FIT5047 – Intelligent Systems
Solutions to Tutorial on Supervised Machine Learning

Question 1: Introduction to weka and categorical attributes

1. Download and install WEKA. Please note that the Explorer and Tutorial files are somewhat out of date. Use also the `PACE_Bootcamp_TS2_WEKA_Intro` file.
2. Get WEKA started by clicking on the Explorer button, or by downloading an `arff` file and clicking on it.
3. Open the `weather.nominal.arff` dataset. Visualize the different variables, and postulate which variables are significant. Why?
   SOLUTION:
   The variables which discriminate between classes, e.g., `outlook`, are the most significant.
4. **J48** (which is algorithm C4.5)
   (a) Run `J48`, with X-validation, and analyze the resulting decision tree. Specifically, trace the computations and changes in class at each node down the different paths in the decision tree.
   (b) Calculate **manually** the Information Gain for `outlook` in level 1 of the tree. Compare this Information Gain with that obtained for `wind` in the lecture.
   SOLUTION:
   $IG(S, \text{outlook}) = H(S) - \frac{5}{14}H(S_{\text{sunny}}) - \frac{5}{14}H(S_{\text{rain}}) - \frac{4}{14}H(S_{\text{overcast}})$
   From the lecture: $H(S) = 0.94$
   $H(S_{\text{sunny}}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$
   $H(S_{\text{overcast}}) = 0$
   $H(S_{\text{rain}}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97$
   $IG(S, \text{outlook}) = 0.94 - \frac{5}{14} \times 0.97 - \frac{5}{14} \times 0.97 - \frac{4}{14} \times 0 = 0.94 - 0.693 = 0.247$
   (c) Copy `weather.nominal.arff` into your local user space, and try adding the following instances to the ARFF file (you can use `WordPad`):

   ```
   rainy,boiling,high,TRUE,yes
   hot,high,TRUE,yes
   ```

   What happens? How would you solve these problems?
   SOLUTION:
   The files cannot load because "boiling" is undefined in instance 1, and the outlook value is missing in instance 2.
   "boiling" must be added to the temperature attribute values, and and outlook value must be added to instance 2.
   (d) Now, remove these instances, and try adding the following instances to the ARFF file:

   ```
   overcast,mild,?,FALSE,yes
   rainy,mild,high,TRUE,yes
   rainy,hot,high,TRUE,?
   ```

   What happens? What is the difference in the resultant decision tree?
   SOLUTION:
   There is an extra value for temperature ("?") and there is a third class for play ("?").

The overall performance is unaffected, but in the new tree, the majority class is not clean.

5. Briefly explain the meaning of the summary measures produced by WEKA, i.e., Kappa statistic, Mean absolute error, Root mean squared error, Relative absolute error, Root relative squared error, TP Rate, FP Rate, Precision, Recall, F-measure and ROC.

   SOLUTION:

   - **Kappa statistic** – measures the pairwise agreement between the actual class and the predicted class.
   - **Mean absolute error** – measures how close predictions are to outcomes. The formula is:
     $$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - y_i|$$
     where p_i is the prediction for item i and y_i is the outcome. For two classes, p_i and y_1 are in the $\{0, 1\}$ range; for regression, they are actual values; and for more classes, the values depend on the meaning of the classes.
   - **Root mean square error** – similar to MAE, measures how close predictions are to outcomes. The formula is:
     $$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2}$$
   - **Relative absolute error** – this measure is computed by dividing the MAE by the MAE obtained by just predicting the mean of target values (and then multiplying by 100). Therefore, smaller values are better and values $> 100\%$ indicate a scheme is doing worse than just predicting the mean.
   - **Root relative squared error** – similar to RAE, but here we divide the RMSE by the RMSE obtained by just predicting the mean of target values (and then multiplying by 100).
   - **True positive rate (TPR)** – ratio of predicted positive divided by actual positive (for each class). For example, for class `a`, `J48`'s TPR is $5/9 = 0.556$.
   - **False positive rate (FPR)** – ratio of predicted positive divided by actual negative (for each class). For example, for class A, k-NN's FPR is $3/5 = 0.6$.
   - **Precision** – $\frac{\text{predicted correct}}{\text{predicted}}$.
   - **Recall** – $\frac{\text{predicted correct}}{\text{correct}}$.
   - **F-measure** – harmonic mean of Precision and Recall.
     F-measure $= \frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}}$
   - **Receiver-operator characteristic (ROC) area** – ROC is a graphical plot that illustrates the performance of a **binary** classifier. The curve is created by plotting the TP rate against the FP rate at various threshold settings. The ROC area is the area under the curve – an area of 1 represents a perfect classifier, and an area around 0.5 represents a useless classifier.
     Note: if the area under the curve is a lot under 0.5, then you can invert the result of the classifier, and obtain a good classifier.

6. **NaiveBayes**
   (a) Run `NaiveBayes` on the `weather.nominal.arff` dataset. Explain your results.
   (b) Calculate **manually** the probability of playing ball, given that the outlook is overcast, the temperature is hot, the humidity is normal, and the wind is weak.

   SOLUTION:

$$\begin{aligned}\Pr(\text{YES}|\text{overcast,hot,normal,weak}) &= \Pr(\text{overcast}|\text{YES}) \times \Pr(\text{hot}|\text{YES}) \times \\ &\quad \Pr(\text{normal}|\text{YES}) \times \Pr(\text{weak}|\text{YES}) \times \Pr(\text{YES})\end{aligned}$$

$\Pr(\text{YES}|\text{overcast,hot,normal,weak}) = \frac{4}{9} \times \frac{2}{9} \times \frac{6}{9} \times \frac{6}{9} \times \frac{9}{14} = \alpha \times 0.028$

$$\begin{aligned}\Pr(\text{NO}|\text{overcast,hot,normal,weak}) &= \Pr(\text{overcast}|\text{NO}) \times \Pr(\text{hot}|\text{NO}) \times \\ &\quad \Pr(\text{normal}|\text{NO}) \times \Pr(\text{weak}|\text{NO}) \times \Pr(\text{NO})\end{aligned}$$

$\Pr(\text{NO}|\text{overcast,hot,normal,weak}) = \frac{0}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} \times \frac{5}{14} = \alpha \times 0$
$\Pr(\text{YES}|\text{overcast,hot,normal,weak}) = 1$ and $\Pr(\text{NO}|\text{overcast,hot,normal,weak}) = 0$.

7. **IBk** (which is k-NN)
   (a) Run `IBk` on the `weather.nominal.arff` dataset. Explain your results.
   (b) Use the Jaccard coefficient to calculate the similarity between the following two data records.

   ```
   sunny, hot, high, FALSE, no
   sunny, mild, normal, FALSE, yes
   ```

   SOLUTION:

$$Jc = \frac{|\{sunny, FALSE\}|}{|\{sunny, hot, mild, high, normal, no, yes, FALSE\}|} = \frac{2}{8} = 0.25$$

8. Compare the performance of `J48` with that of `NaiveBayes` and `IBk`.

   SOLUTION:
   Accuracy IBk: 57.1429%
   Accuracy NaiveBayes: 57.1429%
   Accuracy J48: 50.00%

   `NaiveBayes` performs better than `J48` and equals `IBk`.

Question 2: Continuous attributes

1. Copy `weather.arff` into your local user space, and try running `J48` on it. What happens?

   SOLUTION:
   Accuracy `J48`: 64.2857%, which is much better than the previous results. The difference in performance might be atributed to the numerical features, which arguably contain more information.

2. Now remove the numeric attributes using the supervised `AttributeSelection` filter, which you will find by clicking `Choose` under the `Filter` heading in the `Preprocess` tab. Analyze your results.

   SOLUTION:
   The accuracy is reduced to 42.8571%, the tree has a depth of 2. As we expect, the performance decreases even further under this scheme. We can argue that, with Machine

Learning in general, and Decision Trees in particular, when we have less informative features, the learning performance will go down and vice versa.

3. Run `J48` on `weather.arff`, and analyze the resultant decision tree and summary values.
   SOLUTION:
   Accuracy `J48`: 64.2857%, with precision, recall and F-measure arround 0.63. Although this is a better result compared to the previous one, it is not a particularly strong result. The reason for this might be that we train/test our models with a small ammount of data. With only a few training examples, our models have less statistics to learn the decision from.

   Another reason is that, with a small test set, a single correct/wrong decision can change the accuracy drastically. Thus, there might be a chance that the two models might be mostly comparable (with a large test set), while reporting very different accuracies on a small test set.

4. Run `J48`, `NaiveBayes` and `IBk` on `weather.arff`, and analyze the resultant summary values. Compare the performance of the three algorithms, and also compare the performance of each algorithm with its performance on the `weather.nominal.arff` dataset.
   SOLUTION:
   Accuracy `J48`: 64.2857%
   Accuracy `NaiveBayes`: 64.2857%
   Accuracy `IBk`: 78.5714%
   IBk performs much better in this scenario compared to the previous one, while NaiveBayes and J48 see smaller increases. In this case, the IBk might have better gain from the numerical features. However, we do not know this for sure, as our train/test set is too small, and as we have mentioned above, a single correct/wrong instance can make a huge difference.