

Machine Learning Project

Fruit Recognition

We first started our project by analysing the data we have in our hands as train data. Train data for the fruit recognition project has some attributes that are just there for connecting the class id to real world definition of the fruit like Genus, Family and there is an attribute called the media id which is irrelevant to the classification of a fruit so we have removed those features from the data at the start since they were all irrelevant to the classification of data. We have used Weka and Python to experiment with the algorithms and classification methods for this project.

Our initial step was to work with ZeroR algorithm as our base approach on our Training Data to see if our files were good for classification. When we imported the CSV training data, the classid was imported as numeric. Classid's value is of no use to us. We need to consider classid as nominal for the classifier to treat it as separate entities to classify to. Also the dates were of the numerical form but they should also be of the form nominal since they can't be perceived as numerical values because they are not of the form of a number. So after we have changed Date and ClassId to nominal values we had 1024 attributes and Longitude, Latitude as numerical attributes and Date and ClassId as nominal values.

ZeroR:

=== Summary ===

Correctly Classified Instances	83	1.0751 %
Incorrectly Classified Instances	7637	98.9249 %
Kappa statistic	0	
Mean absolute error	0.0026	
Root mean squared error	0.0363	
Relative absolute error	100	%
Root relative squared error	100	%
Total Number of Instances	7720	

The result of ZeroR algorithm in Weka on the training set for the fruit project

J48:

After setting the types of all attributes to the suitable form and trying out the training set using ZeroR we started to try the different methods on Weka. First we tried J48 decision tree classifier since we were familiar with the use of it from the homework. We started with using the default options for the J48 and ran it on the Training Data. It returned a 84.8575% correct classification rate with the training data.

=== Summary ===

Correctly Classified Instances	6551	84.8575 %
Incorrectly Classified Instances	1169	15.1425 %
Kappa statistic	0.8481	
Mean absolute error	0.0005	
Root mean squared error	0.0151	
Relative absolute error	18.5617 %	
Root relative squared error	41.5427 %	
Total Number of Instances	7720	

Summary of the results after we ran J48, captured from Weka.

Then we've tried to validate our classification model with a cross validation set of 10 folds.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      3517           45.557 %
Incorrectly Classified Instances    4203           54.443 %
Kappa statistic                     0.4535
Mean absolute error                 0.0015
Root mean squared error             0.0338
Relative absolute error             56.5797 %
Root relative squared error         92.984 %
Total Number of Instances          7720
```

Summary of the results after we tried cross-validation with 10 folds.

After the results of cross-validation we thought our decision tree was overfitting since it was giving a much better performance with training data without any validation and it nearly gave half accuracy with validation, therefore we decided to try pruning the tree a bit to see if we could make it more generalized.

We tried increasing the value of confidenceFactor to 0.3, 0.4, and 0.5. The results were not any better but even worse with 10-fold cross validation. Just to check we even tried reducing it to 0.1.

That gave us much worse accuracy. The results are as follows:

```
=== Summary ===

Correctly Classified Instances      3485           45.1425 %
Incorrectly Classified Instances    4235           54.8575 %
Kappa statistic                     0.4493
Mean absolute error                 0.0015
Root mean squared error             0.034
Relative absolute error             57.0014 %
Root relative squared error         93.4451 %
Total Number of Instances          7720
```

Summary of the cross-validation results with confidenceFactor = 0.3.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	3485	45.1425 %
Incorrectly Classified Instances	4235	54.8575 %
Kappa statistic	0.4493	
Mean absolute error	0.0015	
Root mean squared error	0.034	
Relative absolute error	57.0014 %	
Root relative squared error	93.4451 %	
Total Number of Instances	7720	

Summary of the cross-validation results with confidenceFactor = 0.4.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	3485	45.1425 %
Incorrectly Classified Instances	4235	54.8575 %
Kappa statistic	0.4493	
Mean absolute error	0.0015	
Root mean squared error	0.034	
Relative absolute error	57.0014 %	
Root relative squared error	93.4451 %	
Total Number of Instances	7720	

Summary of the cross-validation results with confidenceFactor = 0.5.

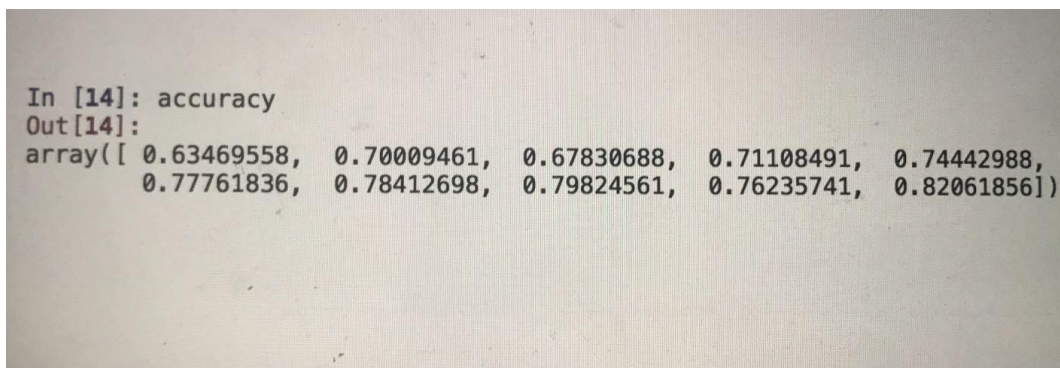
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	3484	45.1295 %
Incorrectly Classified Instances	4236	54.8705 %
Kappa statistic	0.4492	
Mean absolute error	0.0015	
Root mean squared error	0.034	
Relative absolute error	57.0033 %	
Root relative squared error	93.4459 %	
Total Number of Instances	7720	

Summary of the cross-validation results with confidenceFactor = 0.1.

KNN:

In the first meeting since the CSV files weren't working with Weka we have decided to work using Python. We found out that the state of art Scikit-Learn library could be useful for our purposes. Initially we tried to classify our data using Decision Tree's that can be found under the library, but we got around 25% accuracy from those, in cross-validation. Then we decided to use another classification method and we started experimenting a few other methods. One smart idea we came up with was finding scientific articles and research papers tackling the Fruit Recognition Problem in a similar manner. While researching we found one paper that was suggesting the use of KNN method (Kumar & Gill 2015). We decided to try this algorithm with Python Scikit-Learn library, we started with using 6 neighbours for the method. Then we decreased the number of neighbours from 6 to 1 since the accuracy was increasing as we decreased the number of neighbours. The results can be seen in the screenshot.



```
In [14]: accuracy
Out[14]:
array([ 0.63469558,  0.70009461,  0.67830688,  0.71108491,  0.74442988,
        0.77761836,  0.78412698,  0.79824561,  0.76235741,  0.82061856])
```

This is the accuracy results for every fold of a 10 fold cross-validation with KNN with 1 neighbour classifier using Python

Then we also decided to try the KNN classification on Weka which goes under the name of IBk. We right up started with using one neighbour since it was giving the best result when we tried it with Python.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      5715           74.0285 %
Incorrectly Classified Instances    2005           25.9715 %
Kappa statistic                     0.7394
Mean absolute error                  0.0009
Root mean squared error              0.0252
Relative absolute error              33.2923 %
Root relative squared error          69.2545 %
Total Number of Instances           7720
```

Results of 10 fold cross-validation with KNN method in Weka on the training set.

References

Kumar, A., & Gill, G. S. (2015). Computer vision based model for fruit sorting using K-nearest neighbour classifier. *Int. J. Electr. Electron. Eng.*, 2, 1694-2426.