CP423 Assignment 3

Technical Report

Obi Ihejirika - 190970850

Jeetindra Dhori - 180726000

Griffin Krunic - 180432940

Group Member Contributions

Questions were distributed among group members as follows:

- Obi Ihejirika completed question 1
- Griffin Krunic completed and Jeetindra Dhori question 2

Question 1 Explanation

Training: training_sentiment.py

Usage:

1) python training_sentiment.py --imdb --naive

load_data(file, database_dir)

- This function loads the data from the specified file and returns it as a pandas dataframe.
- file: a string representing the name of the file to load.
- database_dir: a string representing the directory where the data file is located.

preprocess(text)

- This function preprocesses the input text by converting it to lowercase, removing punctuation and stopwords, and tokenizing the text.
- text: a string representing the text to preprocess.

train_and_evaluate(args, X, y)

- This function trains and evaluates the classifier using the specified arguments, input features (X), and labels (y).
- args: a Namespace object containing the command-line arguments.
- X: a pandas series containing the preprocessed text data.
- y: a pandas series containing the labels for the data.

main()

- This is the main function of the script.
- It parses the command-line arguments to determine which dataset and classifier to use.
- It loads the data, preprocesses it, trains and evaluates the model using cross-validation, and saves the trained model to a file.
- The results of the evaluation are printed to the console, and a confusion matrix is displayed using matplotlib.

if name == 'main':

- This code block is executed when the script is run from the terminal and calls the main function

Sentiment Prediction: predict_sentiment.py

Usage:

2) python predict_sentiment.py "I hate the news. But the sun is shiny. So its a good day."
3) python predict_sentiment.py "I hate the news. It gives too much text to classify"

main()

- This is the main function of the script.
- It parses the command-line argument to get the input text.
- It calls the `predict_sentiment()` function to predict the sentiment of the input text.
- The predicted sentiment is printed to the console.

if name == 'main':

- This code block is executed when the script is run from the terminal and calls the main function

preprocess(text)

- This function preprocesses the input text by converting it to lowercase, removing punctuation and stopwords, and tokenizing the text.
- `text`: a string representing the text to preprocess.

predict_sentiment(text)

- This function loads the pre-trained model, preprocesses the input text, and predicts the sentiment using the loaded model.
- `text`: a string representing the text to predict the sentiment for.
- The function returns the predicted sentiment as a string.

Question 2 Explanation

Clustering - cluster_news.py

Usage:

4) python3 cluster_news.py --ac
5) python3 cluster_news.py --whc
6) python3 cluster_news.py --kmeans
7) python3 cluster_news.py --dbscan
8) python3 cluster_news.py --ncluster

main(args):

- This function uses uses the argparse library to parse command-line arguments to use in the execution of the appropriate clustering functions. The user can define the number of clusters and the clustering algorithms to employ when running the script from the terminal
- It loads and preprocesses news group data, performs clustering using various algorithms based on user input, evaluates clustering performance, and prints the results for different numbers of clusters.

if __name__ == "__main__":

- The user can define the number of clusters and the clustering algorithms to employ by running the script from the terminal. The main function is then executed by using and this function, which uses the argparse library to parse command-line arguments and pass them to the "main" function.

load_news_group_data(path):

- This function loads text documents from a directory and its subdirectories, and returns the text content and corresponding labels for each document. It uses regular expressions and file I/O operations to achieve this. The loaded data is stored in two lists: news_group_data contains the text content of each document, and news_group_labels contains the corresponding labels. The function returns these two lists.

preprocess_news_group_data(data):

- The TfidfVectorizer class from the scikit-learn library is used in this function to transform the text data into a matrix of TF-IDF features. The final matrix is sent back.

reduce_news_group_data(news_groups, n_components=100):

- This programme accepts a dataset of newsgroups and, using truncated singular value decomposition (SVD), reduces its dimensionality to a predetermined number of components (default 100). The condensed dataset is then returned.

cluster_news_group_data(news_groups, n_clusters, algorithm_type):

- This function takes in a dataset of news groups, the number of clusters desired, and the type of clustering algorithm to use. Depending on the algorithm specified, it will cluster the data using either K-means, WHC , AC, DBSCAN

evaluate_clustering_performance(true_data, pred):

- This function computes the adjusted mutual information score , adjusted rand score, and completeness score for clustering using the true labels of a dataset and the anticipated cluster labels. The three scores are then returned as a tuple.

save_news_group_model(model, algorithm, n_clusters):

- takes a clustering model, the algorithm used to create it, and the number of clusters as inputs, then uses the joblib library to save the model as a file in the "models" directory with a filename based on the algorithm and number of clusters.

type_kmeans(news_groups, n_clusters):

- This function clusters news group data using k-means algorithm with a specified number of clusters, saves the resulting model using "save_news_group_model" function, and returns the predicted labels for the data.

type_whc(news_groups, n_clusters):

- This function reduces the news group data dimension using the reduce_news_group_data function, clusters the reduced data using the Ward hierarchical clustering algorithm with a specified number of clusters, saves the resulting model using the "save_news_group_model" function, and returns the predicted labels for the data.

type_ac(news_groups, n_clusters):

- This function reduces the dimension of the news group data using the reduce_news_group_data function, performs agglomerative clustering using the specified number of clusters, saves the resulting model using the "save_news_group_model" function, and returns the predicted labels for the data.

type_dbscan(news_groups, n_clusters):

- This function performs density-based clustering on the news group data using the specified number of clusters, saves the resulting model using the "save_news_group_model" function, and returns the predicted labels for the data.