Intelligent Systems
Dr. Octavio Loyola-Gonzalez

Nicolas Gonzalez Albo Mendez
A01730763

H3 – Testing k-NN using emotion features

Introduction

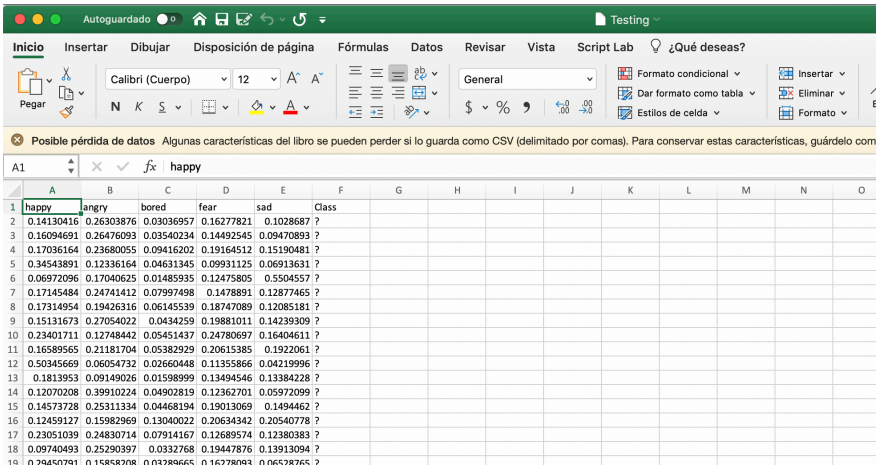
A k-NN algorithm (k-Nearest Neighbor) is a widely used algorithm classifier in machine learning. It works by selecting the nearest class to our data according to the number of neighbors that surround it. The number of neighbors is selected by the user and different types of distances like Euclidean or Manhattan can be chosen.

Development

Using the python script from HW1, with some modifications that excluded sentiment features, emotions from Testing.csv were extracted. The columns from output1-60.csv, output61-120.csv and output121-150 were copied to the original Testing.csv, ID and Text columns were removed. Also, a new Column named “Class” was added and filled with the symbol “?”.

```python
def extract_data():
    with open('Testing121.csv') as csv_file:
        csv_reader = csv.DictReader(csv_file) # Parses the contents of Training.csv as dictionaries

        with open('output.csv','w') as file:
            fieldnames = ['ID', 'Text', 'happy', 'angry', 'bored', 'fear', 'sad']

            csv_writer = csv.DictWriter(file, fieldnames=fieldnames) # Establishes the headers of output.csv
            csv_writer.writeheader()

            for row in csv_reader:
                text = row['Text']
                e_s_dict = api_calls(text) # Calls the APIs of Meaning Cloud and Paralled dots
                print(e_s_dict)
                new_row = dict(row)
                new_row.update(e_s_dict)  # Merges the data of Training.csv with the responses of the APIs
                csv_writer.writerow(new_row)


''' Starts execution of the program '''
if __name__ == "__main__":
    extract_data()
```

Figure 1. output.csv features are shown.



	A	B	C	D	E	F
1	happy	angry	bored	fear	sad	Class
2	0.14130416	0.26303876	0.03036957	0.16277821	0.1028687	?
3	0.16094691	0.26476093	0.03540234	0.14492545	0.09470893	?
4	0.17036164	0.23680055	0.09416202	0.19164512	0.15190481	?
5	0.34543891	0.12336164	0.04631345	0.09931125	0.06913631	?
6	0.06972096	0.17040625	0.01485935	0.12475805	0.5504557	?
7	0.17145484	0.24741412	0.07997498	0.1478891	0.12877465	?
8	0.17314954	0.19426316	0.06145539	0.18747089	0.12085181	?
9	0.15131673	0.27054022	0.0434259	0.19881011	0.14239309	?
10	0.23401711	0.12748442	0.05451437	0.24780697	0.16404611	?
11	0.16589565	0.21181704	0.05382929	0.20615385	0.1922061	?
12	0.50345669	0.06054732	0.02660448	0.11355866	0.04219996	?
13	0.1813953	0.09149026	0.01598999	0.13494546	0.13384228	?
14	0.12070208	0.39910224	0.04902819	0.12362701	0.05972099	?
15	0.14573728	0.25311334	0.04468194	0.19013069	0.1494462	?
16	0.12459127	0.15982969	0.13040022	0.20634342	0.20540778	?
17	0.23051039	0.24830714	0.07914167	0.12689574	0.12380383	?
18	0.09740493	0.25290397	0.0332768	0.19447876	0.13913094	?

Figure 2. Modified Testing.csv showing the new emotion's features

After all the emotion features were extracted, both Training.csv and Testing.csv were imported to Weka and then converted to .arff files for further manipulation. It is important to mention that in both files, Class attribute had to be changed to Nominal.

In the Classify tab of Weka, inside of the Lazy folder, IBk was picked, that is the k-NN classifier that was used for this homework. Considering that we have already uploaded the Training.arff file, The "Supplied test set" is chosen.

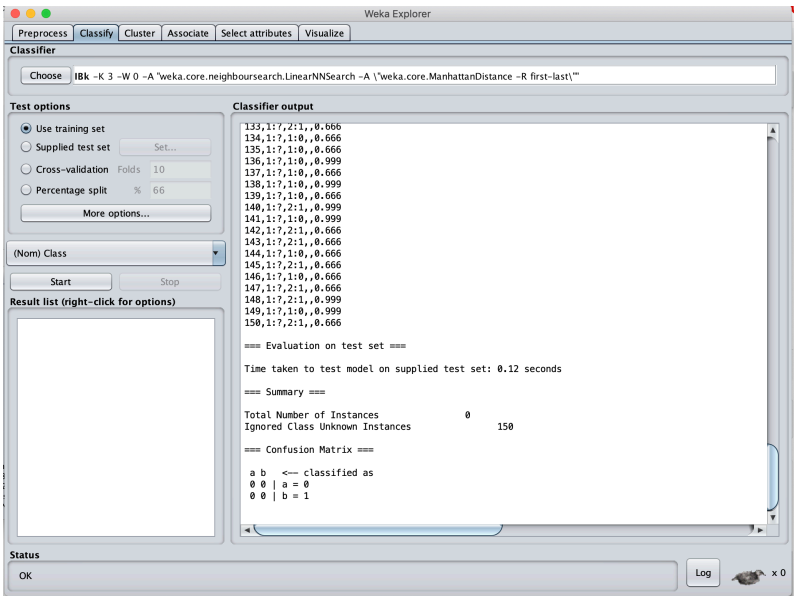


Figure 3. Classify tab of Weka.

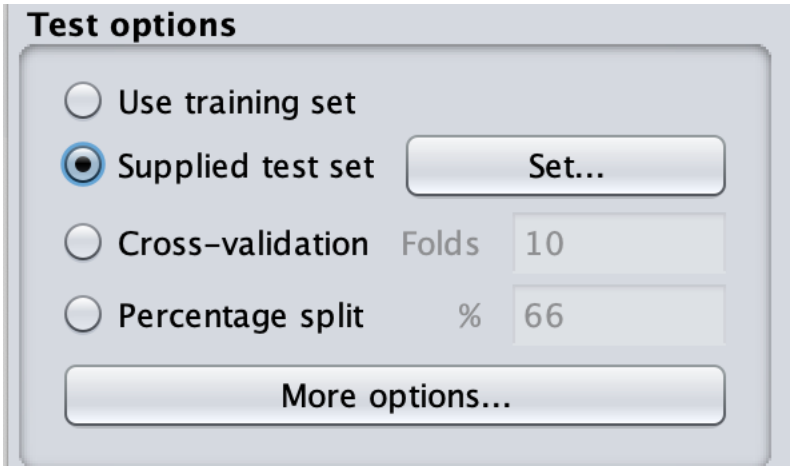


Figure 4. Supplied test set option is picked in Test options

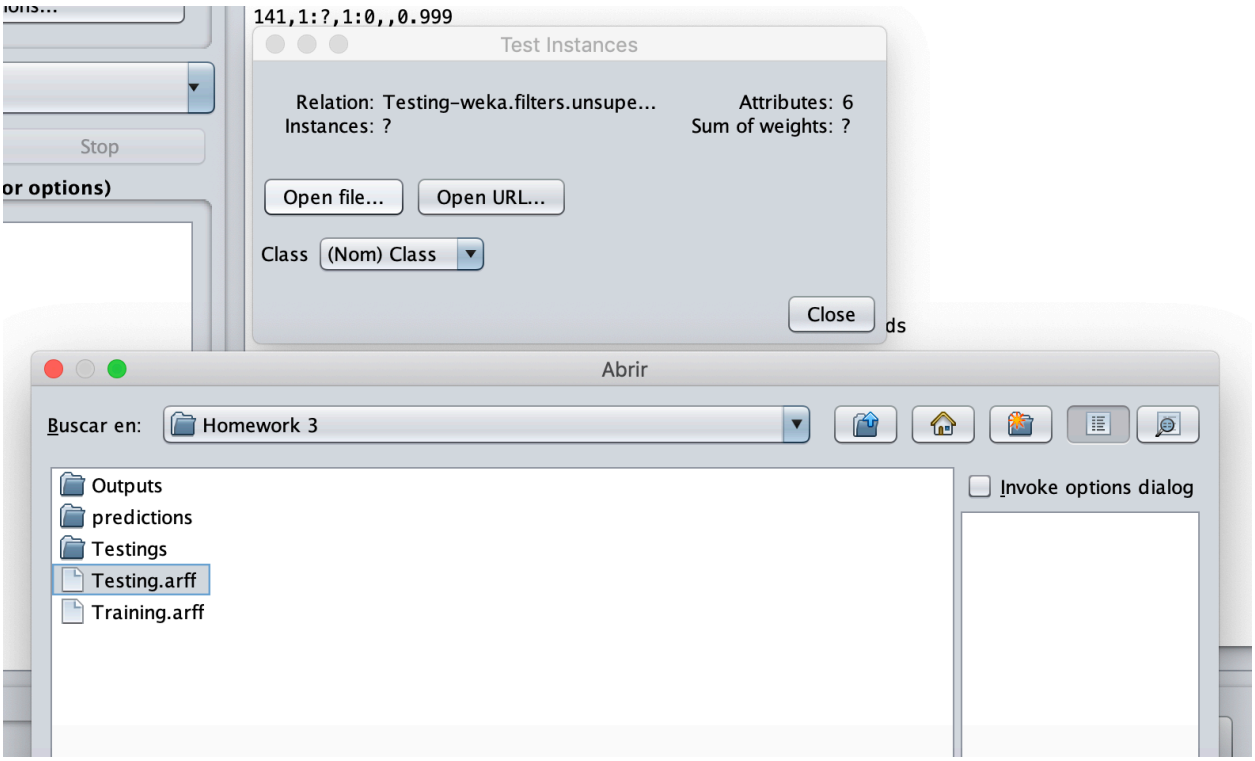


Figure 5. Testing.arff is chosen as the test set.

After all the above steps are made, now we can customize our options for our k-NN classifier.

Nicolas Gonzalez Albo Mendez
A01730763

Results

k-NN with N=3 and Euclidean distance

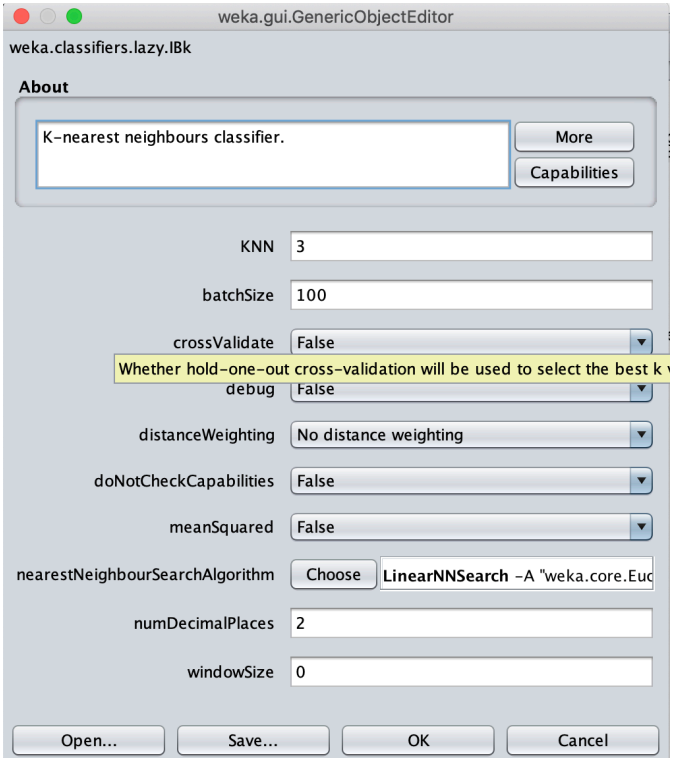


Figure 6. k-NN configuration panel.

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \'
Relation:     Training-weka.filters.unsupervised.attribute.Reorder-R1,2,3,4,5,6-weka.filters.unsupe
Instances:    500
Attributes:   6
              happy
              angry
              bored
              fear
              sad
              Class
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 3 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Predictions on test set ===

inst#,actual,predicted,error,prediction
1,1:?,2:1,,0.999
2,1:?,2:1,,0.666
3,1:?,1:0,,0.999
4,1:?,2:1,,0.666
5,1:?,2:1,,0.666
6,1:?,1:0,,0.999
7,1:?,1:0,,0.999
8,1:?,1:0,,0.999
9,1:?,1:0,,0.999
```

Figure 7. CSV output of the Testing dataset.

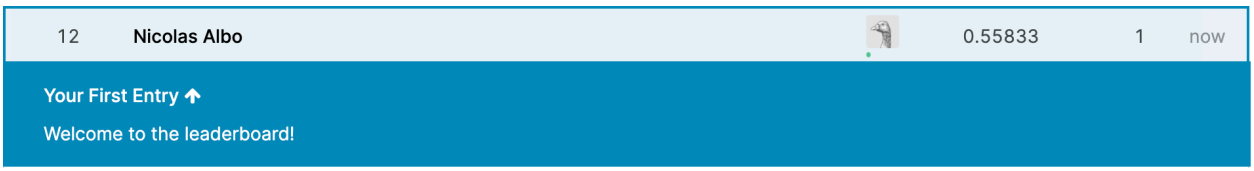


Figure 8. Result obtained in Kaggle.

k-NN with N=3 and Manhattan distance

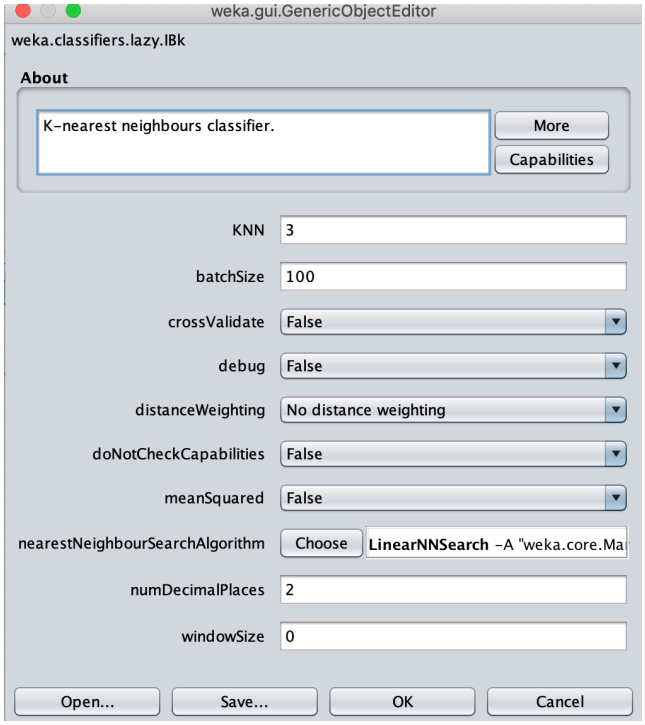


Figure 9. k-NN configuration panel.

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"
Relation:     Training-weka.filters.unsupervised.attribute.Reorder-R1,2,3,4,5,6-weka.filters.unsuper
Instances:    500
Attributes:   6
              happy
              angry
              bored
              fear
              sad
              Class
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 3 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Predictions on test set ===

inst#,actual,predicted,error,prediction
1,1:?,2:1,,0.999
2,1:?,1:0,,0.666
3,1:?,1:0,,0.666
4,1:?,2:1,,0.666
5,1:?,2:1,,0.666
6,1:?,1:0,,0.999
7,1:?,1:0,,0.666
8,1:?,1:0,,0.999
9,1:?,1:0,,0.999
```

Figure 10. CSV output of the Testing dataset.

Nicolas Gonzalez Albo Mendez
A01730763

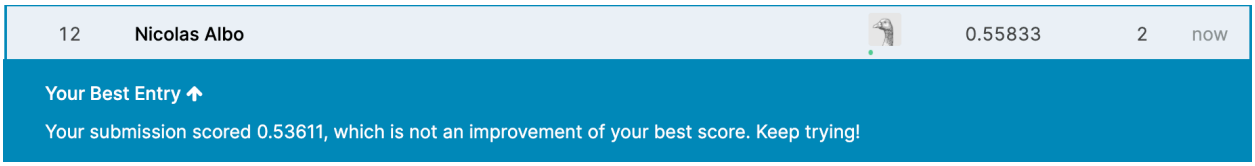


Figure 11. Result obtained in Kaggle.

Conclusion

A k-NN is a powerful algorithm classifier, it is a little bit basic and easy to understand. Maybe my results in Kaggle weren't as high as I expected, but I'm sure that if I keep modifying my k-NN configurations, I can rank higher.