**Programming Assignment02**

|  |
| --- |
| **Submission guide**  1. Write answer following questions in this file  2. Write your code using provided Jupyter notebook file   * Do not use other packages that are not already imported in the script * After completing your code, run script and submit with the printed results for answering questions in this word file |

**1. Naïve Bayes (35pts)**

The goal of this problem is to build naive Bayes (NB) models to classify the sentiment of tweets for US airline companies.

The following code loads the dataset.

|  |
| --- |
| data=pd.read\_csv('https://drive.google.com/uc?export=download&id=12i9GohkICVTq9OHv4W5l89N3J94b77wf') |

The data used in this problem provide frequencies of words for individual tweets in addition to 'tweet\_id', 'airline', 'airline\_sentiment.'

'airline\_sentiment' is the target variable, and the sentiment of tweets is one of 'negative', 'neutral', 'positive.'

The column names except for 'tweet\_id', 'airline', and 'airline\_sentiment' variables denote terms used in tweets.

**Part 1: Bernoulli NB**

To build a Bernoulli NB, if a certain term is used in a tweet, the values of the variables are converted to 1; otherwise, they are converted to 0.

1-(1) After the conversion, train a Bernoulli NB using training set (the converted trnX, trnY) (alpha=1). The prior probabilities of the classes are proportional to the ratios of the classes observed in the training set. Then, calculate the overall accuracy and accuracy values corresponding to each target class (sentiment) for the training and validation sets, respectively. (5pts)

train accuracy: 0.839 / train negative class: 0.85 train neutral class: 0.89 train positive class: 0.93

valid accuracy: 0.7968 / valid negative class: 0.81 valid neutral class: 0.87 valid positive class: 0.91

1-(2) Find the top 10 most probable terms in each target according to the model trained in Question 1-(1) and summarize them with the probability of existence of terms in the following tables. (3pts)

[Negative]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | Reason | 0.472 |
| 2 | Sit | 0.271 |
| 3 | Bag | 0.257 |
| 4 | Family | 0.208 |
| 5 | Going | 0.184 |
| 6 | Agents | 0.179 |
| 7 | Fail | 0.177 |
| 8 | Member | 0.175 |
| 9 | Supposed | 0.169 |
| 10 | Yes | 0.168 |

[Neutral]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | Reason | 0.436 |
| 2 | soon | 0.272 |
| 3 | Back | 0.267 |
| 4 | Bag | 0.257 |
| 5 | Missing | 0.222 |
| 6 | Going | 0.217 |
| 7 | Lga | 0.198 |
| 8 | Pick | 0.182 |
| 9 | Yes | 0.179 |
| 10 | Sit | 0.175 |

[Positive]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | Ridiculous | 0.432 |
| 2 | Reason | 0.375 |
| 3 | Oh | 0.288 |
| 4 | Baggage | 0.268 |
| 5 | Soon | 0.254 |
| 6 | Back | 0.247 |
| 7 | Bag | 0.243 |
| 8 | Family | 0.235 |
| 9 | Fail | 0.188 |
| 10 | Forhours | 0.180 |

1-(3) Describe your opinion related to the results of Question 1-(2). (5pts)

Bernoulli Naive Bayes 모델에서 각 단어의 확률 값은 해당 단어가 각 클래스(긍정/중립/부정)에 속할 확률을 나타냅니다. 이 확률 값은 베르누이 분포에서 파생된 것으로, 각 클래스의 단어 빈도수를 이진 값으로 변환하여 계산됩니다. 이진값으로 변환된 단어 빈도수는 해당 단어가 tweet에 등장했는지 여부만을 나타내며, 이를 이용해 각 클래스에서 해당 단어가 등장할 확률이 계산된 결과 입니다.

1-(4) Find the top 10 terms whose probability of existence is high in positive (pos) tweets, but low in negative (neg) tweets according to the model trained in Question 1-(1). In addition, find the top 10 terms whose probability of existence is high in negative tweets, but low in positive tweets. (3pts)

|  |  |  |
| --- | --- | --- |
| Order | High in pos, low in neg | High in neg, low in pos |
| 1 | Online | Rebooked |
| 2 | Bos | Rt |
| 3 | Happened | Iad |
| 4 | Expect | Point |
| 5 | Let | Sorry |
| 6 | Received | Issues |
| 7 | Points | Working |
| 8 | Im | Telling |
| 9 | Traveling | Luggage |
| 10 | Leaving | Dm |

1-(5) Describe your opinion related to the results of Question 1-(4). (5pts)

긍정에서는 빈도가 많은 단어들은 해당 tweet가 긍정적인 특징을 가진 경우가 높을 가능성이 있다는 것을 의미합니다. 단어들이 긍정 클래스에서 빈도가 많을 경우, 이 tweet가 긍정적인 내용을 담고 있을 가능성이 높습니다.

긍정에서는 빈도가 적고 부정에서는 빈도가 많은 단어들은 해당 tweet가 긍정 클래스에 속하지 않을 가능성이 높다는 정보를 제공합니다. 위에서 부정에서 많이 확인된 단어들은 부정적인 의미를 가지고 있으므로, 이러한 단어들이 긍정 클래스에서 빈도가 적을 경우, 해당 tweet가 긍정적인 내용을 담고 있지 않을 가능성이 높습니다.

**Part 2: Multinomial NB**

2-(1) Train a multinomial NB using training set (trnX, trnY) (alpha=1). The prior probabilities of the classes are proportional to the ratios of the classes observed in in training set. Then calculate the overall accuracy and accuracy values corresponding to each target class (sentiment) for the training and validation sets, respectively. (3pts)

train accuracy: 0.84

train negative class: 0.86

train neutral class: 0.89

train positive class: 0.93

valid accuracy: 0.7944

valid negative class: 0.81

valid neutral class: 0.87

valid positive class: 0.91

2-(2) Find the top 20 most probable terms in each target according to the model trained in Question 2-(1) and summarize them with the probability of existence of terms in the following tables. (3pts)

[Negative]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | Reson | 0.128 |
| 2 | Sit | 0.067 |
| 3 | Bag | 0.064 |
| 4 | Family | 0.051 |
| 5 | Going | 0.045 |
| 6 | Agents | 0.045 |
| 7 | Fail | 0.043 |
| 8 | Member | 0.043 |
| 9 | Yes | 0.043 |
| 10 | Supposed | 0.041 |

[Neutral]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | Reason | 0.114 |
| 2 | soon | 0.071 |
| 3 | Back | 0.068 |
| 4 | Bag | 0.064 |
| 5 | Missing | 0.054 |
| 6 | Going | 0.052 |
| 7 | Lga | 0.049 |
| 8 | Yes | 0.046 |
| 9 | Pick | 0.043 |
| 10 | forhours | 0.042 |

[Positive]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | Ridiculous | 0.102 |
| 2 | Reason | 0.094 |
| 3 | Oh | 0.070 |
| 4 | Baggage | 0.066 |
| 5 | Soon | 0.060 |
| 6 | Back | 0.058 |
| 7 | Bag | 0.058 |
| 8 | Family | 0.056 |
| 9 | Fail | 0.044 |
| 10 | Agents | 0.043 |

2-(3) Find the top 10 terms whose probability of existence is high in positive (pos) tweets, but low in negative (neg) tweets according to the model trained in Question 2-(1). In addition, find the top 10 terms whose probability of existence is high in negative tweets, but low in positive tweets. (3pts)

|  |  |  |
| --- | --- | --- |
| Order | High in pos, low in neg | High in neg, low in pos |
| 1 | Online | Rebooked |
| 2 | Bos | Rt |
| 3 | Happened | Iad |
| 4 | Expect | Point |
| 5 | Let | Sorry |
| 6 | Received | Issues |
| 7 | Im | Working |
| 8 | Leaving | Telling |
| 9 | Job | Luggage |
| 10 | Points | Dm |

2-(4) Compare the two NB models trained in Questions 1-(1) and 2-(1), considering the results of Questions 1-(1), (2), and (4) and 2-(1), (2), and (3). (5pts)

먼저 Bernoulli NB(이하 BNB)의 단어 별 확률이 계산 된 1-(2),(4) 과 Multinomial NB(이하 MNB)의 단어 별 확률이 계산 된 2-(2),(3)의 결과를 토대로 분석하자면, 각 모델이 특정 클래스를 추론하는데 주요하게 확인한 단어의 구성은 유사하지만, 몇몇의 차이가 있고, 확률 수치 또한 차이가 있습니다.

해당 결과로 알 수 있는 두 모델의 주요 차이점은 각 단어의 확률을 어떻게 계산하는가에 있습니다. MNB에서는 각 단어의 빈도수를 이용하여 확률을 계산하고, BNB에서는 각 단어가 tweet에 있는지 없는지의 이진 특징으로 확률을 계산합니다.

(1-(2), 2-(2) 에서 확인 된)

이러한 특징 확률의 차이는 두 모델이 서로 다른 결과를 낼 수 있으며, 이는 MNB와 BNB에서 동일한 tweet에 대해 서로 다른 확률을 제공할 수 있습니다. 예를 들어, 어떤 단어가 긍정적인 tweet에서 매우 빈번하게 나타나고, 부정적인 tweet에서는 나타나지 않는 경우, MNB 모델은 BNB 모델보다 긍정적인 tweet에서 이 단어가 존재할 확률을 더 높게 평가할 수 있습니다.(1-(4), 2-(3)에서 확인된)

또 다른 차이점은 zero-frequency problem라고 불리는 0 빈도 문제를 어떻게 처리하는가 입니다. MNB에서는 0 빈도 단어는 존재할 확률이 0이 되므로, 특정 클래스에서만 나타나는 단어의 경우 문제가 발생할 수 있습니다. BNB에서는 0 빈도 문제를 처리하기 위해 작은 상수를 특징 확률에 추가하여, 모든 단어가 양쪽 클래스에서 존재할 확률이 0이 되지 않도록 합니다.

요약하자면, 특징 확률과 0 빈도 문제를 다루는 방법의 차이가 MNB와 BNB에서 동일한 tweet에 대해 서로 다른 확률을 제공할 수 있습니다. 이러한 모델 선택은 데이터의 특성과 해결하려는 구체적인 문제에 따라 다릅니다.

최종적으로,

1-(1)과 2-(1)의 종합 score를 토대로 주어진 문제의 모델 정합성을 판단 할 때,

제공된 데이터의 특성과 문제 해결 복잡도가 비교적 낮으므로, Valid set의 추론 결과가 두 모델 모두 유사하게 나타나고 있습니다. 때문에 어떤 모델이 더 낫다를 판단하기는 어렵습니다.

**2. Decision tree (35pts)**

In this question, you have to train decision tree models to classify the quality levels of different red wines (“quality” variable). The quality level of red wines is one of low(0), medium(1), and high(2). The explanatory variables are as follows:

- fixed acidity

- volatile acidity

- citric acid

- residual sugar

- chlorides

- free sulfur dioxide

- total sulfur dioxide

- density

- pH

- sulphates

- alcohol

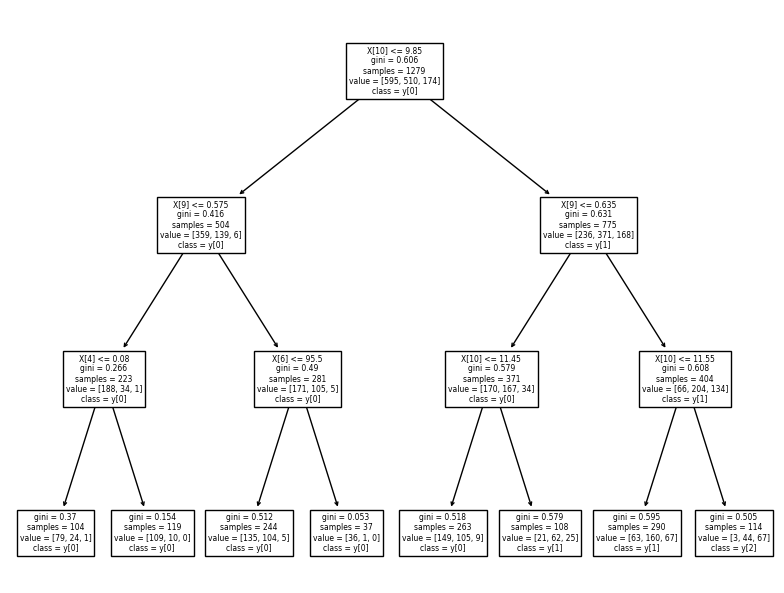
(1) Train decision tree models with different maximum depths such as (1, 2, 3, 4, 5). For training the models, set min\_samples\_leaf = 10 and use the Gini impurity as the criterion to determine the best split, using a training set (trnX, trnY). Then, calculate the accuracy of the models using a validation set (valX, valY) for overall samples and individual classes, and fill the following table. (5pts)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Depth | overall accuracy | Low | Medium | High |
| 1 | 0.569 | 0.697 | 0.575 | 0.866 |
| 2 | 0.528 | 0.637 | 0.553 | 0.866 |
| 3 | 0.597 | 0.688 | 0.616 | 0.891 |
| 4 | 0.644 | 0.753 | 0.650 | 0.884 |
| 5 | 0.619 | 0.719 | 0.631 | 0.887 |

(2) Based on the results of Question (1), which model is the best? Describe your rationale. (5pts)

Depth 가 4일경우 가장 성능이 좋습니다. 이유는 전체 acc 뿐 아니라 각 class에 대한 성능이 우수 하기 때문입니다. 이는 f1 score 측면에서도 우수하기 때문입니다.

(3) Draw the trained tree with feature names when the maximum depth is set to 3. (3pts)



(4) According to the tree drawn for Question (3), explain the rule for class “high” that contains the most cases of class “high.” (3pts)

Decision Tree에서 class "high" 중 가장 많은 경우를 포함하는 규칙을 설명하기 위해서는 이 클래스에 해당하는 leaf node까지의 path를 검토해야 합니다. 이 path는 입력 instance가 "high"로 분류되기 위해 만족해야 하는 조건들을 나타냅니다.

Decision Tree에서 root node에서 leaf node까지의 path는 해당 클래스에 속하는 입력 instance로 분류하기 위해 만족해야 하는 조건들의 집합을 나타냅니다. 이 path에서 각각의 internal node는 입력 feature 중 하나에 대한 binary Decision을 나타내고, edge는 그 판단의 가능한 결과 (예: true 또는 false)를 나타냅니다. "high" 소득 클래스의 조건은 root node에서 해당 leaf node까지의 path를 따라 결정들을 결합하여 얻을 수 있습니다.

(5) According to the tree drawn for Question (3), explain the rule for class “low” that contains the most cases of class “low.” (3pts)

“low” class의 판단 기준은 각 leaf node에서 확인되는 변수를 확인하면서 설명 할 수 있습니다. 해당 tree 에서 확인하면, X의 10 번째 인자가 특정 범위 이하를 만족할 때 “low” class로 분류되기 위한 조건을 생성합니다.

이런 식으로 각 변수들 간의 조건을 생성하여 설정한 depth 까지 node의 조건으로 나눠진 값으로 class “low”를 결정하게됩니다.

(6) Convert the target variable into have two classes by combining the medium and high classes into one classes. Then, train decision tree models by different maximum depths such as (1, 2, 3, 4, 5). Set min\_samples\_leaf = 10 and use the Gini impurity as the criterion to determine the best split, using a training set (trnX, trnY). Then, calculate the accuracy of the models using a validation set (valX, valY) for overall samples and individual classes, and fill the following table. (5pts)

|  |  |  |  |
| --- | --- | --- | --- |
| Depth | overall accuracy | Low | Medium, high |
| 1 | 0.697 | 0.697 | 0.697 |
| 2 | 0.647 | 0.647 | 0.647 |
| 3 | 0.666 | 0.666 | 0.666 |
| 4 | 0.722 | 0.722 | 0.722 |
| 5 | 0.738 | 0.738 | 0.738 |

(7) Compare the two tree models of maximum depth 1 and 2 obtained for Question (6). (5pts)

max\_depth 1 acc : 0.696875

max\_depth 2 acc : 0.646875

**3. -means clustering (30pts)**

This problem uses data generated from four normal distributions to apply k-means clustering.

`y’ variable denotes which normal distribution generates individual samples.

k-means implemented in sci-kit learn can assign initial centroids through ‘init’. When init is set as by array ( = the number of clusters, = the number of features), each row is used as a centroid.

Ref: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

(1) Select randomly 4 samples from the given data set and use them as initial centroids. This procedure is repeated for 100 times. Then, calculate the average values of the silhouette coefficient and adjusted rand index values for 100 iterations. (6pts)

For random selection of samples, refer the following page.

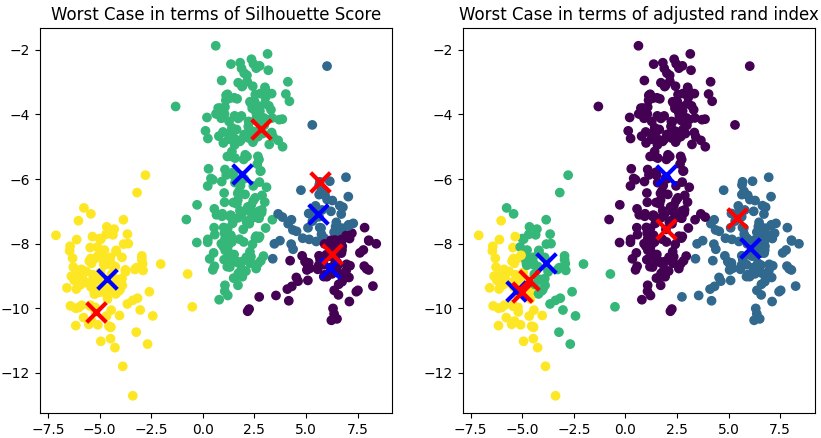
Ref: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.choice.html>

|  |  |
| --- | --- |
| silhouette coefficient | adjusted rand index |
| 0.556 | 0.85 |

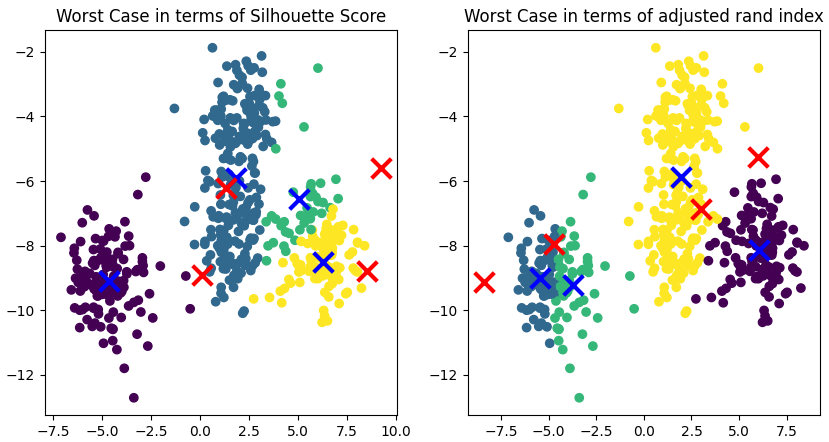
(2) Select randomly one sample from each normal distribution and use them as initial centroids. This procedure is repeated for 100 times. Then, calculate the average values of the silhouette coefficient and adjusted rand index values for 100 iterations. (6pts)

|  |  |
| --- | --- |
| silhouette coefficient | adjusted rand index |
| 0.574 | 0.89 |

(3) Draw scatter plots for the given data with initial centroids and final centroids for the worst cases among 100 trials in Question (1) in terms of silhouette coefficient and adjusted rand index, respectively. The initial centroids should be marked as red ‘X’ and the final centroids should be marked as blue ‘X’. (6pts)



(4) Draw scatter plots for the worst case of Question (2) in the same way as in Question (3). (6pts)



(5) Based on the different results from 100 trials for each case (Questions (1) and (2)), compare two different methods to determine initial centroids. (6pts)

첫 번째 결과의 평균 실루엣 점수는 약 0.56이며, 두 번째 결과는 약간 더 높은 0.57입니다. 일반적으로 0.5 이상의 실루엣 점수는 좋은 클러스터링 결과로 간주합니다. 따라서 두 결과 모두 평균 실루엣 점수에서는 좋은 클러스터링 성능을 갖는다 라고 할 수 있습니다.

하지만, adjust rand 점수를 확인 하면, 첫 번째 결과의 평균은 0.85, 두 번째 결과는 더 높은 0.89입니다. Adjust rand index는 -1과 1 사이의 값을 가지며, 1은 완벽한 클러스터링 일치를 나타내며, 0은 무작위로 선택한 것과 다름없는 클러스터링 결과를 나타냅니다. 이러한 특성을 확인 했을 때, 두 번째 결과가 첫 번째 결과보다 약간 더 나은 성능을 보입니다.

종합적으로, 두 결과 모두 좋은 클러스터링 성능을 보여주며, 두 번째 결과는 Average Adjust rand index score에서 조금 더 우수한 결과를 보입니다.