```
    p1 = pred_data1_04["Sentiment"][j]
    p2 = pred_data2_04["Sentiment"][j]
    p3 = pred_data3_04["Sentiment"][j]
    p4 = pred_data4_04["Sentiment"][j]

    if p1 == sentiment :
        p1_score04 = p1_score04 + 1
        same1 = "Yes"
    else :
        same1 = "No"

    if p2 == sentiment :
        p2_score04 = p2_score04 + 1
        same2 = "Yes"
    else :
        same2 = "No"

    if p3 == sentiment :
        p3_score04 = p3_score04 + 1
        same3 = "Yes"
    else :
        same3 = "No"

    if p4 == sentiment :
        p4_score04 = p4_score04 + 1
        same4 = "Yes"
    else :
        same4 = "No"

    j = j + 1

    accuracydetail04.add_row([j, same1, same2, same3, same4, sentiment])

    if (same1 == "Yes" and same2 == "Yes" and same3 == "Yes" and same4
 == "Yes") :
        continue
    else :
        accuracydetail_onlydifferent04.add_row([j, same1, same2, same3,
 same4, sentiment])
        n_different04 = n_different04+1

accuracy_comparison04.add_row(["Accuracy",   p1_score04,   p2_score04,
 p3_score04, p4_score04])

print(accuracy_comparison04)
```

0,4 Test Size Comparison – Correctly detected sentiment compared with labelled prediction dataset (Prediction comparison with labelled prediction dataset)

```
print("Accuracy Prediction Comparison Detail")
print(accuracydetail04)
print("Only", n_different04, "Different Prediction")
```

0,4 Test Size Comparison – Correctly detected sentiment compared with labelled prediction dataset (Prediction comparison with labelled prediction dataset – only different)

```
print("Accuracy Prediction Comparison Detail Sort by Different Kernels")
print(accuracydetail_onlydifferent04)
print("Only",   n_different04,   "Different   Prediction")print("Only",
n_different04, "Different Prediction")
```

Importing predicted dataset in test size 0,5 - Linear

```
import numpy as np
import pandas as pd

pred_data1_05  =  pd.read_csv('prediction_data_0,5.csv',  encoding="ISO-
8859-1")

pred_data1_05
```

Importing predicted dataset in test size 0,5 -RBF

```
import numpy as np
import pandas as pd

pred_data2_05 = pd.read_csv('prediction_data2_0,5.csv', encoding="ISO-
8859-1")

pred_data2_05
```

Importing predicted dataset in test size 0,5 - Polynomial

```
import numpy as np
import pandas as pd

pred_data3_05 = pd.read_csv('prediction_data3_0,5.csv', encoding="ISO-
8859-1")

pred_data3_05
```

Importing predicted dataset in test size 0,5 - Polynomial

```
import numpy as np
import pandas as pd

pred_data4_05 = pd.read_csv('prediction_data4_0,5.csv', encoding="ISO-
8859-1")

pred_data4_05
```

0,5 Test Size Comparison – Correctly detected sentiment compared with labelled prediction dataset (Counting)

```
213.accuracy_comparison05 = PrettyTable([" ","Linear", "RBF", "Polynomial",
   "Sigmoid"])
214.accuracydetail05  =  PrettyTable(["No","Linear",  "RBF",  "Polynomial",
   "Sigmoid", "Sentiment"])
215.accuracydetail_onlydifferent05  =  PrettyTable(["No","Linear",  "RBF",
   "Polynomial", "Sigmoid", "Sentiment"])
216.len_dp = len(dataset)
217.
218.p1_score05, p2_score05, p3_score05, p4_score05 = 0, 0, 0, 0
219.n_different05 = 0
220.j = 0
221.
222.for j in range(len_dp):
223.    sentiment = dataset["sentiment"][j]
224.    p1 = pred_data1_05["Sentiment"][j]
225.    p2 = pred_data2_05["Sentiment"][j]
226.    p3 = pred_data3_05["Sentiment"][j]
227.    p4 = pred_data4_05["Sentiment"][j]
228.
229.    if p1 == sentiment :
230.        p1_score05 = p1_score05 + 1
231.        same1 = "Yes"
232.    else :
233.        same1 = "No"
234.
235.    if p2 == sentiment :
236.        p2_score05 = p2_score05 + 1
237.        same2 = "Yes"
238.    else :
239.        same2 = "No"
240.
241.    if p3 == sentiment :
242.        p3_score05 = p3_score05 + 1
243.        same3 = "Yes"
244.    else :
245.        same3 = "No"
246.
247.    if p4 == sentiment :
248.        p4_score05 = p4_score05 + 1
249.        same4 = "Yes"
250.    else :
251.        same4 = "No"
252.
253.    j = j + 1
254.
255.    accuracydetail05.add_row([j, same1, same2, same3, same4, sentiment])
256.
257.    if (same1 == "Yes" and same2 == "Yes" and same3 == "Yes" and same4
   == "Yes") :
258.        continue
259.    else :
260.        accuracydetail_onlydifferent05.add_row([j, same1, same2, same3,
   same4, sentiment])
261.        n_different05 = n_different05+1
262.
```

```
263.accuracy_comparison05.add_row(["Accuracy",    p1_score05,    p2_score05,
   p3_score05, p4_score05])
264.
265.print(accuracy_comparison05)
```

0,5 Test Size Comparison – Correctly detected sentiment compared with labelled prediction dataset (Prediction comparison with labelled prediction dataset)

```
13. print("Accuracy Prediction Comparison Detail")
14. print(accuracydetail05)
15. print("Only", n_different05, "Different Prediction")
```

0,5 Test Size Comparison – Correctly detected sentiment compared with labelled prediction dataset (Prediction comparison with labelled prediction dataset – only different)

```
13. print("Accuracy Prediction Comparison Detail Sort by Different Kernels")
14. print(accuracydetail_onlydifferent05)
15. print("Only",   n_different05,   "Different   Prediction")print("Only",
   n_different05, "Different Prediction")
```

Accuracy Comparison based of counting correctly detected sentiment compared with labelled prediction dataset

```
1. accuracy_comparison   =   PrettyTable(["Test   size",   "Linear",   "RBF",
   "Polynomial", "Sigmoid"])
2. accuracy_comparison.add_row(["0,1", p1_score01, p2_score01, p3_score01,
   p4_score01])
3. accuracy_comparison.add_row(["0,2", p1_score02, p2_score02, p3_score02,
   p4_score02])
4. accuracy_comparison.add_row(["0,3", p1_score03, p2_score03, p3_score03,
   p4_score03])
5. accuracy_comparison.add_row(["0,4", p1_score04, p2_score04, p3_score04,
   p4_score04])
6. accuracy_comparison.add_row(["0,5", p1_score05, p2_score05, p3_score05,
   p4_score05])
7.
8. print("Accuracy Comparison based of Counting Dataset Prediction")
9. print(accuracy_comparison)
```

Accuracy Comparison based of counting correctly detected sentiment compared with labelled prediction dataset in percentage

```
1. accuracy_percentage_comparison   =   PrettyTable(["Test   size",   "Linear",
   "RBF", "Polynomial", "Sigmoid"])
2. accuracy_percentage_comparison.add_row(["0,1",
3.
   (str(format(100*(p1_score01/len_dp), ".2f")) + " %"),
4.
   (str(format(100*(p2_score01/len_dp), ".2f")) + " %"),
```

```
5.
   (str(format(100*(p3_score01/len_dp), ".2f")) + " %"),
6.
   (str(format(100*(p4_score01/len_dp), ".2f")) + " %")])
7. accuracy_percentage_comparison.add_row(["0,2",
8.
   (str(format(100*(p1_score02/len_dp), ".2f")) + " %"),
9.
   (str(format(100*(p2_score02/len_dp), ".2f")) + " %"),
10.
   (str(format(100*(p3_score02/len_dp), ".2f")) + " %"),
11.
   (str(format(100*(p4_score02/len_dp), ".2f")) + " %")])
12. accuracy_percentage_comparison.add_row(["0,3",
13.
   (str(format(100*(p1_score03/len_dp), ".2f")) + " %"),
14.
   (str(format(100*(p2_score03/len_dp), ".2f")) + " %"),
15.
   (str(format(100*(p3_score03/len_dp), ".2f")) + " %"),
16.
   (str(format(100*(p4_score03/len_dp), ".2f")) + " %")])
17. accuracy_percentage_comparison.add_row(["0,4",
18.
   (str(format(100*(p1_score04/len_dp), ".2f")) + " %"),
19.
   (str(format(100*(p2_score04/len_dp), ".2f")) + " %"),
20.
   (str(format(100*(p3_score04/len_dp), ".2f")) + " %"),
21.
   (str(format(100*(p4_score04/len_dp), ".2f")) + " %")])
22. accuracy_percentage_comparison.add_row(["0,5",
23.
   (str(format(100*(p1_score05/len_dp), ".2f")) + " %"),
24.
   (str(format(100*(p2_score05/len_dp), ".2f")) + " %"),
25.
   (str(format(100*(p3_score05/len_dp), ".2f")) + " %"),
26.
   (str(format(100*(p4_score05/len_dp), ".2f")) + " %")])
27.
28. print("Accuracy Comparison based of Percentage Dataset Prediction")
29. print(accuracy_percentage_comparison)
```

PAPER NAME

18.K1.0018.docx

WORD COUNT	CHARACTER COUNT
10391 Words	53571 Characters
PAGE COUNT	FILE SIZE
40 Pages	75.3KB
SUBMISSION DATE	REPORT DATE
Jul 5, 2022 2:00 PM GMT+7	Jul 5, 2022 2:02 PM GMT+7

12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 11% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 9% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)