**Students’ names and I.Ds:**

Eleni Antonakaki, LT1200002

Christina Christodoulou, LT1200027

**REPORT**

**Python programs included:**

1. Split.py
2. Used\_functions.py
3. Train.py
4. Test.py
5. Experiments.py
6. Feature\_selection.py

**Files included:**

1. The source xml file: “ABSA16\_Restaurants\_Train\_SB1\_v2.xml”
2. The folder where the ten xml parts are stored: “ten\_xml\_parts”
3. The trained model in pickle: “Logistic\_regression\_model.sav”

* If you wish to run the Split.py, please delete the folder “ten\_xml\_parts” so that it does not overwrite the xml parts.

We converted the xml parts into dataframes, pre-processed the sentences in the “Text” column in each dataframe (“data\_xml\_to\_df” function in Used\_functions.py) and finally created a general dataframe. From the general dataframe, we focused on the columns “Text” as feature (X) and “Polarity” as label (y), (“open\_parts\_to\_dataframes” function in Used\_functions.py). We also created a function in Used\_functions.py named “select\_random\_parts” that takes as input a folder and the number of files we want to process. The function shuffles the files and separates them into files for training and files for testing with no duplicates. The function takes different files each time the programs are run, so the results always change.

We built a language model based on TF-IDF scores from the sentences in the reviews and converted polarity's categorical values into numerical labels (0,1,2,3) using label encoding. We trained a logistic regression model and saved it in pickle form (“train” function in Train.py). We predicted results for a selected number of parts using the saved model in pickle form (“predict\_polarity” function in Test.py). The “train” and “predict\_polarity” functions are mainly used in Experiments.py and in Feature\_selection.py.

In Experiments.py, there is a function named “cross\_validation” that performs 10-fold cross validation on a different selected number of files each time the program is run.

**Results from Experiments.py**

9 parts for training, 1 for testing

**9 parts:**

Accuracy of Logistic Regression model - train set: 82.22979552093476 %

Accuracy of Logistic Regression model - test set: 72.76264591439688 %

Cross-Validation Accuracy for each fold:

[0.735 0.732 0.72 0.716 0.685 0.416 0.502 0.696 0.707 0.719]

Average accuracy for each 10-fold cross validation: 0.6628115880350195

**1 part:**

Accuracy of Logistic Regression model - train set: 80.43478260869566 %

Accuracy of Logistic Regression model - test set: 74.46808510638297 %

Cross-Validation Accuracy for each fold:

[0.75 0.696 0.87 0.696 0.739 0.87 0.609 0.522 0.652 0.435]

Average accuracy for each 10-fold cross validation: 0.683695652173913

3 parts for training, 7 for testing

**3 parts:**

Accuracy of Logistic Regression model - train set: 81.53846153846153 %

Accuracy of Logistic Regression model - test set: 63.80368098159509 %

Cross-Validation Accuracy for each fold:

[0.744 0.707 0.488 0.84 0.543 0.605 0.79 0.407 0.704 0.654]

Average accuracy for each 10-fold cross validation: 0.6482234266787111

**7 parts:**

Accuracy of Logistic Regression model - train set: 83.31234256926953 %

Accuracy of Logistic Regression model - test set: 63.81909547738693 %

Cross-Validation Accuracy for each fold:

[0.673 0.417 0.508 0.523 0.714 0.683 0.687 0.747 0.692 0.717]

Average accuracy for each 10-fold cross validation: 0.6361022283132836

6 parts for training, 4 for testing

**6 parts:**

Accuracy of Logistic Regression model - train set: 80.19401778496362 %

Accuracy of Logistic Regression model - test set: 67.41935483870968 %

Cross-Validation Accuracy for each fold:

[0.781 0.703 0.561 0.658 0.684 0.639 0.761 0.701 0.61 0.714]

Average accuracy for each 10-fold cross validation: 0.6813070800167574

**4 parts:**

Accuracy of Logistic Regression model - train set: 83.51648351648352 %

Accuracy of Logistic Regression model - test set: 65.33864541832669 %

Cross-Validation Accuracy for each fold:

[0.381 0.516 0.56 0.608 0.592 0.728 0.664 0.736 0.696 0.704]

Average accuracy for each 10-fold cross validation: 0.6184825396825396

According to the results above, we noticed that there were not many striking differences among the iterations with different number of parts. By comparing the 9,3 and 6 parts for training, it is evident that the 3 parts offer the lowest results. While the accuracy on the test set of the 9 parts is higher than the accuracy of the 6 parts, the average cross validation score of the 6 parts is higher of the average cross validation score of the 9 parts. By comparing the 1,7 and 4 parts for testing, the results show that the 1 part offers the highest results both on the saved model and on the new model in cross validation.

The accuracy scores in the train set seems to be higher than the test set, perhaps due to the fact that 80% was allocated to the train set and 20% to the test set.

In general, we noticed that the trained logistic regression model offers satisfactory results regardless of the number of parts. Although, the higher the number of parts allocated for training, the higher the results.

**Results from Feature\_selection.py**

Each time the program is run, a different number of features is selected, the n most important features change and the results vary.

9 parts for training, 1 for testing

**Example 1:**

Accuracy of Logistic Regression model - train set: 81.99704869650762 %

Accuracy of Logistic Regression model - test set: 73.87033398821218 %

Cross-Validation Accuracy for each fold:

[0.722 0.435 0.512 0.681 0.732 0.776 0.626 0.756 0.685 0.705]

Average accuracy for each 10-fold cross validation: 0.6629303689979928

**Example 2:**

Accuracy of Logistic Regression model - train set: 82.22003929273085 %

Accuracy of Logistic Regression model - test set: 72.4950884086444 %

Cross-Validation Accuracy for each fold:

[0.745 0.694 0.733 0.694 0.714 0.689 0.724 0.705 0.449 0.504]

Average accuracy for each 10-fold cross validation: 0.6651258298595029

According to the results above, it is evident that feature selection before training can offer better results, but without any striking difference comparing to the results from the Experiments.py, where no feature selection was used. There were slight differences in the accuracies of the test sets between the two logistic regression models, while the accuracies on 10-fold cross validation were about the same.