Text Classification with Python

Majid Hajiheidari

April 18, 2019

Divar Posts Dataset

post2.png

- Released for DataDays 2019
- One million posts

Columns

- id
- archive_by_user
- published_at
- **cat1**
- **cat2**
- **cat3**
- city
- **title**
- **desc**
- price
- image_count
- platform
- mileage
- brand
- year
- type

The Problem: Categorization

- We need to categorize posts based on other posts features;
- We only use text features(title & description)!

Temp Frame

This slide is temp.

Temp Frame

This slide is temp.

First Approach: Naive Bayes Classifier

bayes_formula.jpg

Thomas_Bayes.png

Photo by Matt Buck

Vectorizing the Text: Count Vectorizer

An example: We want to vectorize these 4 setences[1]:

1. Hello, how are you!
2. Win money, win from home.
3. Call me now
4. Hello, Call you tomorrow?

[1] Example from Rahul Vasaikar

Vectorizing the Text: Count Vectorizer

1. We first build a vocabulary:
   $vocabulary =$
   $\{are, call, from, hello, home, how, me, money, now, tomorrow, win, you\}$
2. Then, we vectorize each sentence based on the occurness of each word:

	are	call	from	hello	home	how	me	money	now	tom...	win	you
1	1	0	0	1	0	1	0	0	0	0	0	1
2	0	0	1	0	1	0	0	1	0	0	2	0
3	0	1	0	0	0	0	1	0	1	0	0	0
4	0	1	0	1	0	0	0	0	0	1	0	1

Vectorizing the Text: Count Vectorizer

N pair of samples

Bayes Classifier: Naive One!

It is possible to show that accuracy is minimized, on average, by a very simple classifier that assigns each observation to the most likely class, given its predictor values. In other words, we should simply assign a test observation with predictor vector x_0 to the class j for which

$$P\left(Y = j \mid \mathbf{X} = \mathbf{x}\right)$$

is largest.

Bayes Classifier: Naive One!

We make two assumptions:

1. $X_1, X_2, \ldots, and\ X_m$ are independent from each other;
2. $X_1, X_2, \ldots, X_m \mid Y \sim MN(\cdot, p_1, p_2, \ldots, p_m)$

$$P(Y = j \mid \mathbf{X} = (x_1, x_2, \ldots, x_m)) = \frac{P(\mathbf{X} = (x_1, x_2, \ldots, x_m) \mid Y = j) \cdot P(Y = j)}{P(\mathbf{X} = \mathbf{x})}$$
$$= \frac{P(X_1 = x_1 \mid Y = j) \cdot \ldots \cdot P(X_m = x_m \mid Y = j) \cdot P(Y = j)}{P(\mathbf{X} = \mathbf{x})}.$$

$$\hat{y} = \underset{j \in classes}{\arg\max} \frac{P(X_1 = x_1 \mid Y = j) \cdot \ldots \cdot P(X_m = x_m \mid Y = j) \cdot P(Y = j)}{P(\mathbf{X} = \mathbf{x})}$$
$$= \underset{j \in classes}{\arg\max} P(X_1 = x_1 \mid Y = j) \cdot \ldots \cdot P(X_m = x_m \mid Y = j) \cdot P(Y = j).$$

Bayes Classifier: Naive One!

Let's dive into code!

Hyperparameters

Two important hyperparameters:

1. Size of the vocabulary;
2. Laplace/ Lidstone smoothing parameter(α).

Size of Vocabulary

plot1.png

Laplace/ Lidstone Smoothing Parameter(α)

plot1.png

Grid Search

Temp Frame

This slide is temp.

Temp Frame

This slide is temp.

Temp Frame

This slide is temp.

Temp Frame

This slide is temp.

Thanks for your attention!

Codes in slides (in my GitHub):(github link)
Divar posts dataset:(divar link)
Any questions?