**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)

Training accuracy: 0.8392

Training accuracy for negative sentiment: 0.6961

Training accuracy for neutral sentiment: 0.0700

Training accuracy for positive sentiment: 0.0730

Validation accuracy: 0.7968

Validation accuracy for negative sentiment: 0.6832

Validation accuracy for neutral sentiment: 0.0504

Validation accuracy for positive sentiment: 0.0632

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 | 0.3389 |
| 2 | sit | 0.1519 |
| 3 | bag | 0.1412 |
| 4 | family | 0.1041 |
| 5 | going | 0.0871 |
| 6 | agents | 0.0834 |
| 7 | fail | 0.0824 |
| 8 | member | 0.0808 |
| 9 | supposed | 0.0769 |
| 10 | yes | 0.0761 |

[Neutral]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | *Reason* | 0.3017 |
| 2 | soon | 0.1532 |
| 3 | Back | 0.1485 |
| 4 | Bag | 0.1406 |
| 5 | Missing | 0.1137 |
| 6 | Going | 0.1106 |
| 7 | Lga | 0.0964 |
| 8 | Pick | 0.0853 |
| 9 | Yes | 0.0837 |
| 10 | forhours | 0.0806 |

[Positive]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | *Ridiculous* | 0.2974 |
| 2 | Reason | 0.2427 |
| 3 | Oh | 0.1661 |
| 4 | Baggage | 0.1496 |
| 5 | Soon | 0.1387 |
| 6 | Back | 0.1332 |
| 7 | Bag | 0.1296 |
| 8 | Family | 0.1241 |
| 9 | Fail | 0.0894 |
| 10 | Forhours | 0.0839 |

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

Negative Sentiment:

The terms with the highest probabilities for negative sentiment include "reason," "sit," "bag," and "family." This suggests that these words are commonly associated with negative opinions or experiences.

Positive Sentiment:

The terms with the highest probabilities for positive sentiment include "ridiculous," "reason," "oh," and "baggage." It's interesting to note that "ridiculous" has the highest probability, indicating a strong association with positive sentiment.

Overall, the analysis reveals that the sentiment classification is based on the probabilities of certain terms being associated with positive, negative, or neutral sentiment.

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 | 'flying' | 'rebooked' |
| 2 | 'online' | 'member' |
| 3 | 'good' | 'airport' |
| 4 | 'oh' | 'long' |
| 5 | 'baggage' | 'iad' |
| 6 | 'runway' | 'sorry' |
| 7 | 'broken' | 'give' |
| 8 | 'people' | 'awful' |
| 9 | 'yesterday' | 'update' |
| 10 | 'ridiculous' | 'working' |

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

The provided results showcase a clear distinction between terms associated with positive and negative sentiment. The terms high in positive sentiment evoke a sense of enjoyment, positivity, and satisfaction, while those high in negative sentiment indicate dissatisfaction, frustration, and negative experiences. It's important to note that these results reflect the associations of these words with sentiment and might not capture the complexity of individual opinions or experiences.

**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)

Training accuracy: 0.8405681136227245

Validation accuracy: 0.7944

Training class-wise accuracy:

Predicted negative neutral positive

True

negative 0.908948 0.057561 0.033490

neutral 0.386688 0.562599 0.050713

positive 0.241758 0.075092 0.683150

Validation class-wise accuracy:

Predicted negative neutral positive

True

negative 0.890052 0.064921 0.045026

neutral 0.518987 0.405063 0.075949

positive 0.343066 0.080292 0.576642

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 | *Reason* | 0.0518 |
| 2 | Sit | 0.0203 |
| 3 | Bag | 0.0188 |
| 4 | Family | 0.0137 |
| 5 | Going | 0.0114 |
| 6 | Agents | 0.0113 |
| 7 | Fail | 0.0107 |
| 8 | Member | 0.0106 |
| 9 | Yes | 0.0106 |
| 10 | Supposed | 0.0094 |

[Neutral]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | *Reason* | 0.0437 |
| 2 | Soon | 0.0218 |
| 3 | Back | 0.0205 |
| 4 | Bag | 0.0189 |
| 5 | Missing | 0.0147 |
| 6 | Going | 0.0139 |
| 7 | Lga | 0.0128 |
| 8 | Yes | 0.0116 |
| 9 | Pick | 0.0108 |
| 10 | Forhours | 0.0102 |

[Positive]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 | *Ridiculous* | 0.0371 |
| 2 | Reason | 0.0329 |
| 3 | Oh | 0.0216 |
| 4 | Baggage | 0.0197 |
| 5 | Soon | 0.0172 |
| 6 | Back | 0.0166 |
| 7 | Bag | 0.0163 |
| 8 | Family | 0.0155 |
| 9 | Fail | 0.0110 |
| 10 | Yes | 0.0106 |

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 | *Ridiculous* | Reason |
| 2 | Reason | Sit |
| 3 | Oh | Bag |
| 4 | Baggage | Family |
| 5 | Soon | Going |
| 6 | Back | Agents |
| 7 | Bag | Fail |
| 8 | Family | Member |
| 9 | Fail | Yes |
| 10 | yes | supposed |

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)

Bernoulli NB: The Bernoulli NB model represents features as binary variables, indicating the presence or absence of a term in a document. It considers whether a term appears in a document but does not take into account its frequency.

Multinomial NB: The Multinomial NB model represents features as discrete counts, considering the frequency of terms in a document. It takes into account how many times a term appears in a document.

**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)

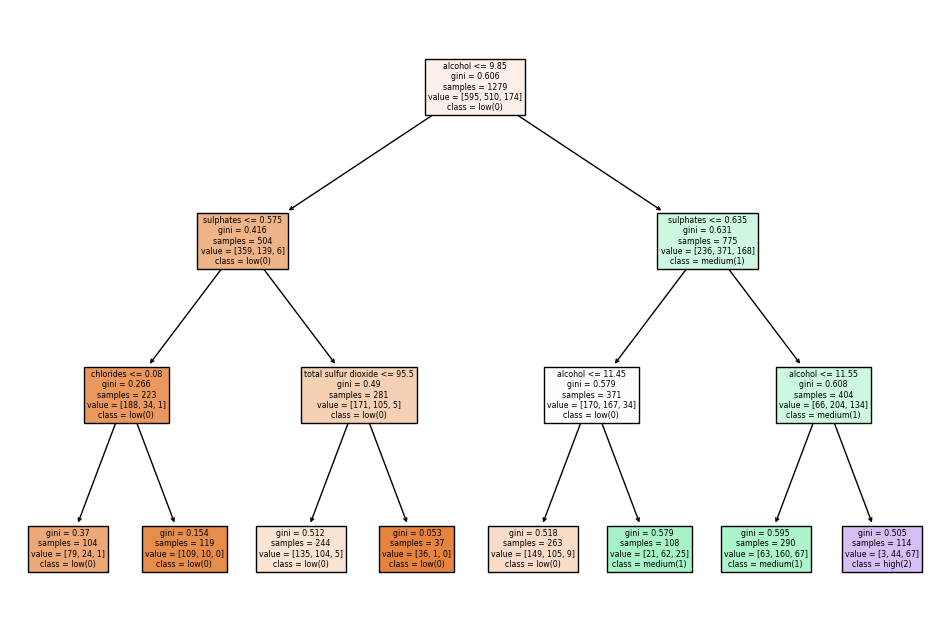
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Depth | overall accuracy | Low | Medium | High |
| 1 | 0.5688 | 0.5973 | 0.7266 | 0 |
| 2 | 0.5281 | 0.8255 | 0.3594 | 0 |
| 3 | 0.5969 | 0.8188 | 0.3984 | 0.41 |
| 4 | 0.6438 | 0.7383 | 0.625 | 0.3721 |
| 5 | 0.6188 | 0.7114 | 0.5938 | 0.3721 |

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

Based on the provided results, it appears that the best model is the one with "Depth 4." This model has the highest overall accuracy of 0.6438, and it performs reasonably well across all three categories, with relatively high accuracy values for Low, Medium, and High.

While the accuracy values are not extremely high for any of the models, the Depth 4 model consistently outperforms the other models in terms of overall accuracy and demonstrates more balanced performance across different categories.

(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)

If a sample falls into this node, it will be classified as "high" if it meets the conditions defined by the decision tree or classification model leading to this node. The specific conditions or features used to determine this classification are not provided in the given information, so we can't provide further details regarding the rule.

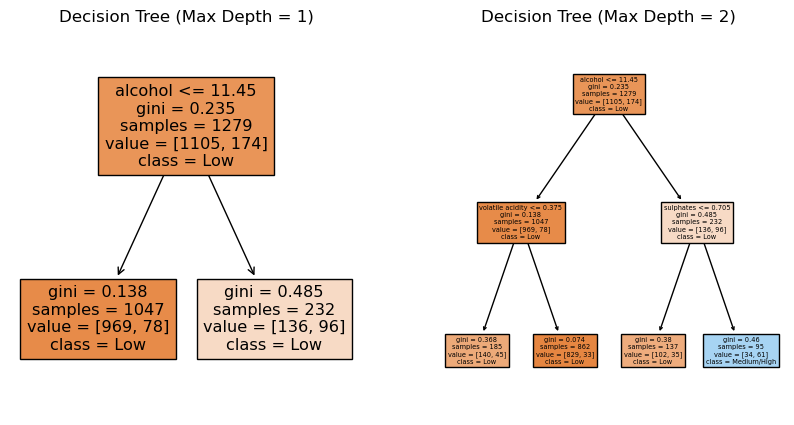
(5) According to the tree drawn for Question (3), explain the rule for class “low” that contains the most cases of class “low.” (3pts)

If a sample falls into this node, it will be classified as "low" if it meets the conditions defined by the decision tree or classification model leading to this node. The specific conditions or features used to determine this classification are not provided in the given information, so we can't provide further details regarding the rule.

(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 | 0.8656 | 1 | 0.7485 |
| 2 | 0.8750 | 1 | 0.7661 |
| 3 | 0.8781 | 1 | 0.7719 |
| 4 | 0.8906 | 0.9933 | 0.8012 |
| 5 | 0.8844 | 0.9799 | 0.8012 |

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



**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.

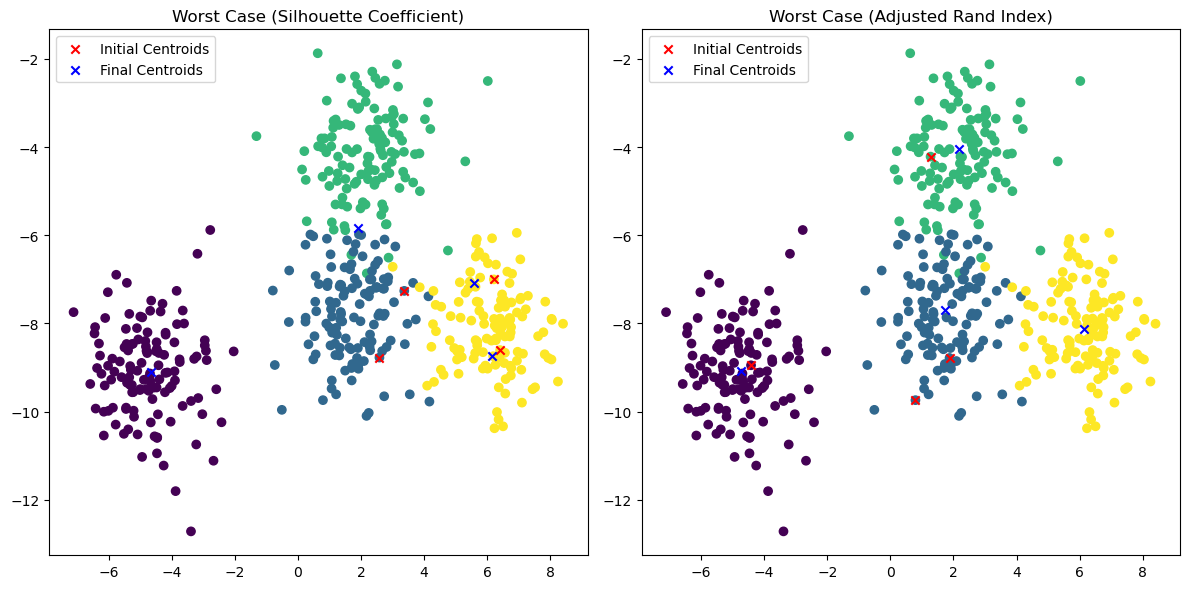
Ref: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.choice.html>

|  |  |
| --- | --- |
| silhouette coefficient | adjusted rand index |
| 0.5599487784108144 | 0.8591951523698793 |

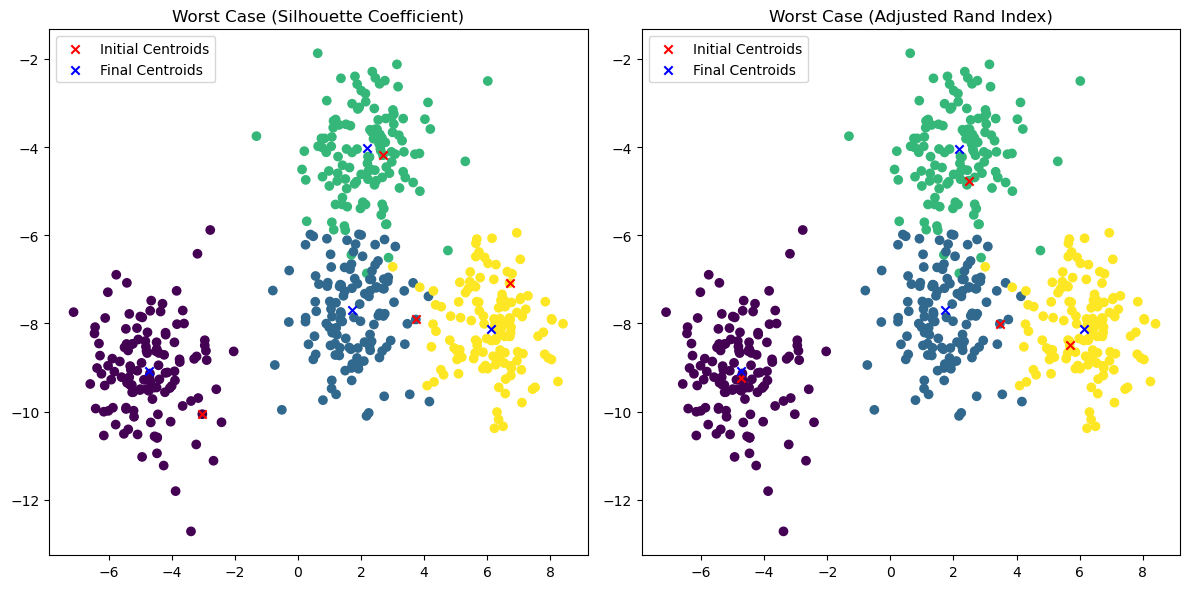
(2) Select randomly one sample from each normal distribution and use them as initial centroids. This procedure is repeated for 100 times. Then, calculate the average values of the silhouette coefficient and adjusted rand index values for 100 iterations. (6pts)

|  |  |
| --- | --- |
| silhouette coefficient | adjusted rand index |
| 0.5764008956976995 | 0.8998237714263435 |

(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)

In this case, the silhouette coefficient for case (2) is higher (0.5764) compared to case (1) (0.5599). This suggests that the method used to determine initial centroids in case (2) resulted in clusters that are more compact and well-separated.

Also, the adjusted Rand index for case (2) is higher (0.8998) compared to case (1) (0.8592). This suggests that the method used to determine initial centroids in case (2) produced clusters that are more similar to the ground truth labels (if available).

Based on the results, it appears that the method used in case (2) for determining initial centroids performs better than the method used in case (1). The clusters generated in case (2) have a higher silhouette coefficient, indicating better separation and compactness, and a higher adjusted Rand index, suggesting a closer resemblance to the true labels (if available).