CS5785 Homework 2

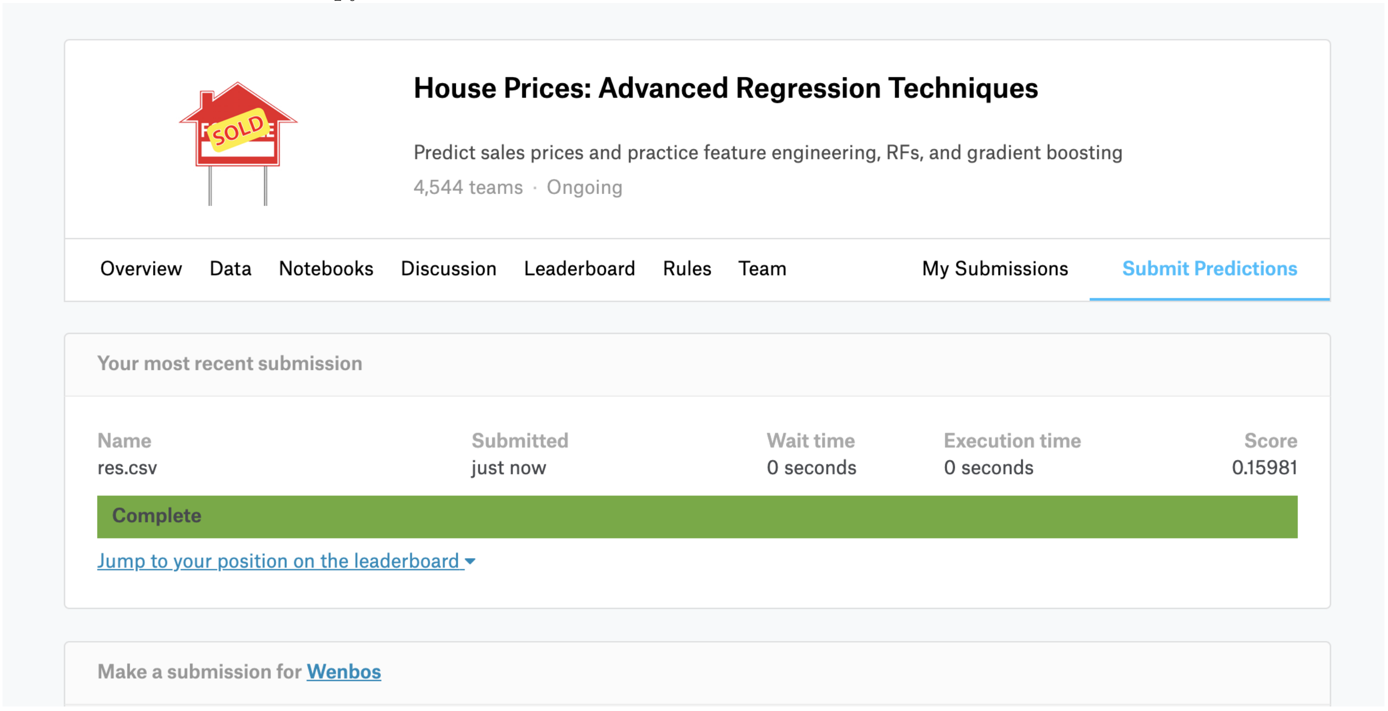
Wenbo Sun, Nayun Xu

October 3, 2019

1 PROGRAMMING EXERCISES:

1.1 Answer to exercise 1

The answers are in the Jupyter Notebook.



House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting

4,544 teams · Ongoing

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
res.csv	just now	0 seconds	0 seconds	0.15981

Complete

Jump to your position on the leaderboard

Make a submission for Wenbos

1.2 Answer to exercise 2

a) All the labels are balanced. I parse the data with the code below, and then count the numbers of positive and negative labels.

```
texts = []
labels = []
for eachLine in fr:
    items = eachLine.rstrip('\n').split('\t')
    texts.append(preprocess(items[0], stopwords))
    labels.append(int(items[1]))
```

b) I did the following preprocessing methods.

- Lowercase all the words. Upper or lowercase words share the same underlined meaning which I think should be considered as the same word.
- Lemmatization of all the words. There are two reasons. The first reason is that the dataset is small, there may not be enough words to let the model learn the similarity between such similar words (with the same root) like 'good' and 'better'. The second reason is that, our goal is to distinguish between positive and negative sentiments, in which case converting the words to root forms will not affect the 'positive vs negative' choice.
- Strip punctuation. I think punctuation does not provide much information about positive or negative sentiments.
- Strip the stop words. The stop words don't hold actual meaning that reveal the sentiments.

d) We can't use the test data to build the dictionary because in practice or application, we will not know the exact text data when we are in the stage of building the dictionary.

I randomly choose the two reviews in the training set.

review 1: So absolutley fantastic.
feature 1 with 1-gram: [(486, 1), (2228, 1)]
feature 1 with 2-gram: [(5179, 1.0)]
review 2: but the movie makes a lot of serious mistakes.
feature 2 with 1-gram: [(79, 1), (295, 1), (646, 1), (2609, 1), (3706, 1)]
feature 2 with 2-gram: [(8569, 1.0), (9646, 1.0), (9647, 1.0), (9648, 1.0)]
Because the feature is high dimensional and highly sparse, I use the sparse representation of a list of tuples. Each tuple t represents a non-zero element in the feature vector where $t[0]$ is the column number and $t[1]$ is the value

e) I choose the log-normalization which is $f(x) = \log(x+1)$. Because first the curve of $\log(x+1)$ increases more and more slowly when x grows which is consistent with the intuition that 10 'good' and 11 'good' does not make the same difference as 0 'god' and 1 'good'. And compared to other methods, such normalize keeps the instance-wise order of each feature value. I do the same for both 1-gram and 2-gram model.

f) The 1-gram accuracy is 0.7017. And the confusion matrix is shown below in Tab 1.

The 2-gram accuracy is 0.6333. And the confusion matrix is shown below in Tab 2.

	label 0	label 1
predict 0	261	140
predict 1	39	160

Table 1: Confusion matrix of 1-gram, Naive Bayes

	label 0	label 1
predict 0	269	189
predict 1	31	111

Table 2: Confusion matrix of 2-gram, Naive Bayes

g) I used a 5-fold cross validation to pick best regularization term for penalty. For 1-gram model, the accuracy of using $l1$ penalty is 0.7983, and using $l2$ penalty is 0.8083. For 2-gram model, the accuracy of using $l1$ penalty is 0.6183 and using $l2$ penalty is 0.6383. By showing the top 20 values with largest absolute value, calculating the number of 0 elements in the coefficient vector, calculating the mean of non-zero elements, we find that $l1$ penalty makes the coefficient vector have more 0 elements, and $l2$ penalty makes the coefficient elements relatively small.

For $l1$ penalty and 1-gram model, the 20 most important words according to the absolute value of the corresponding coefficient are: ['arent'] ['delicious'] ['disappointment'] ['loved'] ['poor'] ['interesting'] ['wouldve'] ['great'] ['rude'] ['beautiful'] ['awesome'] ['unfortunately'] ['excellent'] ['wonderful'] ['failed'] ['fantastic'] ['glad'] ['love'] ['liked'] ['role']
For $l2$ penalty and 1-gram model, the 20 most important words according to the absolute value of the corresponding coefficient are: ['great'] ['bad'] ['love'] ['delicious'] ['poor'] ['excellent'] ['nice'] ['loved'] ['fantastic'] ['worst'] ['awesome'] ['disappointment'] ['liked'] ['beautiful'] ['amazing'] ['terrible'] ['awful'] ['interesting'] ['stupid'] ['wonderful']

For $l1$ penalty and 2-gram model, the 20 most important symbols according to the absolute value of the corresponding coefficient are: ['money' 'game'] ['give' '2'] ['think' 'disappointed'] ['plug' 'work'] ['reversible' 'plug'] ['poor' 'sound'] ['great' 'phone'] ['best' 'phone'] ['life' 'real'] ['recommend' 'place'] ['wont' 'regret'] ['happy' 'product'] ['good' 'price'] ['easy' 'use'] ['great' 'product'] ['love' 'phone'] ['good' 'though'] ['loved' 'place'] ['excellent' 'performance'] ['food' 'amazing']

For $l2$ penalty and 2-gram model, the 20 most important symbols according to the absolute value of the corresponding coefficient are: ['work' 'great'] ['highly' 'recommend'] ['waste' 'time'] ['great' 'phone'] ['one' 'best'] ['great' 'product'] ['food' 'good'] ['easy' 'use'] ['really' 'good'] ['dont' 'waste'] ['good' 'price'] ['dont' 'buy'] ['great' 'food'] ['food' 'delicious'] ['wont' 'back'] ['poor' 'quality'] ['great' 'service'] ['5' 'star'] ['love' 'phone'] ['stay' 'away']

We can find that the adjectives which show obvious positive or negative sentiments are most important. We can also find that the 2-gram model captured the 'won't regret' which uses double negative.

h) See above.

i) Logistic regression with 1-gram performs the best.Because the 2-gram BOW produces too many zero elements in the test data, in other words, many reviews in the test data are transformed to all-zero feature vectors which contains no information for the classifier to use.
I learnt that people will use positive adjective like 'great', 'delicious', 'nice', positive verbs like 'love' and double negative phrase like 'won't regret' to express positive sentiments. And negative adjectives like 'disappointment', 'poor', 'rude', negative verbs like 'aren't', 'failed' to express negative sentiments.

2 WRITTEN EXERCISES:

1) OLS on $\mathbf{X}_{aug}$ and $\mathbf{Y}_{aug}$ is to:

$$\min_{\beta \in \mathcal{R}^{1+p}} \sum_{i=1}^{n+p} (\mathbf{Y}_{aug_i} - \beta^T \mathbf{X}_{aug_i})^2$$

which is equal to

$$\min_{\beta \in \mathcal{R}^{1+p}} \sum_{i=1}^{n} (\mathbf{Y}_{aug_i} - \beta^T \mathbf{X}_{aug_i})^2 + \sum_{j=n+1}^{n+p} (\mathbf{Y}_{aug_j} - \beta^T \mathbf{X}_{aug_j})^2$$

since for $n+1 \leq j \leq n+p$, $\mathbf{Y}_{aug_j} = 0$ and $\beta^T \mathbf{X}_{aug_j} = \beta_{j-n}\sqrt{\lambda}$, the second term is equal to $\sum_{j=n+1}^{n+p} (\beta_{j-n}\sqrt{\lambda})^2 = \lambda \sum_{j=1}^{p} {\beta_j}^2$. Therefore we have

$$\min_{\beta \in \mathcal{R}^{1+p}} \sum_{i=1}^{n} (\mathbf{Y}_i - \beta^T \mathbf{X}_i)^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2$$

The augment new data points will only fit perfectly (contribute 0 loss) when the corresponding dimension of β is zero, and larger the $abs(\beta)$ is, larger the loss is, which intuitively serves as penalty for β.

2) a) We have

$$P(Y = H | F = T, H = F) = P(Y = H, F = T, H = F)/P(F = T, H = F) = 3/(6+3) = 1/3$$

$$P(Y = F | F = T, H = F) = P(Y = F, F = T, H = F)/P(F = T, H = F) = 6/(6+3) = 2/3$$

$$P(Y = P|F = T, H = F) = P(Y = P, F = T, H = F)/P(F = T, H = F) = 0/(6 + 3) = 0$$

b) Use a naive Bayes classifier, we have

$$P(Y = H|F = T, H = F) = P(F = T, H = F|Y = H)P(Y = H)/P(F = T, H = F)$$

$$\propto P(F = T|Y = H)P(H = F|Y = H)/P(F = T, H = F) = 5/70 * 61/60 * 70/100/P(F = T, H = F)$$

Similarly we have

$$P(Y = F|F = T, H = F) \propto 15/20 * 8/20 * 20/100/P(F = T, H = F)$$

$$P(Y = P|F = T, H = F) \propto 5/10 * 1/10 * 10/100/P(F = T, H = F)$$

By the we assign 0.046 to Pneumonia, 0.4 to Healthy , 0.55 to Flu.

3) The Naive Bayes classifier is based on:

$$p(Y = j|X = (x_1, x_2, x_3, ..., x_n)) \propto p(Y = j) \prod_{i=1}^{n} p(X_i = x_i|Y = j)$$

In this problem, x_1 is age, x_2 is income, x_3 is student, x_4 is credit-rating, and Y is buys-computer.

$$p(Y = \text{yes}) = \frac{9}{14}, p(Y = \text{no}) = \frac{5}{14}$$

.

$$p(x_1 = \leq 30|Y = \text{yes}) = \frac{2}{9}, p(x_1 = \leq 30|Y = \text{no}) = \frac{3}{5}$$
$$p(x_2 = \text{medium}|Y = \text{yes}) = \frac{4}{9}, p(x_2 = \text{medium}|Y = \text{no}) = \frac{2}{6}$$
$$p(x_3 = \text{yes}|Y = \text{yes}) = \frac{2}{3}, p(x_3 = \text{yes}|Y = \text{no}) = \frac{1}{5}$$
$$p(x_3 = \text{fair}|Y = \text{yes}) = \frac{2}{3}, p(x_3 = \text{fair}|Y = \text{no}) = \frac{2}{5}$$

Since $\frac{9*2*4*2*2}{14*9*9*3*3} > \frac{5*3*2*1*2}{14*5*6*5*5}$, the Naive Bayes classifier will predict this example as 'yes'.