Microsoft Learn
Student Ambassadors

ML Bootcamp Session 9 : Supervised Learning – Naïve Bayes

Natural Language Processing

Field of Machine Learning dealing with linguistics that builds and develops **Language Models**.

Language Modeling determines the **likelihood** of word sequences occurring in a sentence via probabilistic and statistical techniques.

NLP and Classification

Spam Filter

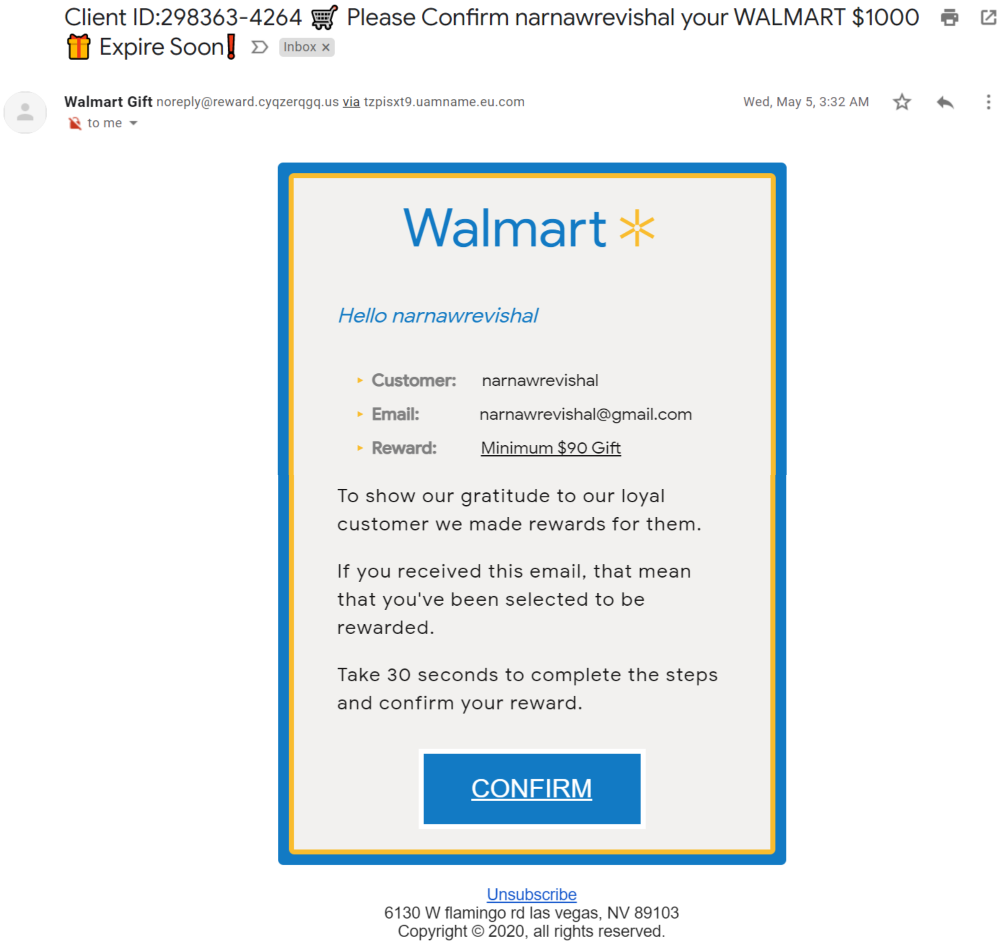

Image from Vishal Narnaware's Gmail spam folder

NLP and Classification

Positive or Negative Review

Our goodwill and sense of nostalgia for the Warrens goes only so far in this third film.

June 2, 2021 | Rating: 2/4 | Full Review...

Tomris Laffly
RogerEbert.com
TOP CRITIC

For the many reasons that this franchise works, Farmiga and Wilson are chief among them, as they take what could otherwise be hokey 1970s ghost-hunter characters and infuse them with a deep sense of faith, humanity, and above all, love.

June 1, 2021 | Rating: 3/4 | Full Review...

Katie Walsh
Tribune News Service
TOP CRITIC

Patrick Wilson and Vera Farmiga are excellent in these roles, and -- just like a good TV show -- I will watch these characters do things, even when some episodes aren't as good as the others.

June 2, 2021 | Rating: B | Full Review...

Chris Stuckmann
ChrisStuckmann.com
TOP CRITIC

Chaves takes the reins from Wan and runs with it, bringing a wholly different experience that still feels like a warm reunion with horror's favorite couple.

June 1, 2021 | Rating: 4/5 | Full Review...

Meagan Navarro
Bloody Disgusting
TOP CRITIC

Source: Rotten Tomatoes, CRITIC REVIEWS FOR *THE CONJURING: THE DEVIL MADE ME DO IT*

NLP and Classification

Paper Tagging

[1] arXiv:2106.00672 [pdf, other]

What Matters for Adversarial Imitation Learning?

Manu Orsini, Anton Raichuk, Léonard Hussenot, Damien Vincent, Robert Dadashi, Sertan Girgin, Matthieu Geist, Olivier Bachem, Olivier Pietquin, Marcin Andrychowicz

Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI); Neural and Evolutionary Computing (cs.NE)

[2] arXiv:2106.00660 [pdf, other]

Markpainting: Adversarial Machine Learning meets Inpainting

David Khachaturov, Ilia Shumailov, Yiren Zhao, Nicolas Papernot, Ross Anderson

Comments: Proceedings of the 38th International Conference on Machine Learning (ICML 2021)

Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI); Cryptography and Security (cs.CR); Computer Vision and Pattern Recognition (cs.CV); Computers and Society (cs.CY)

[3] arXiv:2106.00654 [pdf, other]

A reinforcement learning approach to improve communication performance and energy utilization in fog-based IoT

Babatunji Omoniwa, Maxime Gueriau, Ivana Dusparic

Comments: Submitted and published in IEEE proceedings

Subjects: **Machine Learning (cs.LG)**; Networking and Internet Architecture (cs.NI); Signal Processing (eess.SP)

[4] arXiv:2106.00651 [pdf, other]

Asymptotics of representation learning in finite Bayesian neural networks

Jacob A. Zavatone-Veth, Abdulkadir Canatar, Cengiz Pehlevan

Comments: 12+28 pages, 2+1 figures

Subjects: **Machine Learning (cs.LG)**; Disordered Systems and Neural Networks (cond-mat.dis-nn); Machine Learning (stat.ML)

[5] arXiv:2106.00638 [pdf, other]

Quantifying Predictive Uncertainty in Medical Image Analysis with Deep Kernel Learning

Zhiliang Wu, Yinchong Yang, Jindong Gu, Volker Tresp

Subjects: **Machine Learning (cs.LG)**; Computer Vision and Pattern Recognition (cs.CV)

Source: arXiv, Machine Learning subdomain in Computer Science

Text Representation

Bag of Words

```
Doge to the moon
to the moon to the moon
To the moon of
I know the way
By the way in the sea
To the ocean on the horizon
To the sky
To me on the earth by the waves on the shore
I know a land
On a sea
So near to the sea's shore
And in a land
So far
So far
So far...
Let's go on to sky.
```

Q: How to convert this to model recognizable format?

A: Create a vocabulary of (say) most 20 recurring words (Tokens), then create a vector having entries number of times that word appeared.

Text: DeepAI Text Generation API (based on GPT-2 model by OpenAI)

Text Representation

Bag of Words

```
Doge to the moon
to the moon to the moon
To the moon of
I know the way
By the way in the sea
To the ocean on the horizon
To the sky
To me on the earth by the waves on the shore
I know a land
On a sea
So near to the sea's shore
And in a land
So far
So far
So far...
Let's go on to sky.
```

20 Most Recurring words:
the to on moon so a far i know way by in sea shore land doge of ocean horizon sky

```python
from collections import Counter

txt_low = txt.lower().replace('\'', '').replace('\n', '').replace('.', '')
word_lst = txt_low.split(' ')

wrdcounts = Counter(word_lst)

lst = sorted(wrdcounts.items(), key=lambda x:x[1], reverse=True)
sortwords = dict(lst)

vocab_size = 20
for (key, value) in sortwords.items():
    if vocab_size > 0:
        print(f'{key}: {value}')
        vocab_size -= 1
    else:
        break
```

Text: DeepAI Text Generation API (based on GPT-2 model by OpenAI)

Text Representation

Bag of Words

20 Most Recurring words: the to on moon so a far i know way by in sea shore land doge of ocean horizon sky

```
Doge to the moon
to the moon to the moon
To the moon of
I know the way
By the way in the sea
To the ocean on the horizon
To the sky
To me on the earth by the waves on the shore
I know a land
On a sea
So near to the sea's shore
And in a land
So far
So far
So far...
Let's go on to sky.
```

$$\langle x_1 \quad x_2 \quad x_3 \quad x_4 \quad \dots \quad x_{19} \quad x_{20} \rangle$$

$$\langle 1 \quad 1 \quad 0 \quad 1 \quad \dots \quad 0 \quad 0 \rangle$$

$$\langle 2 \quad 2 \quad 0 \quad 2 \quad \dots \quad 0 \quad 0 \rangle$$

$$\vdots$$

$$\langle 3 \quad 1 \quad 2 \quad 0 \quad \dots \quad 0 \quad 0 \rangle$$

$$\vdots$$

$$\langle 0 \quad 1 \quad 1 \quad 0 \quad \dots \quad 0 \quad 1 \rangle$$

Text: DeepAI Text Generation API (based on GPT-2 model by OpenAI)

Bayes' Theorem

"Bayes' theorem is to the theory of probability what Pythagoras's theorem is to geometry."

- Sir Harold Jeffreys

Bayes' Theorem

$Consider\ \boldsymbol{x} = \langle x_1 \quad x_2 \quad \cdots \quad x_n \rangle$
$and\ y = \{0, 1\}$

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}, y)}{P(\boldsymbol{x})}$$

$Or,$

$$P(y|\boldsymbol{x}) \propto \boxed{P(\boldsymbol{x}, y)}$$

$Now, we\ just\ need\ to\ calculate\ P(\boldsymbol{x}, y)$

$$P(\boldsymbol{x}, y) = P(x_1,\ x_2,\ x_3, \cdots,\ x_n, y)$$

$$= \underbrace{\boxed{P(x_1|x_2, x_3, \cdots, x_n, y)}}_{2^{n-1}K} \times \underbrace{\boxed{P(x_2|x_3, \cdots, x_n, y)}}_{2^{n-2}K} \times \underbrace{\cdots}_{\cdots} \times P(x_n|y) \times P(y)$$

$By\ Chain\ Rule\ of\ Probability$

$Curse\ of\ Dimensionality!$

Naïve Bayes

The Naïve Assumption

$$P(\boldsymbol{x}, y) = P(x_1|x_2, x_3, \cdots, x_n, y) \times P(x_2|x_3, \cdots, x_n, y) \times \cdots \times P(x_n|y) \times P(y)$$

$Assume\ features\ are\ conditionally\ independent,$
$Thus,$

$$P(\boldsymbol{x}, y) = P(x_1|y) \times P(x_2|y) \times \cdots \times P(x_n|y) \times P(y)$$

$Or,$

$$P(y|\boldsymbol{x}) \propto P(y) \times \prod_{i=1}^{n} \boxed{P(x_i|y)}$$

$$Posterior \propto Prior \times Likelihood$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

The choice of how we model these probabilities leads to the different implementations of naive Bayes classifiers.

Naïve Bayes

Next Steps: Maximum Likelihood Estimate

$$Make\ assumption\ regarding\ distribution\ of\ P(x_i|y)$$

$$Calculate\ Likelihood\ Function$$

$$Use\ Maximum\ Likelihood\ Estimation\ to\ find\ best\ parameters$$

$$Proof\ skipped$$

$$Use\ Laplace\ or\ Lidstone\ Smoothing$$

$$Will\ get\ clearer\ with\ an\ example\ \ldots$$

Naïve Bayes

Example: Spam Classifier

$$P(y_j) = \frac{messagecount(y = y_j)}{N_{message}}$$

$$P(w_i|y_j) = \frac{count(w_i, y_j)}{\sum_{w \in V} count(w, y_j)}$$

	Doc	Words	Class
Training	1	Drugs Buy Drugs	Spam
	2	Drugs drugs buy	Spam
	3	Drugs Now!	Spam
	4	Don’t take drugs	Ham
Test	5	Drugs sale now!	?

Naïve Bayes

Example: Spam Classifier

	Doc	Words	Class
Training	1	Drugs Buy Drugs	Spam
	2	Drugs drugs buy	Spam
	3	Drugs Now!	Spam
	4	Don't take drugs	Ham
Test	5	Drugs sale now!	?

$$P(y_j) = \frac{messagecount(y = y_j)}{N_{message}}$$

$$P(w_i|y_j) = \frac{count(w_i, y_j)}{\sum_{w \in V} count(w, y_j)}$$

$\boldsymbol{Priors}$:

$$P(Spam) = \frac{3}{4}$$

$$P(Ham) = \frac{1}{4}$$

$\boldsymbol{Conditional\ Probabilities}$:

$$P(drugs|Spam) = \frac{5}{8} \qquad P(drugs|Ham) = \frac{1}{3}$$

$$P(sale|Spam) = \frac{0}{8} \qquad P(sale|Ham) = \frac{0}{3}$$

$$P(now|Spam) = \frac{1}{8} \qquad P(now|Ham) = \frac{0}{3}$$

$\boldsymbol{Choosing\ Class}$:

$$P(Spam|d5) \propto \frac{3}{4} \times \frac{5}{8} \times \frac{0}{8} \times \frac{1}{8}$$

$$P(Ham|d5) \propto \frac{1}{4} \times \frac{1}{3} \times \frac{0}{3} \times \frac{0}{3}$$

Naïve Bayes

Example: Spam Classifier

	Doc	Words	Class
Training	1	Drugs Buy Drugs	Spam
	2	Drugs drugs buy	Spam
	3	Drugs Now!	Spam
	4	Don’t take drugs	Ham
Test	5	Drugs sale now!	?

$$P(y_j) = \frac{messagecount(y = y_j)}{N_{message}}$$

$$P(w_i|y_j) = \frac{count(w_i, y_j)}{\sum_{w \in V} count(w, y_j)}$$

$\boldsymbol{Priors}$:

$$P(Spam) = \frac{3}{4}$$

$$P(Ham) = \frac{1}{4}$$

$\boldsymbol{Conditional\ Probabilities}$:

$$P(drugs|Spam) = \frac{5}{8} \qquad P(drugs|Ham) = \frac{1}{3}$$

$$P(sale|Spam) = \frac{0}{8} \qquad P(sale|Ham) = \frac{0}{3}$$

$$P(now|Spam) = \frac{1}{8} \qquad P(now|Ham) = \frac{0}{3}$$

$\boldsymbol{Choosing\ Class}$:

$$P(Spam|d5) \propto 0$$

$$P(Ham|d5) \propto 0$$

Decide Randomly?

Naïve Bayes

Laplace Smoothing

Statistically, bad idea to assign 0 probability just because it is not present in training data

Solution: Add 1 to every word.

$$P(y_j) = \frac{messagecount(y = y_j)}{N_{message}}$$

$$P(w_i|y_j) = \frac{n_k + 1}{n + |Vocabulary|}$$

Naïve Bayes

Laplace and Lidstone Smoothing

Statistically, bad idea to assign 0 probability just because it is not present in training data

Solution: Add 1 to every word.

$$P(y_j) = \frac{messagecount(y = y_j)}{N_{message}}$$

$$P(w_i|y_j) = \frac{n_k + 1}{n + |Vocabulary|}$$

Or more generally,

$$P(w_i|y_j) = \frac{n_k + \alpha}{n + \alpha|Vocabulary|} \qquad Lidstone\ Smoothing$$

Naïve Bayes

Example: Spam Classifier (Revisited)

	Doc	Words	Class
Training	1	Drugs Buy Drugs	Spam
	2	Drugs drugs offer	Spam
	3	Drugs Now!	Spam
	4	Don’t take drugs	Ham
Test	5	Drugs sale now!	?

$$P(y_j) = \frac{messagecount(y = y_j)}{N_{message}}$$

$$P(w_i|y_j) = \frac{count(w_i, y_j) + 1}{\sum_{w \in V} count(w, y_j) + |V|}$$

Priors:

$$P(Spam) = \frac{3}{4}$$

$$P(Ham) = \frac{1}{4}$$

Conditional Probabilities:

$$P(drugs|Spam) = \frac{(5+1)}{(8+6)}$$

$$P(sale|Spam) = \frac{(0+1)}{(8+6)}$$

$$P(now|Spam) = \frac{(1+1)}{(8+6)}$$

$$P(drugs|Ham) = \frac{(1+1)}{(3+6)}$$

$$P(sale|Ham) = \frac{(0+1)}{(3+6)}$$

$$P(now|Ham) = \frac{(0+1)}{(3+6)}$$

Choosing Class:

$$P(Spam|d5) \propto \frac{3}{4} \times \frac{6}{14} \times \frac{1}{14} \times \frac{2}{14}$$

$$P(Ham|d5) \propto \frac{1}{4} \times \frac{2}{9} \times \frac{1}{9} \times \frac{1}{9}$$

Naïve Bayes

Example: Spam Classifier (Revisited)

	Doc	Words	Class
Training	1	Drugs Buy Drugs	Spam
	2	Drugs drugs offer	Spam
	3	Drugs Now!	Spam
	4	Don't take drugs	Ham
Test	5	Drugs sale now!	?

$$P(y_j) = \frac{messagecount(y = y_j)}{N_{message}}$$

$$P(w_i|y_j) = \frac{count(w_i, y_j) + 1}{\sum_{w \in V} count(w, y_j) + |V|}$$

Priors:

$$P(Spam) = \frac{3}{4}$$

$$P(Ham) = \frac{1}{4}$$

Conditional Probabilities:

$$P(drugs|Spam) = \frac{(5+1)}{(8+6)} \qquad P(drugs|Ham) = \frac{(1+1)}{(3+6)}$$

$$P(sale|Spam) = \frac{(0+1)}{(8+6)} \qquad P(sale|Ham) = \frac{(0+1)}{(3+6)}$$

$$P(now|Spam) = \frac{(1+1)}{(8+6)} \qquad P(now|Ham) = \frac{(0+1)}{(3+6)}$$

Choosing Class:

$$\boxed{P(Spam|d5) \propto 0.0033}$$

$$P(Ham|d5) \propto 0.0007$$

Numbers are too small.
May lead to underflow!

Naïve Bayes

Numerical Detail

$$P(y|\boldsymbol{x}) \propto P(y) \times \prod_{i=1}^{n} P(x_i|y)$$

numerical underflow when P is large

Solution: Take the log

$$\log P(y) + \sum_{i=1}^{n} \log P(x_i|y)$$

log preserves the order,

$$\log x \leq \log y \qquad for \ \mathrm{x} \leq y$$

therefore, no need to transform back.

Multinomial Naïve Bayes

The one with numbers

MultinomialNB implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification.

The parameters θ_y is estimated by a smoothed version of maximum likelihood,

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing.

Source: scikit-learn, Multinomial Naïve Bayes manual

Bernoulli Naïve Bayes

The one with ones

BernoulliNB implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions. (scikit-learn)

Input vector must be binary-valued.

The decision rule for Bernoulli naive Bayes is based on

$$P(x_i|y) = P(i|y)x_i + (1 - P(i|y))(1 - x_i)$$

BernoulliNB might perform better on datasets with shorter documents.

Source: scikit-learn, Bernoulli Naïve Bayes manual

Gaussian Naïve Bayes

The one with one point one

GaussianNB implements the Gaussian Naive Bayes algorithm for classification. (scikit-learn)

The likelihood of the features is assumed to be Gaussian.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The parameters σ_y and μ_y are estimated using maximum likelihood.

Source: scikit-learn, Gaussian Naïve Bayes manual

Naïve Bayes

Gaussian Naïve Bayes

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma^2_y}\right)$$

```
class _BaseNB(ClassifierMixin, BaseEstimator, metaclass=ABCMeta):
    """Abstract base class for naive Bayes estimators"""

    @abstractmethod
    def _joint_log_likelihood(self, X):
        """Compute the unnormalized posterior log probability of X

        I.e. ``log P(c) + log P(x|c)`` for all rows x of X, as an array-like of
        shape (n_classes, n_samples).

        Input is passed to _joint_log_likelihood as-is by predict,
        predict_proba and predict_log_proba.
        """
```

```
    def _joint_log_likelihood(self, X):
        joint_log_likelihood = []
        for i in range(np.size(self.classes_)):
            jointi = np.log(self.class_prior_[i])
            n_ij = - 0.5 * np.sum(np.log(2. * np.pi * self.var_[i, :]))
            n_ij -= 0.5 * np.sum(((X - self.theta_[i, :]) ** 2) /
                                 (self.var_[i, :]), 1)
            joint_log_likelihood.append(jointi + n_ij)

        joint_log_likelihood = np.array(joint_log_likelihood).T
        return joint_log_likelihood
```

Source: scikit-learn, https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/naive_bayes.py

Metrics

Confusion Matrix

ID	Message	Actual	Predicted
M-1	Married local hamsters looking for discreet action now! 5 real matches instantly to your phone. Text MATCH to 69969 Msg cost 150p 2 stop txt stop BCMSFWC1N3XX	Spam	Ham
M-2	Sorry, went to bed early, nightnight	Ham	Ham
M-3	Shall i start from hear.	Ham	Ham
M-4	Bring home some Wendy =D	Ham	Ham
M-5	URGENT! Your mobile was awarded a £1,500 Bonus Caller Prize	Spam	Spam
M-6	Customer Loyalty Offer:The NEW Nokia6650 Mobile from ONLY £10 at TXTAUCTION!	Spam	Spam
M-7	No. I.ll meet you in the library	Ham	Spam
M-8	Ok da, i already planned. I will pick you.	Ham	Spam
M-9	Yep then is fine 7.30 or 8.30 for ice age.	Ham	Ham
M-10	I'm hungry buy smth home...	Ham	Ham
M-11	Lol now I'm after that hot air balloon!	Ham	Ham

Metrics

Confusion Matrix

		Predicted	
		Ham	Spam
Actual	Ham	M-2, M-3, M-4, M-9, M-10, M-11	M-7, M-8
	Spam	M-1	M-5, M-6

$$F_1\,score = \frac{1}{Recall^{-1} + Precision^{-1}}$$

$$Accuracy = \frac{(6+2)}{(6+2+1+2)} = 0.72$$

$$Precision = \frac{2}{(2+2)} = 0.5$$

$$Recall = \frac{2}{(2+1)} = 0.66$$

Next Session