Introdution:

In this assignment, I concatenate the validation and training sets in order to use them in cross-validation. Then, I calculate the f1-accuracy for the validation, training, and test sets.

Part 1: BBoW

a As a baseline, report the performance of the random classifier and the majority-class classifier.

```
Accuracy in random approach                   : 25.59%
Accuracy in most frequent approach (Class1)   : 14.18%
```

c Report the list of hyper-parameters you considered for each classifier, their range, as well as the best values for these hyper-parameters, chosen based on the validation set performance.

Bernouli Naïve Bayes:

```
Params:
      alpha: 10^k, k∈(0,1)

Best Params:
      alpha: 1.259
```

Decision Tree:

```
Params:
       min_samples_leaf        : 2 , 10, 50 ,
       max_leaf_nodes          : 50, 75, 100,
       max_depth               : 20, 25, 35  , 60
      ,
       criterion               : entropy,
       ccp_alpha               : 10^k, k∈(-3,-1)

Best Params:
      max_depth                : 25,
      max_leaf_nodes           : 75,
      min_samples_leaf         : 2
```

Logistic Regression:

```
Params:
        solver                  : saga,
        penalty                 : l1 ,
        C                       : 1, 5, 10, 25, 50,
      ,
        solver                  : saga,
        penalty                 : l2,
        C                       : 1, 5, 10, 25, 50,
        random_state            : 0

Best params:
        C                       : 1,
        penalty                 : l1,
        solver                  : saga
```

SVC:

```
Params:
        penalty                 : l2,
        loss                    : hinge, squared_hinge,
        C                       : 5, 10, 20, 35, 50, 75, 120,
      ,
        penalty                 : l1,
        loss                    : squared_hinge,
        C                       : 5, 10, 20, 35, 50, 75, 120,

Best Params:
       C                        : 10,
       loss                     : squared_hinge,
       penalty                  : l2
```

d Report the training, validation, and test F1-score for all the classifiers.

Bernouli Naïve Bayes:

```
Best f1_macro of 46.51% in cross-validation.

f1-macro on Train set: 51.71%
f1-macro on Valid set: 50.55%
f1-macro on Test  set: 45.95%
```

Decision Tree:

```
Best f1_macro of 61.37% in cross-validation.

f1-macro on Train set: 67.71%
f1-macro on Valid set: 64.78%
f1-macro on Test  set: 62.01%
```

LogisticRegression:

```
Best f1_macro of 52.70% in cross-validation.

f1-macro on Train set: 79.77%
f1-macro on Valid set: 79.77%
f1-macro on Test  set: 52.12%
```

SVC:

```
Best f1_macro of 54.73% in cross-validation.

f1-macro on Train set: 79.81%
f1-macro on Valid set: 78.66%
f1-macro on Test  set: 51.18%
```

e Comment on the performance of different classifiers. Why did a particular classifier perform better than the rest? What was the role of the hyper-parameters in finding the best results.

First, all classifiers perform better than random and most frequent classifiers. The performance in Bernoulli naïve bayes (BNB) and decision tree (DT), unlike Logistic regression and SVC, shows the models are not overfitted since the f1-score in the Valid, Train, and Test sets are not much different. So that the difference for LR and SVC (linear models) is near 30%, and it is a reason to be overfitted. DT has near f1-scores and also has the most f1-score on the test set.

DT and BNB are non-linear classifiers. However, BNB has statistical assumptions of bernoulli and independent distribution of features, while these assumptions are not valid in words. Therefore, without this assumption and the power of more complex decision boundaries, DT has better performance.

Hyper-parameters have an essential role in gaining an optimal model. They may result in increasing the learning speed, accuracy and also using to prevent overfitting. In this assignment, without hyper-parameter tuning, the accuracy is far from the model with tuned ones.

Mostafa DadkhahKalateh
2153428

Part 2: FBoW

b Train Naive Bayes, Decision Tree, Logistic regression and Linear SVM for this task.

Gaussian Naïve Bayes:

```
Params:
        var_smoothing: 10^k, k∈(-3,-2.5)

Best Params:
        var_smoothing: 0.00147
```

Decision Tree:

```
Params:
        min_samples_leaf        : 2, 5, 10, 50,
        max_leaf_nodes          : 35, 50, 75,
        max_depth               : 10, 15, 20, 25
       ,
        criterion               : entropy,
        ccp_alpha               : 10^k, k∈(-4,0)

Best Params:
        max_depth               : 20
        max_leaf_nodes  : 50
        min_samples_leaf        : 2
```

LogisticRegression:

```
Params:
        solver                  : saga,
        penalty                 : l1,
        C                       : 1, 5, 10, 25, 50,
        random_state            : 0
       ,
        solver                  : saga,
        penalty                 : l2,
        C                       : 1, 5, 10, 25, 50,
        random_state            : 0

Best params:
        C                       : 25,
        penalty                 : l1,
        solver                  : saga
```

SVC:

```
Params:
        penalty                 : l2,
        loss                    : hinge, squared_hinge,
        C                       : 5, 10, 20, 35, 50, 75, 120,
      ,
        penalty                 : l1,
        loss                    : squared_hinge,
        C                       : 5, 10, 20, 35, 50, 75, 120,

Best Params:
      C                         : 20,
      loss                      : squared_hinge,
      penalty                   : l2
```

c Report the training, validation, and test F1-score for all the classifiers.

Gaussian Naïve Bayes:

```
f1_macro of 44.27% in cross-validation.

f1-macro on Train set: 54.26%
f1-macro on Valid set: 53.18%
f1-macro on Test  set: 44.13%
```

Decision Tree:

```
Best f1_macro of 62.21% in cross-validation.

f1-macro on Train set: 67.35%
f1-macro on Valid set: 66.88%
f1-macro on Test  set: 64.47%
```

LogisticRegression

```
Best f1_macro of 48.84% in cross-validation.

f1-macro on Train set: 55.43%
f1-macro on Valid set: 52.82%
f1-macro on Test  set: 47.26%
```

SVC:

```
Best f1_macro of 44.77% in cross-validation.

f1-macro on Train set: 59.06%
f1-macro on Valid set: 57.21%
f1-macro on Test  set: 40.24%
```

d Comment on the performance of different classifiers. Why did a particular classifierperform better than the rest? What was the role of the hyper-parameters in findingthe best results.

In all models, the cross-validation and test set accuracy are near each other, which shows the models are not involved with high variance problem and they are more than random and the most frequent classifiers as well. However, in SVC, the performance of Training and validation sets is higher than that of cross-validation and test sets. This can be a reason for overfitting. DT is a low bias and low variance model with the most test accuracy.

Linear and quadratic classifiers (Logistic Regression (LR), SVC, and Gaussian Naïve Bayes (GNB)) have a less test set performance rather than DC. This is because of the more power of DC in complex decision boundaries.

As mentioned above, Hyper-parameters tuning helps to increase the learning speed, accuracy and prevent overfitting. They are necessary to gain an optimal model.

e Compare the performance with the binary bag-of-words based classifiers. Why is there a difference in the performance? Give a brief explanation comparing BBoW Naive Bayes and FBoW Naive Bayes and similarly for other models.

The performance in the linear models (LR and SVC) gets decreased. This is because the complexity of the model is more in FBoW since it uses a continuous, instead of binary, variable.

In non-linear models, DT has better performance since it can benefit from considering different thresholds in FBoW for continuous variables.

f Which representation is better? Why?

In this problem, FBOW is better for two reasons: the representation of whether a variable exists in text or not has less information than the ratio of repetition number of the word over number of all words in the text.