GENERAL ASSEMBLY

Naive Bayes

Patrick Smith

OPENING

So….*What is* Naive Bayes?

Naive Bayes

Naive Bayes is a method that uses the probabilities of all of the attributes of each class in a dataset to make a prediction.

Naive Bayes

Naive Bayes is a method that uses the probabilities of all of the attributes of each class in a dataset to make a prediction.

We use Naive Bayes to model a predictive problem *utilizing probability*

Naive Bayes

If we find the probability of each attribute in a class, the probability of the class given the probability of the attribute is called *conditional probability*

Naive Bayes

If we find the probability of each attribute in a class, the probability of the class given the probability of the attribute is called *conditional probability*

Naive Bayes utilizes this conditional probability

Naive Bayes

If we find the probability of each attribute in a feature, the probability of the feature given the probability of the attribute is called *conditional probability*

Naive Bayes utilizes this conditional probability

Naive Bayes

With Naive Bayes, the *naive* comes in with the assumption that there is independence between our features

Naive Bayes

With Naive Bayes, the *naive* comes in with the assumption that there is independence between our features

Circling back to conditional probability, with Naive Bayes we “un-link” the class conditional probability distributions

Naive Bayes

Coming full-circle, it all comes back to Bayes Rule

Naive Bayes: Deep Dive

Naive Bayes

$$ P(Class = i) = \frac{P(Event_1|Class = i)P(Event_2|Class = i)...P(Event_n|Class = i)}{P(Event)},$$

Naive Bayes

$$ P(Class = i) = \frac{P(Event_1|Class = i)P(Event_2|Class = i)...P(Event_n|Class = i)}{P(Event)},$$

for each i in the set of classes (again, if we had only two classes, i = 0 or i = 1), and where Event_1, ..., Event_n forms a partition of Event. We've seen this before, in the first lesson of the week.

Naive Bayes

To circle back and review, let's take an example, we can use canonical Spam classifier problem:

$$\Pr(S|W) = \frac{\Pr(W|S) \cdot \Pr(S)}{\Pr(W|S) \cdot \Pr(S) + \Pr(W|H) \cdot \Pr(H)}$$

Naive Bayes

This is the ordinary posterior distribution. Note the denominator is just the total probability. We can interpret the various components as follows;

- `P(S|W)` : Probability that Message is spam given word W occurs in it.
- `P(W|S)` : Probability that word W occurs in a spam message.
- `P(W|H)` : Probability that word W occurs in a Ham message.

Naive Bayes

- **BernoulliNB** is designed for binary/boolean features

- The **Multinomial Naive Bayes classifier** is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work

- **GaussianNB** is designed for continuous features (that can be scaled between 0,1) and is assumed to be normally distributed

Using Naive Bayes in SciKit

We've gone over the formalism of Bayesian analysis several times now, so we should be safe there. Let's get more hands-on work with analyzing Naive Bayes for computing.

Write your own Naive Bayes classifier

Naive Bayes: Insult Demo/Lab

Naive Bayes

We're going to be looking at comments like this:

Moon Master99BBQ
Insult Connoisseur

"You're all upset, defending this hipster band...and WE'RE the douches for reading the news and discussing it? Put down the PBR, throw away the trucker hat, shave off that silly shadow-beard, put down your "99%er" sign, and get a job, ION."

1. Explore a list of comment words that occur more than 50x

Naive Bayes

1. Explore a list of comment words that occur more than 50x

1.5 Try it again with stopword removal

2. Explore ngrams between 2 and 4

3. Try expanding the list of stopwords

4. Setup a test / train split of your data using any method you wish.

5. Setup a "Pipeline" to vectorize and use MultinomialNB classifier.

5.5 Swap out MultinomialNB with BernoulliNB in the pipeline

5.5b Also try tweeking the paramters of CountVectorizer and TfidfTranformer.

6. Check your score.

Naive Bayes

Naive Bayes

We touched on this briefly in the past but let's reprise the idea of sample size effect on validation score. How do we know the optimal sample size to train and test on?

We can examine the scores of training and cross validation given a number of samples. Plotting the scores is a great way to understand:

Naive Bayes

- How to improve bias / generalization (out of sample prediction)
- Generally how many samples you might need
- The bounds of your models performance

Generally, the learning curves represent the number of samples that have been used, the average scores on the training sets and the average scores on the validation sets.

7. Check the accuracy of your model with the holdout dataset "test_with_solutions.csv"

8. What is your model not getting right?

Conclusion

Q & A