



SESSION # 8

By Team 2
@ Sojung, Jaehoon, Youngjun
Date _ 2017.12.15

CONTENTS

01 Probability

02 Naive Bayes

03 Practice with Sci-kit Learn

04 Quest

01 Probability

베이지 정리

조건부확률

확률이 0이 아닌 사건 A가 일어났을 때 사건 B가 일어날 확률을 사건 A가 일어났을 때의 사건 B의 조건부확률이라 하고

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad (\text{단, } P(A) \neq 0)$$

...와 같이 나타낸다.

O1 Probability

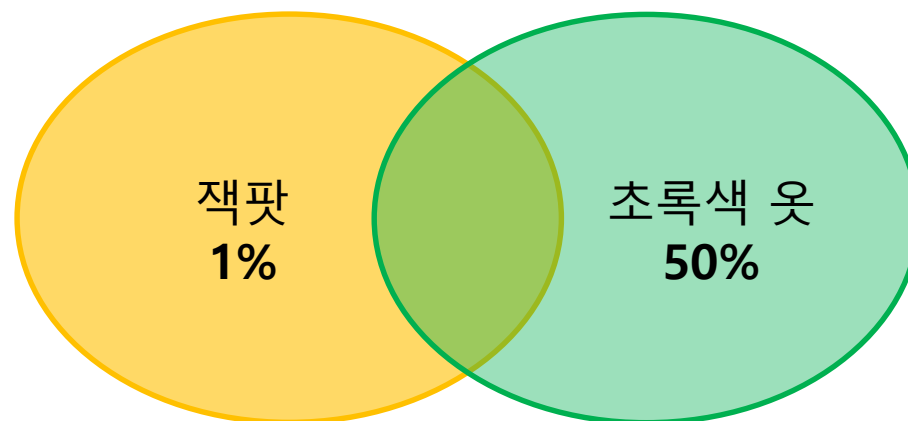
베이지 정리

각 사건이 독립일때

$$P(A \cap B) = P(A) * P(B)$$

$$P(A|B) = P(A)$$

$$P(B|A) = P(B)$$



예 시

카지노에서 잭팟을 맞을 확률은 1%, 초록색 옷을 입을 확률은 50%라고 할때, 초록색 옷을 입고 잭팟을 맞을 확률은?

옷의 색깔과 잭팟 맞을 확률은 관계가 없으므로 서로 독립이다.

$$P(\text{잭팟 맞을 확률} \cap \text{초록색 옷을 입을 확률}) = P(\text{잭팟 맞을 확률}) * P(\text{초록색 옷을 입을 확률})$$

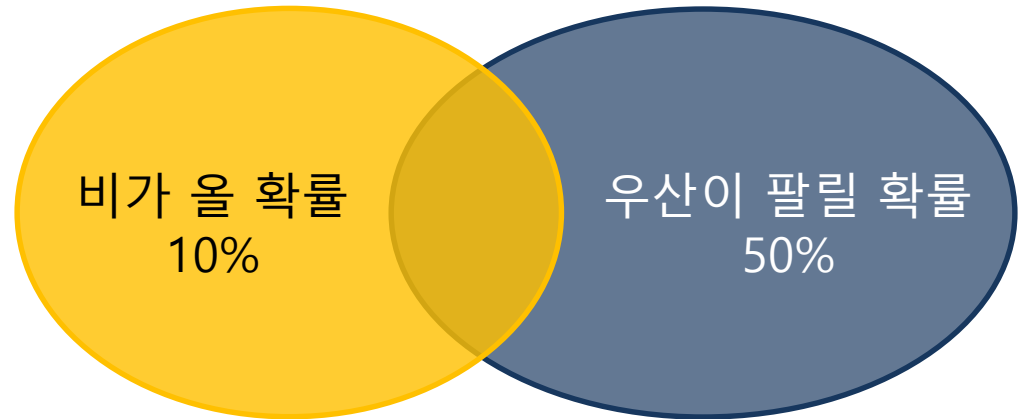
O1 Probability

베이지 정리

각 사건이 독립이 아닐 때

$$P(A \cap B) = P(A) * P(B|A)$$

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$



O1 Probability

베이지 정리

조건부 확률의 정의를 ' 반대로 뒤집는 ' 정리

사건 E가 발생했을 때 사건 F가 발생할 확률 $P(F|E)$ 를 이용해
사건 F가 발생했을 때 사건 E가 발생할 확률 $P(E|F)$ 구하기

$$P(E|F) = \frac{P(F \cap E)}{P(F)} = \frac{P(F \cap E)}{P(F \cap E) + P(F \cap \sim E)} = \frac{P(F|E)P(E)}{P(F|E)P(E) + P(F|\sim E)P(\sim E)}$$

Ex.

사건 E = 특정 메일이 스팸일 확률

사건 F = 메일에 F라는 단어가 포함될 확률

O1 Probability

베이지 정리

Q.

질병에 걸린 (D) 사람이 양성 판정을 받을 (T) 확률 $P(T|D)=0.99$
특정 사람이 질병에 걸릴 확률 $P(D)=0.0001$

이때, 양성 판정을 받은 사람이 질병에 걸린 사람일 확률은?

O1 Probability

베이지 정리

Q.

질병에 걸린 (D) 사람이 양성 판정을 받을 (T) 확률 $P(T|D)=0.99$

특정 사람이 질병에 걸릴 확률 $P(D)=0.0001$

$$P(T|\sim D)=1-P(T|D)=0.01$$

$$P(\sim D)=1-0.0001=0.9999$$

이때, 양성 판정을 받은 사람이 질병에 걸린 사람일 확률은?

$$=P(D|T)=\frac{P(T|D)P(D)}{P(T|D)*P(D)+P(T|\sim D)P(\sim D)} = 0.98 \%$$

O1 Probability

■ 나이브 베이즈

베이즈 정리에 ' 나이브하고 극단적인 가정 ' 하나를 추가한 것

-> "각 단어의 존재 혹은 부재는 서로 조건부 독립적이다."

스팸메일에 A라는 단어와 B라는 단어가 등장할 확률은 서로 독립적

= 스팸메일에 A과 B가 모두 등장할 확률은 두 단어가 각각 등장할 확률의 곱

= 각 단어들의 결합 확률은 개별 확률들의 곱

학습 데이터 (Training Set) + 나이브 베이즈 -> 분류기

O1 Probability

Smoothing

스팸인 메일 중에서
스팸이었던 적이 없는 단어가 있다!

사건 E=특정 메일이 스팸일 확률
사건 F=메일에 F라는 단어가 포함될 확률

만약 학습 데이터에서 한번도 등장하지 않은 단어가 메시지에 등장한다면?

$$P(E|F) = \frac{P(F|E)P(E)}{P(F|E)P(E) + P(F|\sim E)P(\sim E)} = 0$$

이 추정치가 곱해질 때 다른 확률의 모든 정보를 없앨 수 있음

- 이 오류를 처리하기 위해 smoothing(정규화) 기법 도입

Ex. $P(F|E) = \frac{k + (\text{단어 } F \text{가 등장한 스팸메일 수})}{2k + (\text{총 스팸 수})}$ (pseudocount k 도입)

- 1) 라플라스 정규화 (pseudocount=1)
- 2) 리드스톤 정규화

O1 Probability

- 학습 데이터

No.	Words	Class
1	fun, couple, love, love	comedy
2	fast, furious, shoot	action
3	couple, fly, fast, fun, fun	comedy
4	furious, shoot, shoot, fun	action
5	fly, fast, shoot, love	action

- 입력 데이터 : {fast, furious, fun} = 사건 D
- 데이터의 Class가 Comedy일 사건=C , Action일 사건=A

$$P(C|D) = \frac{P(D \cap C)}{P(D)} = \frac{P(D \cap C)}{P(D \cap C) + P(D \cap A)} = \frac{P(D|C)P(C)}{P(D|C)P(C) + P(D|A)P(A)}$$

$$P(A|D) = \frac{P(D|A)P(A)}{P(D|C)P(C) + P(D|A)P(A)}$$

O1 Probability

- 학습 데이터

No.	Words	Class
1	fun, couple, love, love	comedy
2	fast, furious, shoot	action
3	couple, fly, fast, fun, fun	comedy
4	furious, shoot, shoot, fun	action
5	fly, fast, shoot, love	action

	Fun	Couple	Love	Fast	Furious	Shoot	Fly	(SUM)
Comedy	3	2	2	1	0	0	1	9
Action	1	0	1	2	2	4	1	11

- 입력 데이터 : {fast, furious, fun} = 사건 D

- $P(D|C)P(C)$

$$= P(\text{fast}|C) * P(\text{furious}|C) * P(\text{fun}|C) * P(C)$$

$$= \frac{1}{9} * \frac{0}{9} * \frac{3}{9} * \frac{9}{20} \quad (\text{smoothing 전}) \quad \text{vs.} \quad = \frac{1+1}{9+2} * \frac{0+1}{9+2} * \frac{3+1}{9+2} * \frac{9+1}{20+2} \quad (\text{laplace smoothing 후})$$

O1 Probability

- 학습 데이터

No.	Words	Class
1	fun, couple, love, love	comedy
2	fast, furious, shoot	action
3	couple, fly, fast, fun, fun	comedy
4	furious, shoot, shoot, fun	action
5	fly, fast, shoot, love	action

	Fun	Couple	Love	Fast	Furious	Shoot	Fly	(SUM)
Comedy	3	2	2	1	0	0	1	9
Action	1	0	1	2	2	4	1	11

- 입력 데이터 : {fast, furious, fun} = 사건 D
- Q. laplace smoothing을 적용한 나이브베이즈 분류기에서
입력 데이터는 어떤 Class로 분류될까요?

O1 Probability

■ 나이브베이즈 분류 이벤트 모델

1. 베르누이 분포 (이항분포)

X 는 0 또는 1, 각 확률은 고정 [Ex- 스팸메일, 성별 분류기]

2. 다항분포

(x_1, \dots, x_n) 이 0 또는 양의 정수

[Ex- 주사위를 던진 결과로 어느 주사위를 던졌는지를 찾아내는 모형]

3. 가우시안 정규분포

X 는 실수, 특정한 값 근처

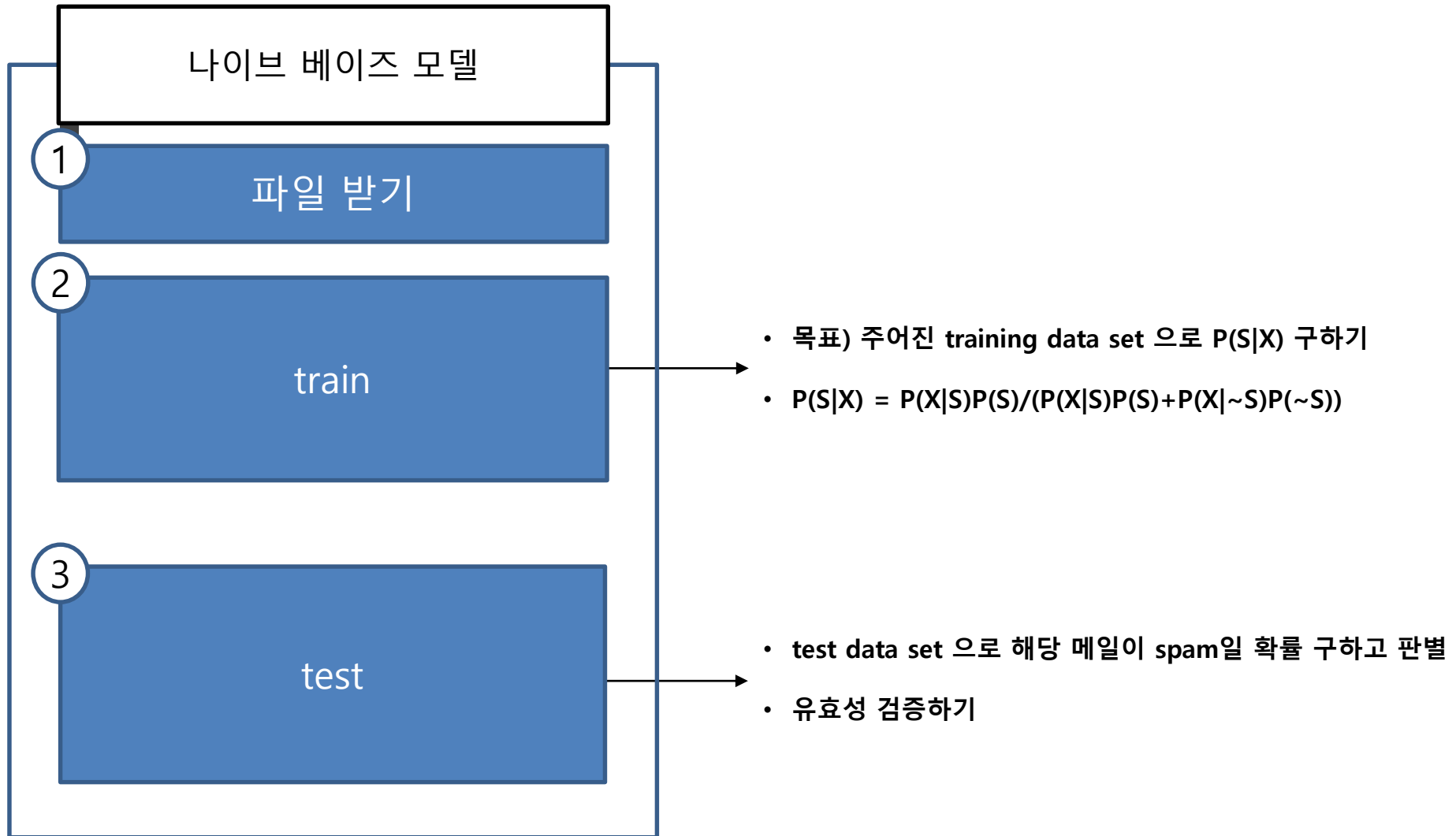
[Ex- 시험 점수로 학생을 찾아내는 모형]

03 Naive Bayes

책 코드 살펴보기

- 메일 제목에 등장하는 단어들로 Spam 여부를 판단하는 나이브 베이즈 분류기를 만들어 본다.
- 각 특성(단어)는 등장하거나, 등장하지 않거나 두가지 경우 이므로 베르누이 분포 모형을 따른다.

02 Naive Bayes



02 Naive Bayes

Import 해야하는 모듈

```
import glob  
import re  
import math  
import random
```

[****]
[정규표현식]
[수학 관련]
[임의의 수 생성]

```
From collections import Counter  
From collections import defaultdict
```

[개수를 세준다]
[딕셔너리 관련]

02 Naive Bayes

Train - 1. 데이터 정제하기

1. 메일 제목을 단어 단위로 나누기 → 정규표현식을 사용한다.
2. 소문자로 변환하기 → lower()
3. 중복되는 단어 제거하기 → set()

```
def tokenize(message):  
    message = message.lower()  
    all_words = re.findall("[a-z0-9]+", message)  
    return set(all_words)
```

>>> 실행 결과

```
print(tokenize("Hi my name is Aring Aring, Nice to meet you"))  
{'nice', 'name', 'hi', 'my', 'to', 'you', 'aring', 'is', 'meet'}
```

02 Naive Bayes

Train - 2. $P(X|S)$, $P(X|\sim S)$ 구하기

1. 단어가 나왔을 때 스팸 메일이었던 횟수, 스팸이 아니었던 횟수 구하기

* training_set 을 입력인수로 받음

* training_set은 (메시지 내용, 스팸 여부 형식으로 구성)

```
(': Re: [Webdev] mod_usertrack', False)
```

* defaultdict(): 딕셔너리의 형식 지정

```
def count_words(training_set):
    counts = defaultdict(lambda: [0,0])
    for message, is_spam in training_set:
        for word in tokenize(message):
            counts[word][0 if is_spam else 1] += 1
    return counts
```

>>> 실행 결과

```
'reports': [0, 5], 'moon': [0, 4], 'pravda': [0, 2]
```

02 Naive Bayes

Train - 2. $P(X|S)$, $P(X|\sim S)$ 구하기

2. [단어, $P(X|S)$, $P(X|\sim S)$]를 리턴하는 함수 만들기

- * smoothing을 위한 가짜 빈도수 k 고려하기
- * dict.items(): 딕셔너리의 (key, value) 쌍을 복사한 list를 리턴

```
dict_items([('bomber', [0, 1]), ('zzzzteana', [0, 50]), ('moscow', [0, 1]),
```

- * count_words의 결과값 / 총 스팸 수 / 총 ~스팸 수 / k 를 입력인수로

```
def word_probabilities(counts, total_spams, total_non_spams, k=0.5):
    return [(w,
              (spam + k) / (total_spams + 2*k),
              (non_spam + k) / (total_non_spams + 2*k))
            for w, (spam, non_spam) in counts.items()]
```

>>> 실행 결과

```
[('moscow', 0.0015337423312883436, 0.0007194244604316547),
```

```
('zzzzteana', 0.0015337423312883436, 0.02422062350119904)]
```

02 Naive Bayes

Train - 3. $P(S|X)$ 구하기

- * $\text{prob_if_spam} = P(X|S)$ * $\text{prob_if_not_spam} = P(X|\sim S)$
- * [단어, $P(X|S)$, $P(X|\sim S)$] / '판별하고자 하는 이메일의 제목'을 입력인수로 받음
- * 부동소수점 문제를 피하기 위해 \log , \exp 를 사용
- * 단어들이 조건부 독립이라는 가정 하에 $P(X|S) = \text{mul}(P(X_{w1,2,...,n}|S))$

Ex) 단어 A가 등장, 단어B가 등장

$$\rightarrow P(X|S) = \text{mul}(P(X_{wa}|S), P(X_{wb}|S))$$

단어 A는 등장하지 않고, 단어 B만 등장했다면

$$\rightarrow P(X|S) = \text{mul}(P(\sim X_{wa}|S), P(X_{wb}|S))$$

- * $P(S|X)$ 를 리턴

02 Naive Bayes

Train - 3. $P(S|X)$ 구하기

```
def spam_probability(word_probs, message):  
    message_words = tokenize(message)  
    log_prob_if_spam = log_prob_if_not_spam = 0.0  
  
    for word, prob_if_spam, prob_if_not_spam in word_probs:  
        if word in message_words:  
            log_prob_if_spam += math.log(prob_if_spam)  
            log_prob_if_not_spam += math.log(prob_if_not_spam)  
        else:  
            log_prob_if_spam += math.log(1.0 - prob_if_spam)  
            log_prob_if_not_spam += math.log(1.0 - prob_if_not_spam)  
  
    prob_if_spam = math.exp(log_prob_if_spam)  
    prob_if_not_spam = math.exp(log_prob_if_not_spam)  
    return prob_if_spam / (prob_if_spam + prob_if_not_spam)
```

02 Naive Bayes

Train – 4. 종합

* 나이브 베이즈 클래스 만들기

이를 통해서 Training set 에서 스팸 메시지와 스팸이 아닌 메시지의 개수 구한다.

```
class NaiveBayesClassifier:
    def __init__(self, k=0.5):
        self.k = k
```

>>> NaiveBayesClassifier()를 적용한 결과

```
'reports': [0, 5], 'moon': [0, 4], 'pravda': [0, 2]
```

```
[('moscow', 0.0015337423312883436, 0.0007194244604316547),
 ('zzzzteana', 0.0015337423312883436, 0.02422062350119904)]
```

```
return spam_probability(self.word_probs, message)
```

02 Naive Bayes

Test

* 데이터를 training set과 test set으로 나눈다.

```
def split_data(data, prob):  
    results = [], []  
    for row in data:  
        results[0 if random.random() < prob else 1].append(row)  
    return results
```

random.random(): 0-1사이의 임의의 수를 리턴

따라서 변수 prob의 확률 만큼의 data가 train data에 속하게 된다.

>>> **split_data()**를 적용한 결과

results[0] = training set

results[1] = test set

02 Naive Bayes

Test

glob.glob()

>>> 해당 경로의 모든 파일 목록을 불러옴

불러오지 못하는 이메일은 예외처리를 한다.

>>> 예외 처리 뒤 반복문 계속 진행

re.sub("해당문자", "대체문자", "적용할 객체")

정규 표현식의 '^' 은 문자열의 시작을 의미함

```
path = r"C:\Users\LYJ\Desktop\Spam\*.*"
data = []

for fn in glob.glob(path):
    is_spam = "ham" not in fn

    with open(fn, 'r') as file:
        try:
            lines = file.readlines()
        except UnicodeDecodeError as e: continue
        else:
            for line in lines:
                if line.startswith("Subject"):
                    subject = re.sub(r"^Subject", "", line).strip()
                    data.append((subject, is_spam))

random.seed(0)
train_data, test_data = split_data(data, 0.75)
classifier = NaiveBayesClassifier()
classifier.train(train_data)
```

'ham'의 포함 여부로 is_spam에 T/F 지정

^Subject

02 Naive Bayes

모델 평가

* Counter(): 각 데이터가 등장한 횟수를 딕셔너리 형식으로 리턴

```
classified = [(subject, is_spam, classifier.classify(subject))
               for subject, is_spam in test_data]

counts = Counter((is_spam, spam_probability > 0.5)
                 for _, is_spam, spam_probability in classified)
```

```
classified: (': EnenKio truth (Answers)', True, 0.7054599657954321)
```

>>> Counter 실행 결과

```
Counter([(False, False): 670, (True, True): 65, (True, False): 44, (False, True): 23])
```

→ accuracy, precision, recall 등을 계산할 수 있음

02 Naive Bayes

분류 결과 분석

classified: (': EnenKio truth (Answers)', True, 0.7054599657954321)

```
# 스팸일 확률을 오름차순으로 정렬
classified.sort(key=lambda row: row[2])

# 스팸이 아닌 메시지 중에서 스팸일 확률이 가장 높은 메시지
spammiest_hams = list(filter(lambda row: not row[1], classified))[-5:]

# 스팸 중에서 스팸일 확률이 가장 낮은 메시지
hammiest_spams = list(filter(lambda row: row[1], classified))[:5]
```

02 Naive Bayes

분류 결과 분석

스팸일 확률이 가장 높은 단어, 스팸이 아닐 확률이 가장 높은 단어 추출
 나이브 베이즈 클래스 `_init_`에 있는 `self.word_probs` 사용

```
def p_spam_given_word(word_prob):
    word, prob_if_spam, prob_if_not_spam = word_prob
    return prob_if_spam / (prob_if_spam + prob_if_not_spam)

words = sorted(classifier.word_probs, key=p_spam_given_word)

spammiest_words = words[-5:]
hammiest_words = words[:5]
```

>>> 실행결과

```
[('clearance', 0.0352760736196319, 0.00023980815347721823), ('rates', 0.04141104294478527, 0.00023980815347721823),
[('spambayes', 0.0015337423312883436, 0.0486810551558753), ('users', 0.0015337423312883436, 0.03860911270983213), (
```

02 Naive Bayes

모델 개선 방법

1. 이메일의 내용에도 나이브 베이즈 모델 적용
2. 단어의 최소 빈도 수 정해 기준보다 적게 나온 단어 무시
3. stemmer를 통해 동의어를 묶어주기
4. 다른 변수 사용하기

03 Practice with sci-kit learn

Kaggle

2010년 설립된 예측모델 및 분석 대회 플랫폼이다. 기업 및 단체에서 데이터와 해결과제를 등록하면, 데이터 과학자들이 이를 해결하는 모델을 개발하고 경쟁한다. (위키피디아)



Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics



Kaggle · 9,797 teams · 2 years to go

타이타닉호의 생존자를 나이브베이즈로 예측해보자!

(<https://www.kaggle.com/c/titanic/data>)

03 Practice with sci-kit learn

Scikit-learn

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning

Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.

Modules: preprocessing, feature extraction. — Examples

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

① Importing Libraries

```
import numpy as np
import pandas as pd
```

```
from scipy.stats import itemfreq
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
```

① Data Import

```
train = pd.read_csv("train.csv", dtype={"Age" : np.float64})
test = pd.read_csv("test.csv", dtype={"Age" : np.float64})
```


03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

>>> Train.head(10)

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

③ One-Hot Encoding

- 범주형 변수의 이항변수화
- 0과 1로만 구성된 가변수(Dummy Variable) 로의 재구성
- 파이썬에서는 자동으로 변수별 범주의 종류, 개수를 파악

$$Y = \begin{pmatrix} 0 \\ 1 \\ 4 \\ 9 \end{pmatrix} \xrightarrow{\text{One hot encoding}} Y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

③ One-Hot Encoding

둘 중 하나의 방법을 사용하면 된다.

* **pandas.factorize()**

```
train['Sex'] = pd.factorize(train['Sex'])[0]
test['Sex'] = pd.factorize(test['Sex'])[0]
```

사용한 예시에서는 이 코드를 사용하였지만 좋은 방법은 아니다.
왜냐하면 `pd.factorize`는 Index 값을 가져오는 것이기 때문이다.
리스트로 반환하기 때문에 마지막에 [0]으로 슬라이싱을 해주어야 한다.

* **pandas.get_dummies()**

```
df_gender = pd.get_dummies(train['Sex'])
```

`get_dummies()`는 앞서 보았던 것과 같은 형태로 반환한다.

`df_gender.head()`

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

③ One-Hot Encoding

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	0	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	0	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	0	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	1	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	1	14.0	1	0	237736	30.0708	NaN	C

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

④ Handling Missing Data

77개의 결측치가 있다.

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	0.352413	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	0.477990	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	0.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	0.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	1.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	1.000000	80.000000	8.000000	6.000000	512.329200

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

⑤ Data Preprocessing

둘 중 하나의 방법을 사용하면 된다.

* pandas.DataFrame.**fillna()**

```
train.fillna(train.mean(), inplace = True)
test.fillna(test.mean(), inplace = True)
```

Value to use to fill missing data.

결측치가 NA, NaN으로 정의되어있다.

* sklearn.preprocessing.**Imputer()**

```
imp = Imputer(missing_values="NaN", strategy="mean")
train["Age"] = imp.fit_transform(train[["Age"]]).ravel()
```

결측치가 정수 형태 또는 NaN

적용 대상이 될 데이터
(numpy array of shape)

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

⑥ Data Slicing

```
trainData = pd.DataFrame.as_matrix(train[['Pclass', 'Sex', 'Age']])
```

```
trainTarget = pd.DataFrame.as_matrix(train[['Survived']]).ravel()
```

```
testData = pd.DataFrame.as_matrix(test[['Pclass', 'Sex', 'Age']])
```

trainData

```
array([[ 3.,      0.,      3.09104245],
       [ 1.,      1.,      3.63758616],
       [ 3.,      1.,      3.25809654],
       ...,
       [ 3.,      1.,      3.39111734],
       [ 1.,      0.,      3.25809654],
       [ 3.,      0.,      3.4657359 ]])
```

testData

```
array([[ 3.,      0.,      3.54095932],
       [ 3.,      1.,      3.8501476 ],
       [ 2.,      0.,      4.12713439],
       ...,
       [ 3.,      0.,      3.65065824],
       [ 3.,      0.,      3.41024269],
       [ 3.,      0.,      3.41024269]])
```

03 Practice with sci-kit learn

Gaussian Naïve Bayes Classifier

⑦ Gaussian Naive Bayes Implementation

```
classifier = GaussianNB()

classifier.fit(trainData, trainTarget)

GaussianNB(priors=None)

predictedValues = classifier.predict(testData).astype(int)

itemfreq(predictedValues)

array([[ 0, 260],
       [ 1, 158]], dtype=int64)

testResults = test[['PassengerId']]
testResults['Survived'] = predictedValues
```


03 Practice with sci-kit learn

Kaggle

kaggle

Search kaggle



Competitions

Datasets

Kernels

Discussion

Jobs

...



Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics



Kaggle · 9,797 teams · 2 years to go

[Overview](#)[Data](#)[Kernels](#)[Discussion](#)[Leaderboard](#)[Rules](#)[Team](#)[My Submissions](#)[Submit Predictions](#)

Overview

[Description](#)[Evaluation](#)[Frequently Asked Questions](#)[Tutorials](#)

Start here if...

You're new to data science and machine learning, or looking for a simple intro to the Kaggle prediction competitions.

Competition Description


The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this challenge, we ask you to complete the analysis of what sorts of people were likely to survive. In particular, we ask you to apply the tools of machine learning to predict which passengers survived the tragedy.

03 Practice with sci-kit learn

Kaggle


 Kaggle · 9,797 teams · 2 years to go

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [My Submissions](#) [Submit Predictions](#)

Make a submission for [Asphalt](#)

You have 10 submissions remaining today. This resets 12 hours from now (00: 00 UTC).










Step 1
Upload submission file


Upload Submission File
생성한 csv파일은 여기에 드랍!

File Format
Your submission should be in CSV format.
You can upload this in a zip/gz/rar/7z archive, if you prefer.

Number of Predictions
We expect the solution file to have 418 prediction rows. This file should have a header row. Please see sample submission file on the [data page](#).

Step 2
Describe submission

B **/** |     |   **H** |  |  



















Styling with Markdown supported

Briefly describe your submission.

[Make Submission](#)

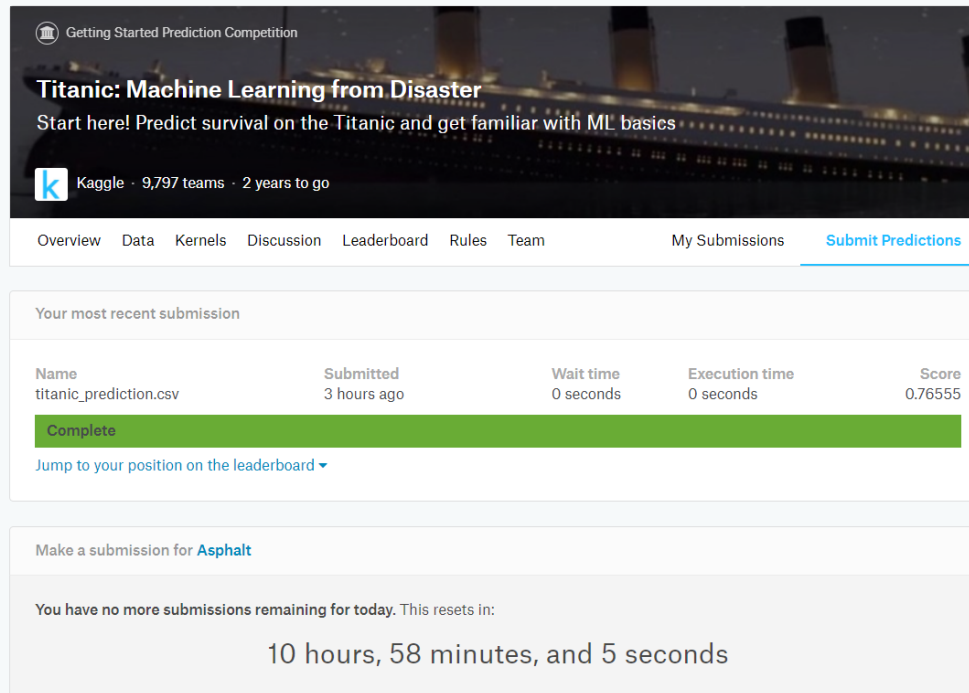
03 Practice with sci-kit learn

Kaggle

Overview	Data	Kernels	Discussion	Leaderboard	Rules	Team	My Submissions	Submit Predictions		
6248	new	swleev						0.77033	4	1d
6249	new	yashagrawal3						0.77033	3	21h
6250	new	ttgreen99						0.77033	2	20h
6251	▲ 348	ElChiang						0.77033	2	19h
6252	new	Miguel Arce						0.77033	1	18h
6253	new	Reginaldo						0.77033	1	17h
6254	new	Films						0.77033	5	3h
6255	new	Markus Schaber						0.77033	2	10h
6256	new	Asphalt						0.77033	10	~10s
Your Best Entry ↑ Your submission scored 0.76555, which is not an improvement of your best score. Keep trying!										
6257	▼ 512	Mike Letts						0.77033	12	2h
6258	new	Katie Wu						0.77033	1	2h
6259	new	Eshwar						0.77033	1	1h
6260	new	Robert Lowry						0.77033	1	30m
📍		Gender Based Model						0.76555		
6261	▼ 667	yakan10						0.76555	1	2mo
6262	▼ 667	kumar4372						0.76555	2	2mo
6263	▼ 667	Maple 3						0.76555	6	2mo
6264	▼ 667	NathanWelch						0.76555	4	2mo

03 Practice with sci-kit learn

Kaggle



The screenshot shows the Kaggle interface for the 'Titanic: Machine Learning from Disaster' competition. At the top, there's a header with the competition title and a brief description: 'Start here! Predict survival on the Titanic and get familiar with ML basics'. Below this, it says 'Kaggle · 9,797 teams · 2 years to go'. A navigation bar includes links for Overview, Data, Kernels, Discussion, Leaderboard, Rules, Team, My Submissions, and Submit Predictions. The main content area shows 'Your most recent submission' with a table of submission details. The table has columns for Name, Submitted, Wait time, Execution time, and Score. The submission 'titanic_prediction.csv' is listed as 'Submitted 3 hours ago', 'Wait time 0 seconds', 'Execution time 0 seconds', and 'Score 0.76555'. A green bar indicates the submission is 'Complete'. Below the table, there's a link to 'Jump to your position on the leaderboard'. At the bottom, there's a section for making a submission for 'Asphalt', which is currently disabled, showing a message: 'You have no more submissions remaining for today. This resets in: 10 hours, 58 minutes, and 5 seconds'.

Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

Kaggle · 9,797 teams · 2 years to go

Overview Data Kernels Discussion Leaderboard Rules Team My Submissions [Submit Predictions](#)

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
titanic_prediction.csv	3 hours ago	0 seconds	0 seconds	0.76555

Complete

[Jump to your position on the leaderboard](#)

Make a submission for **Asphalt**

You have no more submissions remaining for today. This resets in:

10 hours, 58 minutes, and 5 seconds

04 Quest

Quest

특정 단어가 메일에 있을 때 그 메일이 스팸메일 일 확률!

```
from numpy import *

def loadDataSet():
    postingList=[['I', 'got', 'free', 'two', 'movie', 'ticket', 'from', 'your', 'boy', 'friend'],
                  ['free', 'coupon', 'from', 'xx.com'],
                  ['watch', 'free', 'new', 'movie', 'from', 'freemovie.com'],
                  ['best', 'deal', 'promo', 'code', 'here'],
                  ['there', 'will', 'be', 'free', 'pizza', 'during', 'the', 'meeting'],
                  ['scheduled', 'meeting', 'tomorrow'],
                  ['can', 'we', 'have', 'lunch', 'today'],
                  ['I', 'miss', 'you'],
                  ['thanks', 'my', 'friend'],
                  ['it', 'was', 'good', 'to', 'see', 'you', 'today'],
                  ['free', 'coupon', 'last', 'deal'],
                  ['free', 'massage', 'coupon'],
                  ['I', 'sent', 'the', 'coupon', 'you', 'asked', 'it', 'is', 'not', 'free'],
                  ['coupon', 'promo', 'code', 'here']]
```

04 Quest

Quest

특정 단어가 메일에 있을 때 그 메일이 스팸메일 일 확률!

코드가 복잡하니 다음 코드와 해설 동영상을 참고해주세요

<https://www.youtube.com/watch?v=d3IEY-hyhag>

<https://github.com/minsuk-heo/machinelearning/blob/master/machineLearningInAction/03.naivebayes/bayes.py>

[마신러닝] 나이브 베이즈(Naive Bayes) 분류 (2/2) - 파이썬으로 구현하기

```

setOfWords2Vec

def setOfWords2Vec(vocabList, inputSet):
    returnVec = [0]*len(vocabList)
    for word in inputSet:
        if word in vocabList:
            returnVec[vocabList.index(word)] = 1
        else: print "the word: %s is not in my Vocabulary!" % word
    return returnVec

```

index	Email
14	Coupon, promo code here!

```

set([code, 'deal', 'coupon', 'is', 'it', 'massage', 'see', 'today', 'thanks', 'have',
miss, 'your', 'best', 'friend', 'from', 'movie', 'there', 'two', 'tomorrow',
to, 'new', 'you', 'meeting', 'sent', 'pizza', 'scheduled', 'be', 'we', 'good',
'promo', 'watch', 'free', 'lunch', 'freemovie.com', 'here', 'got', 'not', 'during',
ticket', 'boy', 'I', 'last', 'xx.com', 'was', 'will',
'can', 'the', 'my', 'asked', 'today'])

```

1:17 / 5:33

Source is from <https://www.manning.com/books/machine-learning-in-action>, author: Peter Harrington

05 References

- <http://scikit-learn.org/stable/>
- <https://datascienceschool.net/view-notebook/293ece8b0d124fbaa4d4d52bb8f1cb42/>
- <http://nbviewer.jupyter.org/github/metamath1/ml-simple-works/blob/master/naive/naive.ipynb>

THANK YOU !