Trevor Koon  
CS699  
Data Mining project  
11/16/2017

Employee Attrition Classifier



**Data mining objective:**

The main objective of this data mining task is to predict employee attrition. Given the set of attributes within the data set are we able to classify an employee who will leave the company (Attrition = Yes.)? Several attribute selection and classification algorithms will be used in combination. The combination that produces the best evaluation metrics such as Accuracy or area under the ROC curve will be chosen as the future classification module.

**Description of the Data:**

The data set initially contained 35 attributes and 1470 instances. A brief description of each attribute follows:  
  
*Age*: Numeric value, age of the employee

*Attrition*: Nominal Yes or No values, indicates the employee has left the company if Yes. This attribute is our class label and what we are trying to predict.

*BusinessTravel*: Nominal ‘Travel\_Rarely, Travel\_Frequently, Non-Travel’ values. Describes how often the employee travels.

*DailyRate*:Numeric: value providing daily income rate for the employee

*Department*: Nominal value ‘Sales, Research & Development, Human Resources’ providing which department the employee works in

*DistanceFromHome*: Numeric value describing the distance the employee must travel to/from work. The larger the value the further away the employee lives.

*Education*: Nominal values include ‘1:Below College, 2:Some College, 3: Bachelor, 4:Master, 5:Doctor’

*EducationField*: Nominal values include: ‘1: Life Sciences, 2: Other, 3: Medical, 4: Marketing, 5: Technical Degree, 6: Human Resources’

*EmployeeCount*: A value of 1 counts the employee tuple (this attribute is removed during preprocessing).

*EmployeeNumber*: An identifier associated with a unique employee (this attribute is removed during preprocessing).

*EnvironmentSatisfaction*: Nominal with values (1: Low, 2: Medium, 3: High, 4: Very High) describes how satisfied the employee is with their work environment.

Gender: Nominal values are either Male or Female.

*HourlyRate*: Numeric hourly payrate for employee

*JobInvolvement*: Nominal with values (1: Low, 2: Medium, 3: High, 4: Very High)

*JobLevel*: Nominal values Nominal with values 1 through 5. The higher the value the more experience/responsibility the employee has.

*JobRole*: Nominal, the job roles are ‘Sales Executive, Research Scientist, Laboratory Technician, Manufacturing Director, Healthcare Representative, Manager, Sales Representative, Research Director, Human Resources.’

*JobSatisfaction*: Nominal with values (1: Low, 2: Medium, 3: High, 4: Very High) describing how the satisfied the employee is with their job.

*MaritalStatus*: Nominal with values (‘Single, Married, Divorced’)

*MonthlyIncome*: Numeric, monthly income of the employee.

*MonthlyRate*: Numeric monthly pay rate of the employee.

*NumCompaniesWorked*: Numeric details how many companies the company has worked for in their career.

*Over18*: Nominal, binary Yes or No signifying the employee is over 18 years of age. This attribute is removed during preprocessing.

*OverTime*: Nominal, binary Yes or No, states whether the employee has worked overtime or not.

*PercentSalaryHike*: Numeric value indicating the employees last raise as a percentage of their salary.

*PerformanceRating*: Numeric Performance rating given in last review.

*RelationshipSatisfaction*: Nominal with values (1: Low, 2: Medium, 3: High, 4: Very High) describing how satisfied the employee is with their relationships at work.

*StandardHours*: Numeric number of hours the employee works. This attribute is removed in preprocessing.

*StockOptionLevel*: Numeric describing how much equity an employee was given in compensation.

*TotalWorkingYears*: Number of years the employee has been in the work force.

*TrainingTimesLastYear*: Numeric how much training the employee received last year

*WorkLifeBalance*: Nominal with values (1: Low, 2: Medium, 3: High, 4: Very High) describes the work life balance the role offers the employee

*YearsAtCompany*: Numeric, number of years the employee has been at the compnay

*YearsInCurrentRole*: Numeric, number of years the employee has been in their current role.

*YearsSinceLastPromotion*: Numeric, number of years since the employee’s last promotion

*YearsWithCurrManager*: Numeric, number of years the employee has worked with their current manager.

**Procedure:**

**Data exploration & preprocessing:**

The data mining task began by converting the raw CSV data into an .ARFF file by importing the file into the WEKA datamining tool. Once the data had been imported the next step was to correctly change attributes that had been imported as numeric values but should counted as nominal ones. Attributes such as EducationLevel despite having the values 1 through 5 should be nominal since each number represents a category such as the value 5 representing employees with a doctoral degree. The WEKA unsupervised NumericToBinary filter aided the conversion of these variables.

Once all the nominal variables were converted from numeric types all attributes were reexamined for inclusion into the final data set. Several attributes such as EmployeeCount and EmployeeID counted and identified tuples, these attributes were removed. There was some redundancy between HourlyRate, DailyRate, MonthlyRate, and MonthlyIncome variables. These variables all detail the employee’s salary and only MonthlyIncome was included in the final data set. The attributes Over18 and StandardHours had the same value for all tuples e.g. every employee in the data set was over 18, thus these were removed.

There were no missing values in the data set. All original tuples were used for training and testing the selected classifier algorithms.

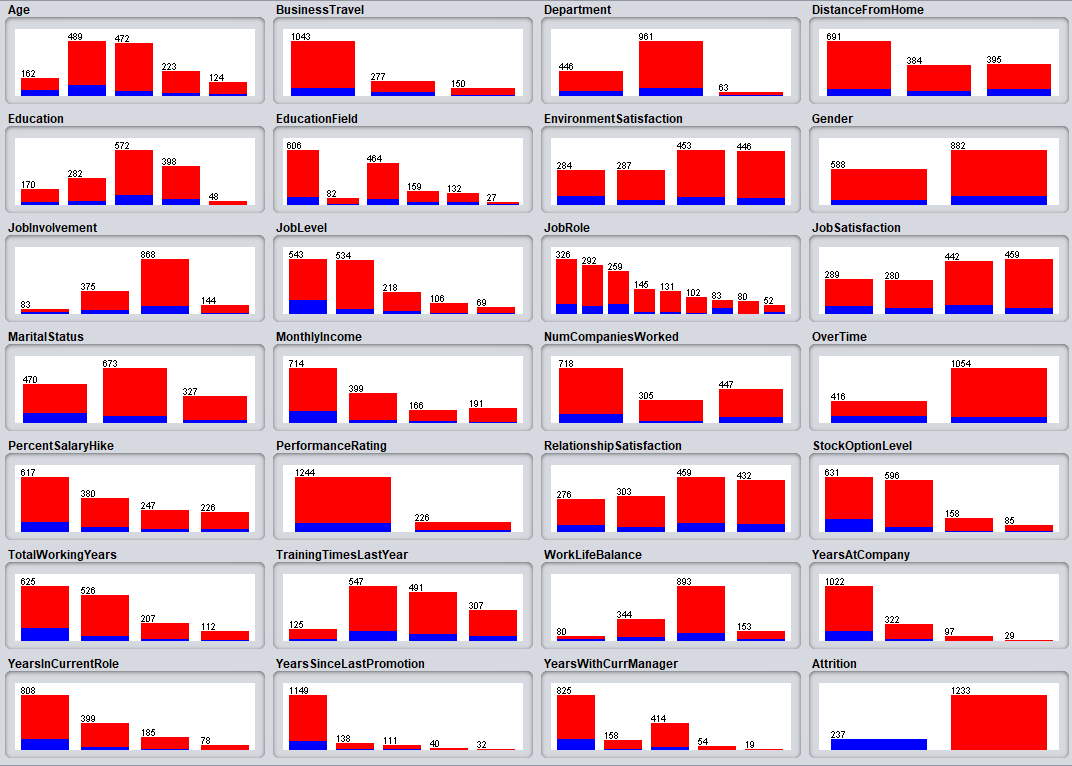


Figure 1 Attribute distributions Red = No Attrition, Blue = Yes Attrition

**Description of chosen Attribute Selection Algorithms**

Custom Attribute Selection;   
My chosen list of attributes based on intuition to what may affect an employee’s choice to leave their employer.

CFSSubsetEval:  
‘Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.  
  
Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.’

GainRatio: ‘Evaluates the worth of an attribute by measuring the gain ratio with respect to the class.’

CorrelationAttributeEval:   
‘Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class.  
  
Nominal attributes are considered on a value by value basis by treating each value as an indicator. An overall correlation for a nominal attribute is arrived at via a weighted average.’

OneR AttributeEval:‘Evaluates the worth of an attribute by using the OneR classifier.’

**Attribute Subset Results:**  
Several selection algorithms provided a ranked list of attributes with their associated score e.g. Correlation or GainRatio in this case the top 12 attributes returned were selected for use in the classification algorithm.

Custom List:

JobSatisifaction, MonthlyIncome, Age, YearsSinceLastPromotion, RelationshipSatisfaction, StockOptionLevel, JobLevel, JobRole, Education, DistanceFromHome, WorkLifeBalance, YearsAtCompany

CFSSubsetEval:

Age, BusinessTravel, EnvironmentSatisfaction, JobInvolvement, JobLevel, JobRole, JobSatisfaction, MaritalStatus, OverTime, StockOptionLevel, WorkLifeBalance, YearsInCurrentRole, Attrition

GainRatio:

OverTime, StockOptionLevel, JobLevel, JobRole, MaritalStatus, Age, MonthlyIncome, TotalWorkingYears, YearsInCurrentRole, BusinessTravel, YearsWithCurrManager, JobInvolvement, Attrition

Correlation:

OverTime, StockOptionLevel, JobLevel, TotalWorkingYears, MaritalStatus, MonthlyIncome, YearsInCurrentRole, YearsAtCompany, YearsWithCurrManager, Age, Department, BusinessTravel, Attrition

OneR

YearsWithCurrManager, EnvironmentSatisfaction, JobInvolvement, JobLevel, JobRole, Gender, EducationField, MaritalStatus, Education, BusinessTravel, Department, DistanceFromHome, Attrition

**Description of chosen Classification Algorithms:**

J48:A classifier that builds a decision tree splitting attributes

based on information gain.

Naïve Bayes: Uses Bayes Theorem to calculate probability of a tuple belonging to each class and selects the class with the highest probability. Attributes are assumed to be independent from one another.

Logistic Regression: Fits a model to predict a binary or multiclass dependent variable. Classification depends on the probability of the dependent variable given the known independent variables.

Multilayer Perceptron: Trains a neural network with back propagation to then predict unknown class attributes.

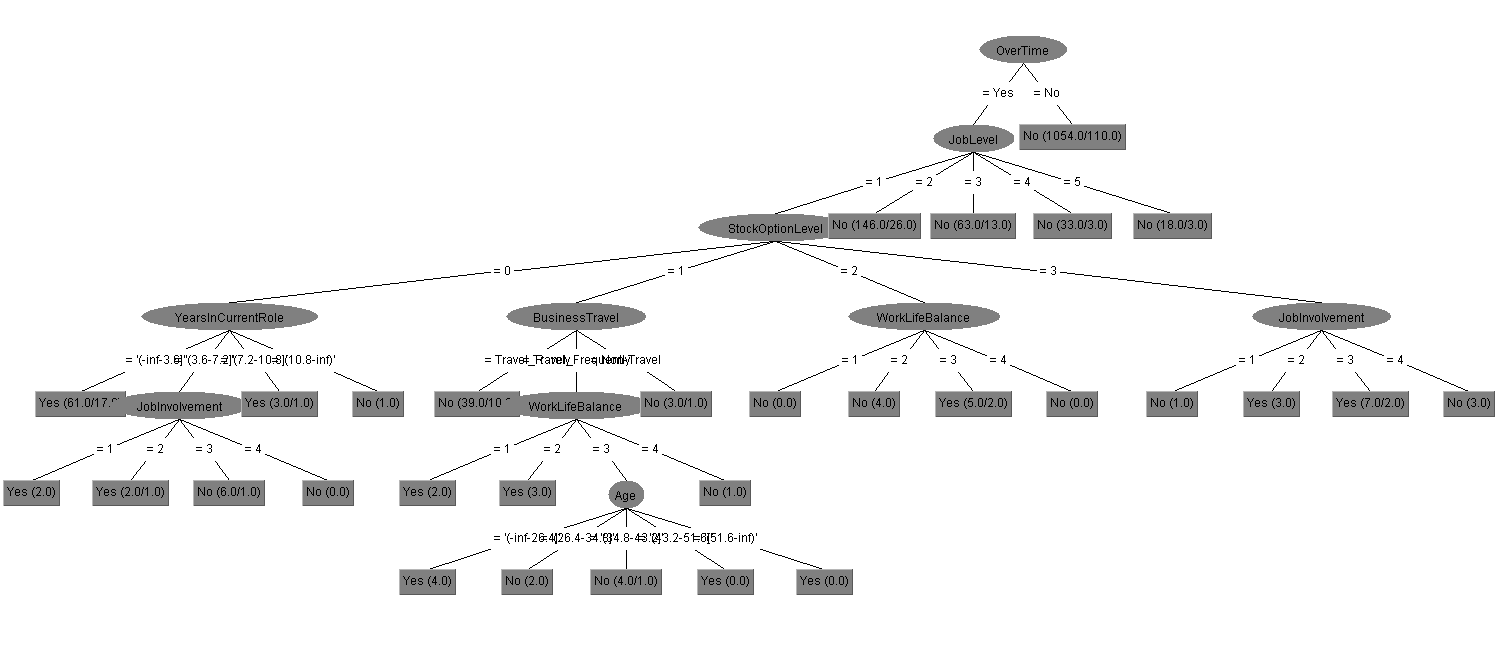


Figure 2Decision Tree built from CFSSubset eval attributes

**Classification results:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Custom Attribute Selection | | | | | |  | |  | |
|  | | **J48** | | **Naïve Bayes** | | **Logistic** | | **Neural Net** | |
| Yes TP Rate: | | 0 | | 0.414 | | 0.177 | | 0.257 | |
| Yes FP Rate: | | 0 | | 0.105 | | 0.033 | | 0.101 | |
| No TP Rate: | | 1 | | 0.895 | | 0.967 | | 0.899 | |
| No FP Rate: | | 1 | | 0.586 | | 0.823 | | 0.743 | |
|  | |  | |  | |  | |  | |
| Accuracy | | 83.88 | | 81.768 | | 83.945 | | 79.5918 | |
| ROC | | 0.495 | | 0.734 | | 0.755 | | 0.683 | |
|  | |  | |  | |  | |  | |
| CFSSubSetEval | | | |  | |  | |  | |
|  | | **J48** | | **Naïve Bayes** | | **Logistic** | | **Neural Net** | |
| Yes TP Rate: | | 0.263 | | 0.439 | | 0.376 | | 0.456 | |
| Yes FP Rate: | | 0.032 | | 0.066 | | 0.041 | | 0.069 | |
| No TP Rate: | | 0.968 | | 0.934 | | 0.959 | | 0.931 | |
| No FP Rate: | | 0.764 | | 0.561 | | 0.624 | | 0.544 | |
|  | |  | |  | |  | |  | |
| Accuracy | | 84.97 | | 85.442 | | 86.5306 | | 85.442 | |
| ROC | | 0.624 | | 0.808 | | 0.836 | | 0.783 | |
|  | |  | |  | |  | |  | |
| GainRatioEval | |  | |  | |  | |  | |
|  | | **J48** | | **Naïve Bayes** | | **Logistic** | | **Neural Net** | |
| Yes TP Rate: | | 0.241 | | 0.519 | | 0.287 | | 0.283 | |
| Yes FP Rate: | | 0.029 | | 0.132 | | 0.032 | | 0.108 | |
| No TP Rate: | | 0.971 | | 0.868 | | 0.968 | | 0.892 | |
| No FP Rate: | | 0.759 | | 0.481 | | 0.713 | | 0.717 | |
|  | |  | |  | |  | |  | |
| Accuracy | | 85.31 | | 81.1565 | | 85.8503 | | 79.3878 | |
| ROC | | 0.639 | | 0.754 | | 0.794 | | 0.683 | |
| CorrelationAttributeEval | | | | |  | |  | |
|  | **J48** | | **Naïve Bayes** | | **Logistic** | | **Neural Net** | |
| Yes TP Rate: | 0.228 | | 0.536 | | 0.338 | | 0.367 | |
| Yes FP Rate: | 0.03 | | 0.152 | | 0.033 | | 0.105 | |
| No TP Rate: | 0.97 | | 0.848 | | 0.967 | | 0.895 | |
| No FP Rate: | 0.772 | | 0.464 | | 0.662 | | 0.633 | |
|  |  | |  | |  | |  | |
| Accuracy | 85.03 | | 79.79 | | 86.53 | | 81.02 | |
| ROC | 0.634 | | 0.765 | | 0.807 | | 0.736 | |
|  |  | |  | |  | |  | |
| OneR |  | |  | |  | |  | |
|  | **J48** | | **Naïve Bayes** | | **Logistic** | | **Neural Net** | |
| Yes TP Rate: | 0 | | 0.245 | | 0.207 | | 0.295 | |
| Yes FP Rate: | 0 | | 0.055 | | 0.028 | | 0.114 | |
| No TP Rate: | 1 | | 0.945 | | 0.972 | | 0.886 | |
| No FP Rate: | 1 | | 0.755 | | 0.793 | | 0.705 | |
|  |  | |  | |  | |  | |
| Accuracy | 83.88 | | 83.197 | | 84.8299 | | 79.047 | |
| ROC | 0.495 | | 0.759 | | 0.768 | | 0.701 | |

**Classification results:**

Quickly we see that True Positive rates for predicting ‘Yes’ tuples where an employee has left their company is a challenge for these classifiers. This is due to a class imbalance problem where the majority of the tuples or about 80% are ‘No’. This is causing our models to overfit and be more likely to predict‘No’ attrition. The result is generally good accuracy among all models with high ROC scores but performance in identifying positive cases. Since an HR manager would really be more interested in having a good estimate of which employee is likely to leave so they can respond e.g. by offering a raise or some other benefit.

One method of correcting the class imbalance problem is to oversample positive tuples to better train the classifiers in identifying positive cases. Undersampling performs a similar function but it undersamples negative tuples.

To see which model(s) performed the best we can look at the below metrics

*Most accurate*: All models produced fairly results in total number of correct predictions.   
The best result in this measure when Logistic regression was used with the attributes derived from the Correlation and CFSSubsetEval algorithms.

*Best ROC Area under curve*: The area under the curve tells us how well the model is producing True Positives vs False Positives using probabilities thresholds to classify a True Positive. A score of 0.5 or lower would be no better than randomly guessing.

In this category Logistic regression again scored the highest within all attribute groups. The CFSSubSet Eval attribute group had the highest area with 0.836. The lowest performing algorithm was J48. Within my custom attribute list and the OneR list the area was only .495 meaning that you could do just as well by randomly guessing whether a tuple was ‘Yes’ or ‘No’.

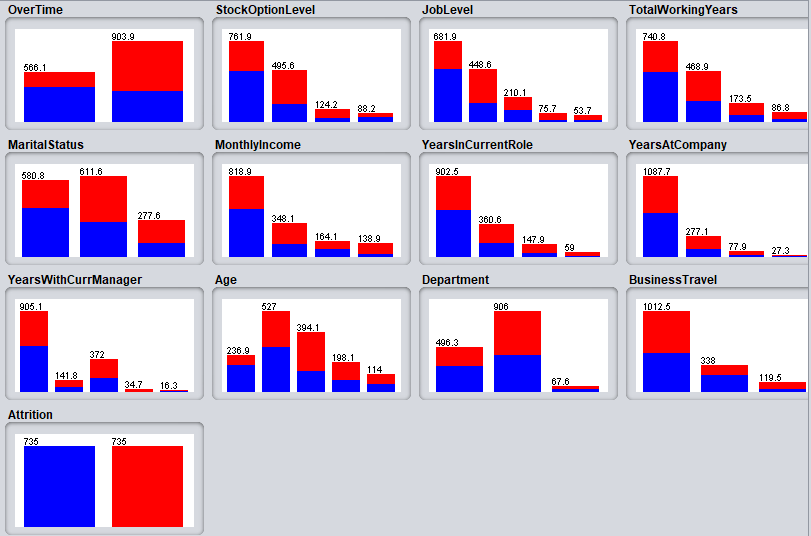
*Best ‘Yes’ True Positive Rate*: Perhaps the most important metric we are measuring for our models to be useful is their ability to identify True Positives. Naïve Bayes was the clear winner in this category. The CorrelationAttributeEval attribute group had a TP Rate of 0.536 meaning about 1 of every 2 actual ‘Yes’ tuples were correctly classified. This rate is not ideal but the model could still be useful given the domain where we are applying data mining. If this were a test to classify whether a patient had cancer then we would want to have a much higher TP rate for detecting the disease since if left undiagnosed it would have greater consequences.

False Positive rate is also worth looking at, Naïve Bayes also had higher false positive rates and thus lower ROC scores. This is because the model was more likely than the others to classify something as Positive vs Negative. The rate was still relatively low at about ~15% within the correlation attribute group.

It is also clear that my custom attribute selection and the OneR based attribute selection group performed the worst. Accuracy was comparable to the other attribute groups but TP rates suffered and when using J48 my selected attributes and OneR’s did not classify a single positive Tuple.

**Fixing the class imbalance problem:**

In an attempt to improve the True Positive rate for classifying ‘Yes’ tuples to make the model more realistic Weka’s ‘ClassBalance Filter was applied to the Correlation attribute subset. The four classifiers were then reran and the results are discussed below.



With the resampling done from the ClassBalance filter the four classifier algorithms were rerun.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CorrelationAttributeEval - Class Balance filter | | | |  |
|  | **J48** | **Naïve Bayes** | **Logistic** | **Neural Net** |
| Yes TP Rate: | 0.654 | 0.709 | 0.713 | 0.422 |
| Yes FP Rate: | 0.295 | 0.338 | 0.269 | 0.176 |
| No TP Rate: | 0.705 | 0.662 | 0.731 | 0.824 |
| No FP Rate: | 0.346 | 0.291 | 0.278 | 0.578 |
|  |  |  |  |  |
| Accuracy | 67.94 | 68.5331 | 72.19 | 62.3 |
| ROC | 0.709 | 0.752 | 0.791 | 0.705 |

Right away the ‘Yes’ TP rates are much higher than anything in first run of the model.   
Accuracy does fall and it is no longer guaranteed that the classifier has a high ‘No’ TP rate because the class instances of both ‘Yes’ and ‘No’ tuples are equal. It is interesting to see that Naïve Bayes which was the clear best choice in the previous run for this set up of attributes is no longer the obvious option. Logistic regression outperforms the others by having the highest ‘Yes’ TP rate at 71% while maintaining a low FP rate. The overall accuracy is also the highest for Logistic Regression. If you wanted to do a bit more tuning with the sampling of tuples to get the ideal balance of overall accuracy and sensitivity you could. It does not need to be a 50/50 split between ‘Yes’ and ‘No’.

This model is much more useful as there is more confidence in identifying employees who are at risk to leave the company.

**Conclusion & Final Thoughts:**

There have been a lot of lessons learned throughout completing the data mining task. It was interesting to see how much a skewed class ratio as was the case here can affect how the classifiers function. There are a lot of practical problems where the positive class we want to identify is relatively rare so we would need to deal with the class imbalance in a similar manner to this project. Examples such as detecting fraud, disease, loan defaults are all cases where the event we are attempting to predict is relatively rare.

It was interesting to see how the different attribute selection algorithms compared to each other as well as the different classification algorithms. My own selection of attributes did not fair nearly as well as the computed group (asides from OneR). The attributes that were selected via algorithms seemed to be intuitive indicators to lead an employee to leave but may not be my first picks. I was surprised how long multilayer perceptron neural networks took to train and build but then did an average job in this task at classifying while Logistic regression and Naïve Bayes were efficient to build and performed well.

**Tools used:**  
Waikato Environment for Knowledge Analysis (Weka)

Microsoft Excel

**References:**  
http://weka.sourceforge.net/doc.stable/