CS36110 Assignment: Employee Absenteeism

Ben Jaeger | bej29

Contents

[1. Initial Building of Machine Learning Models 3](#_Toc23417331)

[1.1. Initial Classifier Creation and Explanation 3](#_Toc23417332)

[1.2. Comparing Results between Classifiers 3](#_Toc23417333)

[References 3](#_Toc23417334)

# 1. Initial Building of Machine Learning Models

## 1.1. Initial Classifier Creation and Explanation

For this assignment I used a J48 and NaiveBayes classifier as prescribed in the assignment, however I also used a RandomForest classifier as I had a passing knowledge of it from previous machine learning work.

J48 is an implementation of a decision tree-based classifier based on the C4.5 release 8 classifier (hence the name). It works by finding the entropy within a dataset, this defines how heterogeneous the dataset is, seemingly unrelated the features are to the decision feature, then it defines the gain that would be made by dividing the dataset across a feature, the gain being a measure of the reduction in entropy. The aim being to greedily reduce entropy, and thus increase the homogeneity at each level. One advantage of J48 is that it is a very simple model to both produce and, arguably more importantly in some situations, a explain to someone who has little or no knowledge of machine learning.

NaiveBayes is a probabilistic classifier, meaning it uses the given dataset to find the probability of an instance belonging to a particular decision feature group. This is done by trying to calculate how likely the outcome is given the features (“how likely is it a banana given it’s long, yellow and sweet”), this is achieved by finding how likely that outcome is in the first place (“how likely is it that this is a banana”) the logic being that less likely outcomes should be prejudiced against and also how likely each of those features is given that outcome (“if this were a banana, how likely is it that its long yellow and sweet”) this aims to prejudice against unlikely combinations resulting in that outcome. NaiveBayes has the advantage of disregarding the interactions between features (hence why its naive) this reduces the calculations needed greatly. It also easily adapts to more data as there is nothing to restructure such as a decision tree.

The last classifier is a RandomForest classifier as I had encountered it before. A RandomForest is an example of an ensemble learning model, meaning that it is a model made up of many smaller models, attempting to negate the negatives of each. The classifier takes many decision trees formed from subsets of the dataset (with replacement) and then “votes” between them, the idea being that this will reduce the chance of the tree over fitting to the data as just one decision tree, such as J48, is sometimes susceptible to doing. The difference between a normal decision tree and those that make up this model is that at each level the tree randomly pick from a subset of the features remaining, when it comes to voting the majority rules at each level for classification (Liaw, 2002).

## 1.2. Comparing Results between Classifiers

When comparing machine learning models there are many ways to measure and compare them. We will be looking at the F1 scores as well as the ROC area

A useful way to compare classifiers is to look at their F1 scores, this is the harmonic mean between the precision (the number of true positives divided by all predicted positives, how often the model is correct when it thinks it is) and the recall (the number of true positives divided by actual positives, how many positives the model misses) (Sasaki, 2007). Due to the nature of the precision and recall there is often a trade off between them, a “wider net” increases recall but reduces precision and vice versa, therefore by considering both we can very effectively evaluate a model’s accuracy. The NaiveBayes classifier had an F1 score of only 0.549 whereas the J48 and RandomForest did much better at 0.718 and 0.788 respectively. Some criticism is levelled against the F1 score as it gives equal precedence to recall and precision, but you may not always have a case where this is applicable.

Another useful metric is the ROC area, this compares the model to a random coin flip by graphing the True Positive Rate (TPR, aka the recall as mentioned before) and False Positive Rate (FPR is the false positive prediction divided by all the actual negative predictions, it measures the positives given where a negative was correct) against each other at various levels of threshold, the ROC area being the area beneath the curve as plotted, 0.5 indicating a classifier being no better than a random guess and 1 being perfect classification. The NaiveBayes classifier again comes last with a ROC area of 0.782, J48 in second with 0.854 and RandomForest wins with 0.934, due to the various uses of machine learning there’s no absolute rule about what’s a good ROC area to get however RandomForest’s 0.934 is undeniably a great result, both in comparison to the other two and with a goal that a ROC area of 1 indicates a perfect classifier.

# References

Liaw, A. &. (2002). Classification and regression by randomForest. *R news, 2(3)*, 18-22.

Sasaki, Y. (2007). The truth of the F-measure. *Teach Tutor mater, 1(5)*, 1-5.