy (?) but (-0.03) even (-0.07) so (-0.02) this (-0.05) manages (0.11) to (-0.06)
be (-0.02) disappointing (?)

this (-0.05) is (0.06) one (0.01) of (-0.00) the (0.05) most (0.03) searingly (?) intense (?) portraits
(?) of (-0.00) slavery (?) ever (0.03) committed (?) to (-0.06) film (0.02) exercising (?) onscreen (?)
brutality (?) to (-0.06) bludgeon (?) slavery's (?) grim (?) cruel (?) and (0.02) conscience-less (?)
degradation (?)

peele (?) has (0.03) a (-0.00) solid (?) script (-0.08) and (0.02) a (-0.00) knack (?) for (-0.00)
storytelling (?) in (0.01) an (-0.01) appealing (?) way (0.01) which (-0.00) makes (0.07) get (0.02) out
(-0.00) more (0.03) enjoyable (0.19) than (0.01) many (0.02) of (-0.00) the (0.05) other (0.01)
suspense/thriller/horror (?) movies (0.02) being (-0.01) made (0.00) today (0.17)
```

Regression classifiers: inspecting the model

no (-0.09) one (0.01) expects (?) greatness (?) from (0.00) a (-0.00) movie (-0.04) called (0.00) dumb (-0.09) and (0.02) dumberer (?) it (0.04) would (-0.04) be (-0.02) dumb (-0.09) to (-0.06) expect (0.04) much (-0.05) by (-0.01) way (0.01) of (-0.00) humor (0.05) or (-0.03) plot (-0.08) or (-0.03) character (-0.00) or (-0.03) energy (?) but (-0.03) even (-0.07) so (-0.02) this (-0.05) manages (0.11) to (-0.06) be (-0.02) disappointing (?)

this (-0.05) is (0.06) one (0.01) of (-0.00) the (0.05) most (0.03) searingly (?) intense (?) portraits (?) of (-0.00) slavery (?) ever (0.03) committed (?) to (-0.06) film (0.02) exercising (?) onscreen (?) brutality (?) to (-0.06) bludgeon (?) slavery's (?) grim (?) cruel (?) and (0.02) conscience-less (?) degradation (?)

peele (?) has (0.03) a (-0.00) solid (?) script (-0.08) and (0.02) a (-0.00) knack (?) for (-0.00) storytelling (?) in (0.01) an (-0.01) appealing (?) way (0.01) which (-0.00) makes (0.07) get (0.02) out (-0.00) more (0.03) enjoyable (0.19) than (0.01) many (0.02) of (-0.00) the (0.05) other (0.01) suspense/thriller/horror (?) movies (0.02) being (-0.01) made (0.00) today (0.17)

By the way...

1. Remember this is still just fitting points to a line…
2. Does OLS work?
3. Yes!

```
# you can always just do OLS
beta, residuals, rank, s = np.linalg.lstsq(X,Y)

# try it out on some text
test1 = "No one expects greatness from a movie called Dumb and Dumberer -- it w
test2 = "This is one of the most searingly intense portraits of slavery ever co
test3 = "Peele has a solid script and a knack for storytelling in an appealing
for test in [test1, test2, test3]:
    test_vec = np.zeros(V)
    indexes  = np.int32([vocab_idx[x] for x in split(test) if x in vocab_idx])
    test_vec[indexes] = 1
    print("Rating: %0.2f\nReview: %s\n" % (beta.dot(test_vec), test))
```

```
Rating: -0.40
Review: No one expects greatness from a movie called Dumb and Dumberer -- it would be dumb to expect much by way of
humor or plot or character or energy -- but even so, this manages to be disappointing.

Rating: 0.12
Review: This is one of the most searingly intense portraits of slavery ever committed to film, exercising onscreen
brutality to bludgeon slavery's grim, cruel and conscience-less degradation.

Rating: 0.55
Review: Peele has a solid script and a knack for storytelling in an appealing way which makes Get Out more
enjoyable than many of the other suspense/thriller/horror movies being made today.
```

Regression classifiers: inspecting the model

We can easily check which words have the strongest effect on our classifier:

```
beta = clf.coef_.squeeze()
best_words  = ", ".join([idx_vocab[x] for x in np.argsort(beta)[::-1][0:50]])
worst_words = ", ".join([idx_vocab[x] for x in np.argsort(beta)[0:50]])
print("Most positive words: \n%s\n\nMost negative words: \n%s\n" % (best_words, worst_words))
```

```
Most positive words:
excellent, superb, perfect, enjoyable, great, atmosphere, amazing, today, simple, loved, highly,
wonderful, best, favorite, perfectly, enjoyed, fun, jack, expecting, easy, surprised, fantastic, liked,
brilliant, documentary, realistic, moving, definitely, enjoy, job, manages, classic, buy, strong,
plenty, season, hilarious, bit, human, unlike, episodes, beautiful, entertaining, powerful, love,
surprise, unique, heart, dance, deal

Most negative words:
worst, waste, dull, fails, poorly, mess, awful, boring, annoying, badly, bad, poor, christmas,
unfortunately, avoid, lame, unless, worse, horrible, save, predictable, 4, terrible, ridiculous, effort,
weak, silly, sorry, nothing, disappointed, oh, supposed, premise, instead, wonder, stupid, attempt,
basically, nudity, cheap, laughs, guess, material, crap, none, attempts, what's, money, minutes, ok
```

Regression classifiers: evaluating performance

We can check the precision, recall and F1 score:

```
tp = np.sum([x == 1 and y == 1 for x,y in zip(true_values, pred_values)])
tn = np.sum([x != 1 and y != 1 for x,y in zip(true_values, pred_values)])
fp = np.sum([x != 1 and y == 1 for x,y in zip(true_values, pred_values)])
fn = np.sum([x == 1 and y != 1 for x,y in zip(true_values, pred_values)])
recall    = tp / (tp + fn)
precision = tp / (tp + fp)
F1        = 2 * (precision*recall)/(precision+recall)
```

OLS model:

precision: 0.84
recall: 0.87
F1: 0.85

Logistic model:

precision: 0.85
recall: 0.86
F1: 0.86

¯_(ツ)_/¯

Using more variables

Everything above was based on using just the first 1,000 words. What if we use more, say, 2,000 words?

```
OLS model:                  Logistic model:

    precision: 0.85 (+.01)      precision: 0.86 (+.01)
    recall: 0.87                recall: 0.87 (+.01)
    F1: 0.86     (+.01)         F1: 0.86
```

Can we just use all the words?

Yes! But if we have that many variables (in particular if ||w|| > n) we should enforce a penalty on the weights (a.k.a. coefficients).

```
Logistic model
(10k words w/o penalty):

    precision: 0.87 (+.02)
    recall: 0.86 (+.01)
    F1: 0.86
```

```
Logistic model
(10k words w/ l2 penalty):

    precision: 0.88 (+.03)
    recall: 0.88 (+.02)
    F1: 0.88 (+.02)
```

- Penalties set a limit on the total "size" of the vector of weights according to various metrics.

- L1 penalty = penalty based on sum of weights
- L2 penalty = penalty based on L2 norm (vector distance from mean)

- In least squares regression this is called ridge regression (L2 norm) or lasso regression (L1 norm)

Inspecting the model: weights with 1,000 words

no (-0.09) one (0.01) expects (?) greatness (?) from (0.00) a (-0.00) movie (-0.04) called (0.00) dumb (-0.09) and (0.02) dumberer (?) it (0.04) would (-0.04) be (-0.02) dumb (-0.09) to (-0.06) expect (0.04) much (-0.05) by (-0.01) way (0.01) of (-0.00) humor (0.05) or (-0.03) plot (-0.08) or (-0.03) character (-0.00) or (-0.03) energy (?) but (-0.03) even (-0.07) so (-0.02) this (-0.05) manages (0.11) to (-0.06) be (-0.02) disappointing (?)

this (-0.05) is (0.06) one (0.01) of (-0.00) the (0.05) most (0.03) searingly (?) intense (?) portraits (?) of (-0.00) slavery (?) ever (0.03) committed (?) to (-0.06) film (0.02) exercising (?) onscreen (?) brutality (?) to (-0.06) bludgeon (?) slavery's (?) grim (?) cruel (?) and (0.02) conscience-less (?) degradation (?)

peele (?) has (0.03) a (-0.00) solid (?) script (-0.08) and (0.02) a (-0.00) knack (?) for (-0.00) storytelling (?) in (0.01) an (-0.01) appealing (?) way (0.01) which (-0.00) makes (0.07) get (0.02) out (-0.00) more (0.03) enjoyable (0.19) than (0.01) many (0.02) of (-0.00) the (0.05) other (0.01) suspense/thriller/horror (?) movies (0.02) being (-0.01) made (0.00) today (0.17)

Inspecting the model: weights with 10,000 words

no (-0.30) one (-0.01) expects (-0.11) greatness (-0.12) from (-0.02) a (-0.14) movie (-0.10) called (0.09) dumb (-0.21) and (0.03) dumberer (?) it (0.13) would (-0.17) be (-0.06) dumb (-0.21) to (-0.13) expect (0.20) much (-0.18) by (-0.09) way (0.03) of (-0.13) humor (0.25) or (-0.11) plot (-0.29) or (-0.11) character (0.03) or (-0.11) energy (0.02) but (-0.07) even (-0.21) so (-0.04) this (-0.12) manages (0.37) to (-0.13) be (-0.06) disappointing (-0.83)

this (-0.12) is (0.15) one (-0.01) of (-0.13) the (-0.01) most (0.22) searingly (?) intense (0.37) portraits (?) of (-0.13) slavery (0.07) ever (0.11) committed (-0.06) to (-0.13) film (0.06) exercising (?) onscreen (?) brutality (-0.04) to (-0.13) bludgeon (?) slavery's (?) grim (0.17) cruel (-0.07) and (0.03) conscience-less (?) degradation (?)

peele (?) has (0.08) a (-0.14) solid (0.56) script (-0.40) and (0.03) a (-0.14) knack (0.01) for (-0.02) storytelling (-0.05) in (0.03) an (-0.06) appealing (-0.20) way (0.03) which (-0.03) makes (0.32) get (0.04) out (0.04) more (0.12) enjoyable (0.70) than (0.00) many (0.06) of (-0.13) the (-0.01) other (0.05) suspense/thriller/horror (?) movies (0.03) being (-0.02) made (0.03) today (0.64)

Inspecting the model: influential words w/ penalty

Most positive words:
7/10, excellent, perfect, 7, 8/10, refreshing, wonderfully, 8, superb, amazing, rare, 10/10, incredible, great, enjoyable, subtle, loved, funniest, favorite, wonderful, highly, perfectly, noir, today, appreciated, 9/10, gem, enjoyed, surprisingly, fantastic, simple, hooked, solid, best, delightful, vengeance, fun, atmosphere, available, beautifully, job, captures, recommended, haunting, spectacular, steals, underrated, definitely, flawless, liked

Most negative words:
worst, waste, 4/10, poorly, awful, disappointment, dull, boring, 3/10, disappointing, lacks, fails, worse, mess, avoid, terrible, laughable, annoying, forgettable, unfunny, poor, 2/10, bad, horrible, badly, 1/10, mst3k, pointless, lame, lousy, save, unfortunately, weak, supposed, dreadful, obnoxious, wooden, mildly, redeeming, ridiculous, oh, unless, predictable, stupid, pretentious, nothing, wonder, pathetic, endless, uninteresting

Inspecting the model: influential words w/o penalty

Most positive words:
7/10, 8/10, refreshing, vengeance, appreciated, hooked, cerebral, 7, chavez, excellent, scariest, 9/10, 10/10, noir, point.<br, kitty, delightful, wonderfully, rare, subtle, cracking, funniest, incredible, hoot, superb, perfect, perfectly, fears, 8, haunting, driven, fashioned, enjoyable, joan, steals, modesty, batman, lonely, whoopi, essential, gently, captures, superbly, relax, apartment, available, definite, highly, someday, flawless

Most negative words:
4/10, 3/10, worst, waste, poorly, disappointment, 1/10, 2/10, unfunny, mst3k, forgettable, lacks, awful, baldwin, refer, laughable, obnoxious, sources, dreadful, tunnel, lifeless, unwatchable, outer, lousy, avoid, boredom, cardboard, mess, fails, wayans, mildly, tiresome, ludicrous, thunderbirds, pistol, photo, trite, pass, dumber, disappointing, dull, worse, fodder, flimsy, stupidity, credibility, uninteresting, huston, boring, endless