T.C.

MARMARA UNIVERSITY

FACULTY of ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CSE4288 Introduction to Machine Learning

Term Project Model Evaluation Report

Group Members

Student ID	Name Surname	Contact
150120013	İrem Aydın	irmaydin14@gmail.com
150120049	Aksanur Konuk	aksanurkonuk@gmail.com
150120054	Elife Kocabey	kocabeyelife@gmail.com
150120066	Zeynep Yılmaz	zenepyilmaz66@gmail.com
150121013	İrem Kıranmezar	iremkiranmezar@gmail.com

Evaluation Metrics and Results

1. Decision Tree (Gini)

The Decision Tree (Gini) model achieves an accuracy of **67.6%**, indicating moderate performance. It performs better in identifying negative sentiment (0) **(F1-score: 0.69)** than positive sentiment (1) **(F1-score: 0.66)**, with lower recall for positive sentiment (0.63), suggesting challenges in detecting all true positives. The confusion matrix highlights a balance in predictions but a tendency to misclassify instances of positive.

```
Model: DECISION TREE (GINI)
Accuracy Score: 0.67553
Confusion MAtrix:
[[702 268]
 [354 593]]
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.72      0.69       970
           1       0.69      0.63      0.66       947

    accuracy                           0.68      1917
   macro avg       0.68      0.67      0.67      1917
weighted avg       0.68      0.68      0.67      1917

Cross-Validation Scores:
fit_time          10.85459
score_time         0.00855
test_accuracy      0.66971
test_precision     0.68103
test_recall        0.63440
test_f1            0.65648
dtype: object
```

2. Decision Tree (Gain)

The Decision Tree (Gain) model has demonstrated a reasonable performance with an accuracy rate of **67.3%**. The recall for negative sentiment is **0.72**, showing better performance, while the recall for positive sentiment is lower at **0.62**. Both macro and weighted average F1 scores are at **0.67,** indicating a balanced performance. Cross-validation results suggest that the model is consistent and has good generalization ability. It may be necessary to make further adjustments, especially to address class imbalances. Additionally, the model's training time is quite efficient.

```
Model: DECISION TREE (GAIN)
Accuracy Score: 0.67345
Confusion MAtrix:
[[702 268]
 [358 589]]
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.72      0.69       970
           1       0.69      0.62      0.65       947

    accuracy                           0.67      1917
   macro avg       0.67      0.67      0.67      1917
weighted avg       0.67      0.67      0.67      1917

Cross-Validation Scores:
fit_time          10.32016
score_time         0.00578
test_accuracy      0.66997
test_precision     0.67687
test_recall        0.64591
test_f1            0.66051
dtype: object
```

3. Naive Bayes

The Naive Bayes model achieves an accuracy of **59.4%**, indicating modest performance. It performs better in identifying negative (F1-score: 0.66) than positive sentiment (F1-score: 0.49), with significantly lower recall for positive seniment (**0.39**), suggesting challenges in detecting true positive cases. The confusion matrix reveals a tendency to misclassify positive instances as negative, as 574 positive cases are misclassified compared to 204 negatives. Overall, the model shows limitations in balancing predictions, particularly for positive cases.

```
Model: NAIVE BAYES
Accuracy Score: 0.59416
Confusion MAtrix:
[[766 204]
 [574 373]]
Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.79      0.66       970
           1       0.65      0.39      0.49       947

    accuracy                           0.59      1917
   macro avg       0.61      0.59      0.58      1917
weighted avg       0.61      0.59      0.58      1917

Cross-Validation Scores:
fit_time          0.16465
score_time        0.01901
test_accuracy     0.59105
test_precision    0.65253
test_recall       0.38524
test_f1           0.48423
dtype: object
```

4. K-Nearest Neighbor

The K-Nearest Neighbors (KNN) model achieved an accuracy of **67.92%**. The confusion matrix shows 638 true negatives, 664 true positives, 332 false positives, and 283 false negatives. The classification report indicates a precision of **69%** for class 0 (negative sentiment) and 67% for class 1 (positive sentiment), with recall of **66%** for negative and **70%** for positive. The F1-scores for both classes are relatively close, with 0.67 for negative sentiment and 0.68 for positive sentiment, suggesting a balanced but suboptimal performance. Cross-validation results show a lower test accuracy of **64.31%**, along with test precision (**64.80%**), recall (**64.59%**), and F1-score (**63.81%**) all indicating that KNN struggled with generalizing across different splits.

```
Model: K-NEAREST NEIGHBORS (KNN)
Accuracy Score: 0.67919
Confusion MAtrix:
[[638 332]
 [283 664]]
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.66      0.67       970
           1       0.67      0.70      0.68       947

    accuracy                           0.68      1917
   macro avg       0.68      0.68      0.68      1917
weighted avg       0.68      0.68      0.68      1917

Cross-Validation Scores:
fit_time          0.03492
score_time        0.21386
test_accuracy     0.64310
test_precision    0.64797
test_recall       0.64590
test_f1           0.63807
dtype: object
```

5. Logistic Regression

Logistic Regression achieved an accuracy of **75.5%**, demonstrating a solid performance in sentiment analysis. The confusion matrix shows 702 true negatives, 745 true positives, 268 false positives, and 202 false negatives, with a slight bias towards predicting positive sentiment. The classification report reveals a precision of **78%** for class 0 (negative sentiment) and **74%** for class 1 (positive sentiment), while recall is higher for class 1 (79%) compared to negative (72%). The F1-score for class 1 is 0.76, slightly higher than negative's 0.75, indicating balanced performance. Cross-validation results support this stability, with consistent accuracy (**75.37%**), precision (**73.78%**), recall (**78.51%**), and F1-score (**76.05%**).

```
Model: LOGISTIC REGRESSION
Accuracy Score: 0.75483
Confusion MAtrix:
[[702 268]
 [202 745]]
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.72      0.75       970
           1       0.74      0.79      0.76       947

    accuracy                           0.75      1917
   macro avg       0.76      0.76      0.75      1917
weighted avg       0.76      0.75      0.75      1917

Cross-Validation Scores:
fit_time          0.43347
score_time        0.00405
test_accuracy     0.75372
test_precision    0.73780
test_recall       0.78513
test_f1           0.76054
dtype: object
```

6. Random Forest Classifier

The Random Forest Classifier achieves an accuracy of **72.8%**, reflecting solid performance in sentiment classification. The model demonstrates a balanced ability to identify both negative (0) and positive (1) sentiments, with F1-scores of 0.73 and 0.72, respectively. It performs slightly better in identifying negative instances (recall: 0.74) than positive ones (recall: 0.72), indicating a slight challenge in capturing all true positives for the positive class. The confusion matrix reveals a relatively balanced distribution of errors, with 254 false positives (negative misclassified as positive) and 267 false negatives (positive misclassified as negative).

Cross-validation results further confirm the model's consistency, with an average test accuracy of **73.65%** and F1-score of **0.7386**, indicating that the model generalizes well across different data subsets without significant overfitting.

```
Model: RANDOM FOREST CLASSIFIER
Accuracy Score: 0.72822
Confusion MAtrix:
[[716 254]
 [267 680]]
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.74      0.73       970
           1       0.73      0.72      0.72       947

    accuracy                           0.73      1917
   macro avg       0.73      0.73      0.73      1917
weighted avg       0.73      0.73      0.73      1917

Cross-Validation Scores:
fit_time          11.85068
score_time         0.05030
test_accuracy      0.73650
test_precision     0.73070
test_recall        0.74719
test_f1            0.73864
dtype: object
```

7. Support Vector Classifier

The Support Vector Classifier (SVC) achieved an accuracy of 75.5%, slightly higher than Logistic Regression. The confusion matrix reveals 698 true negatives, 750 true positives, 272 false positives, and 197 false negatives, with a bias towards correctly predicting positive sentiment (class 1). The classification report shows a precision of 78% for class 0 (negative sentiment) and 73% for class 1 (positive sentiment), while recall is 79% for class 1 and 72% for negative. The F1-scores for both classes are 0.75 for negative and 0.76 for class 1. Cross-validation results indicate stable performance, with a test accuracy of 75.5%, test precision of 73.7%, test recall of 79.1%, and test F1-score of 76.3%.

```
Model: SUPPORT VECTOR CLASSIFIER
Accuracy Score: 0.75535
Confusion MAtrix:
[[698 272]
 [197 750]]
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.72      0.75       970
           1       0.73      0.79      0.76       947

    accuracy                           0.76      1917
   macro avg       0.76      0.76      0.76      1917
weighted avg       0.76      0.76      0.76      1917

Cross-Validation Scores:
fit_time          45.16633
score_time         6.82055
test_accuracy      0.75515
test_precision     0.73721
test_recall        0.79089
test_f1            0.76298
dtype: object
```

8. Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) achieved an accuracy of 67.97%, which is lower compared to Logistic Regression and Support Vector Classifier. The confusion matrix indicates 671 true negatives, 632 true positives, 299 false positives, and 315 false negatives. The classification report shows a precision of 68% for both classes, with recall slightly higher for class 0 (69%) compared to class 1 (67%). The F1-scores for both classes are 0.69 for class 0 and 0.67 for class 1, reflecting a balanced but relatively low performance. Cross-validation results show a test accuracy of 68.64%, test precision of 68.83%, test recall of 67.78%, and test F1-score of 68.28%.

```
Model: ARTIFICIAL NEURAL NETWORK (ANN)
Accuracy Score: 0.67971
Confusion MAtrix:
[[671 299]
 [315 632]]
Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.69      0.69       970
           1       0.68      0.67      0.67       947

    accuracy                           0.68      1917
   macro avg       0.68      0.68      0.68      1917
weighted avg       0.68      0.68      0.68      1917

Cross-Validation Scores:
fit_time          36.27629
score_time         0.00703
test_accuracy      0.68641
test_precision     0.68825
test_recall        0.67782
test_f1            0.68277
dtype: object
```

Comparison Between Models

When we examine the performance of these models, we observe that Logistic Regression and Support Vector Classifier (SVC) achieve the highest accuracy rates. The Logistic Regression model demonstrated a performance similar to SVC with an accuracy of 75.48%. Logistic Regression stands out with its fast execution and balanced metric values. It provided balanced results across classes in terms of precision, recall, and F1-score. Notably, it achieved 78% precision for class 0 and 79% recall for class 1. The cross-validation results also support the test accuracy.

The Support Vector Classifier (SVC) showed a very close performance to Logistic Regression with an accuracy of 75.53%. This model delivered strong metrics, including 78% precision for class 0 and 79% recall for class 1, ensuring balanced performance for both classes. However, due to its significantly longer fit and score times, it may not be suitable for time-critical projects.

The Random Forest Classifier performed slightly lower compared to Logistic Regression and SVC, with an accuracy of 72.82%. The model exhibited similar precision and recall values across classes, achieving 73% precision and recall for

class 1, which is noteworthy. Although the model's fit time is high, its very low score time provides a speed advantage for large datasets.

Naive Bayes emerged as the model with the lowest performance, achieving an accuracy of 59.42%. While it offered high recall (79%) for class 0, its precision and recall values for class 1 were significantly lower, indicating that it struggles to represent class 1 adequately. However, its very short fit and score times make it advantageous for scenarios requiring a fast model.

The Decision Tree (using Gini and Gain criteria) demonstrated moderate performance, with accuracy rates around 67%. Although its precision, recall, and F1-score values are balanced, they are not as strong as those of other models. The performance difference between the Gini and Gain criteria is minimal, so the choice between them does not significantly impact model selection.

The Artificial Neural Network (ANN) achieved an accuracy of 67.97%, performing better than Decision Tree and Naive Bayes but lower than Logistic Regression, SVC, and Random Forest. While the model offered balanced precision and recall values across classes, its very long fit time can be a disadvantage when working with large datasets.

Finally, the K-Nearest Neighbors (KNN) model achieved an accuracy of 67.92%, remaining at a similar accuracy level. Although it provided slightly higher precision for class 0, its overall metrics are not balanced. The model's short fit time but long score time should also be considered in the evaluation.

Overall, Logistic Regression and SVC emerge as the best-performing models for this dataset, with the highest accuracy and balanced metric values. While Random Forest lags slightly in accuracy, it can still be a favorable choice due to its balanced results and fast scoring. The other models are less suitable for this dataset due to their specific limitations and lower performance.

Optimization Steps Taken

We decided to perform optimization on Logical Regression, which has one of the best performance. To optimize, we employed **RandomizedSearchCV** to find the best combination of hyperparameters. The search focused on three key parameters: the regularization strength (C), the solver algorithm (solver), and the maximum number of iterations (max_iter). A range of values was tested for each parameter: C values of 0.1, 1, and 10; solvers 'liblinear' and 'saga'; and iteration counts of 100, 200, and 300. Using 100 random iterations, the model was evaluated using 5-fold cross-validation to identify the best-performing hyperparameter set. The optimal combination of parameters, which resulted in the highest accuracy, was found to be solver='saga', max_iter=100, and C=1. After determining the best parameters, the model was retrained using these values and evaluated on the test set. The performance was measured using accuracy, classification report, and confusion matrix to ensure the model's effectiveness. This optimization process helped enhance the model's performance by fine-tuning its hyperparameters for better prediction accuracy.