**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 naïve Bayes (NB) models to classify whether each message is spam or not.

The data used in this problem provide frequencies of words for each message.

The column “label” is the target variable, indicating whether each message is spam or not (the label “ham” means that it is not a spam message.).

**Part 1: Bernoulli NB**

To build a Bernoulli NB, if a certain word is used in a message, the values of the variables are set to 1; otherwise, they are set to 0.

1-(1). After the conversion, train a Bernoulli NB using training set (alpha=1). Prior probabilities of classes are proportional to ratios of classes in training set. Then, calculate the overall accuracy and accuracy values corresponding to each target class (spam) for the training and validation sets, respectively. (6pts)

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

1-(2) Find the top 10 most probable words in each class according to the model trained in Question 1-(1) and summarize them with the probability of existence of words in the following tables. (7pts)

[Spam]

|  |  |  |
| --- | --- | --- |
| **Order** | **Word** | **Probability** |
| 1 | get | 0.075633 |
| 2 | go | 0.058522 |
| 3 | know | 0.058179 |
| 4 | ltgt | 0.058179 |
| 5 | like | 0.057153 |
| 6 | got | 0.054415 |
| 7 | call | 0.053046 |
| 8 | come | 0.053046 |
| 9 | time | 0.045517 |
| 10 | day | 0.04449 |

[Ham]

|  |  |  |
| --- | --- | --- |
| **Order** | **Word** | **Probability** |
| 1 | call | 0.25 |
| 2 | mobile | 0.132143 |
| 3 | claim | 0.108929 |
| 4 | txt | 0.103571 |
| 5 | reply | 0.092857 |
| 6 | prize | 0.092857 |
| 7 | text | 0.089286 |
| 8 | free | 0.0875 |
| 9 | contact | 0.083929 |
| 10 | 16 | 0.071429 |

1-(3) Find the top 10 words whose probability of existence is high in spam messages, but low in ham messages according to the model trained in Question 1-(1). In addition, find the top 10 words whose probability of existence is high in ham messages, but low in spam messages. (3pts)

|  |  |  |
| --- | --- | --- |
| Order | High in spam, low in ham | High in ham, low in spam |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

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

**Part 2: Multinomial NB**

2-(1) Train a multinomial NB using training set (x\_train, y\_train) (alpha=1). Prior probabilities of classes are proportional to ratios of classes in training set. Then calculate the overall accuracy and accuracy values corresponding to each class for the training and validation sets, respectively. (3pts)

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

[Spam]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

[Ham]

|  |  |  |
| --- | --- | --- |
| Order | Term | Probability |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

2-(3) Find the top 10 words whose probability of existence is high in spam messages, but low in ham messages according to the model trained in Question 2-(1). In addition, find the top 10 words whose probability of existence is high in ham messages, but low in spam messages. (3pts)

|  |  |  |
| --- | --- | --- |
| Order | High in spam, low in ham | High in ham, low in spam |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

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

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

In this question, you have to train decision tree models to classify whether the client has subscribed a term deposit or not (“y” variable). If the client has subscribed a term deposit, the target variable would be 1, and 0 otherwise.

[Variables]

|  |  |
| --- | --- |
| **Variable name** | **Description** |
| **Y** | **Dependent variable, indicating whether the client has subscribed a term deposit (binary: yes – 1 or no – 0)** |
| Age | Age of the individuals in years |
| Job | Type of job (nominal: “admin”, “unknown”, “unemployed”, “management”, “housemaid”, “entrepreneur”, “blue-collar”, “self-employed”, “retired”, “technician”, or “services”) |
| Marital | Marital status (nominal: “married”, “divorced”, or “single”) |
| Education | Education level (nominal: “unknown”, “secondary”, “primary”, or “tertiary”) |
| Default | Has credit in default? (binary: “yes” or “no”) |
| Balance | Average yearly balance in euros |
| Housing | Has housing loan? (binary: “yes” or “no”) |
| Loan | Has personal loan? (binary: “yes” or “no”) |
| Contact | Contact communication type (nominal: “unknown”, “telephone”, or “cellular”) |
| Day | Last contact day of the month |
| Month | Last contact month of year (nominal: “Jan”, “Feb”, … “Nov”, “Dec”) |
| Duration | Last contact duration, in seconds |
| Campaign | Number of contacts performed during this campaign and for this client including last contact. |
| Pdays | Number of days that passed by after the client was last contacted from a previous campaign (numeric: -1 means that client was not previously contacted) |
| Previous | Number of contacts performed before this campaign and for this client |
| Poutcome | Outcome of the previous marketing campaign (nominal: “unknown”, “other”, “failure”, “success”) |

(1) Train decision tree models by different maximum depth (1, 2, 3, 4, 5) with min\_samples\_leaf = 10 and the Gini impurity for the criterion to determine the best split, using a training set (x\_train, y\_train). Then, calculate the accuracy of the models using a validation set (x\_valid, y\_valid) for overall samples and individual classes, and fill the following table. (8pts)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Depth | overall accuracy | Low | Medium | High |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

(2) Based on the results of Question (1), which model is the best? Describe your rationale. (6pts)

(3) Draw the trained tree with feature names when the maximum depth is set to 3. (5pts)

(4) Train decision tree models by different maximum depth (1, 2, 3, 4, 5) with min\_samples\_leaf = 10 and the entropy impurity for the criterion to determine the best split, using a training set (x\_train, y\_train). Then, calculate the accuracy of the models using a validation set (x\_valid, y\_valid) for overall samples and individual classes, and fill the following table. (8pts)

|  |  |  |  |
| --- | --- | --- | --- |
| Depth | overall accuracy | No | Yes |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

(5) Compare the two tree models of maximum depth 2 and 3 obtained for Question (4). (8pts)

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

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

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 samples from the given data set and use them as initial centroids. This procedure is repeated for 10 times for each . You need to report the average and standard deviation of silhouette coefficient values of 10 iterations for each . (6pts)

For random selection of samples, refer the following page.

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

|  |  |
| --- | --- |
| The number of clusters | Silhouette coefficient (standard deviation) |
| 3 | Avg (std) |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |

(2) Use the default initialization strategy “k-means++”. This procedure is repeated for 10 times. Then, calculate the average and standard deviation silhouette coefficient values of 10 iterations for each . (6pts)

|  |  |
| --- | --- |
| The number of clusters | Silhouette coefficient (standard deviation) |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |

(3) Draw scatter plots for the given data with initial centroids and final centroids for the worst and best number of clusters among in Question (1) in terms of silhouette coefficient. The final centroids should be marked as blue ‘X’. (6pts)

(4) Draw scatter plots for the worst and best number of clusters among of Question (2) in the same way as in Question (3). Do not need to plot initial centroids. (6pts)

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