173705
172808

Business School Rank Prediction

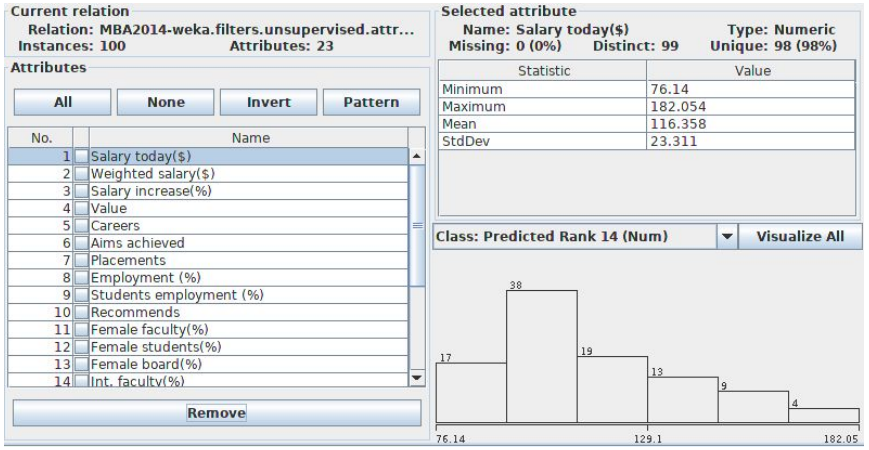
Introduction to the problem

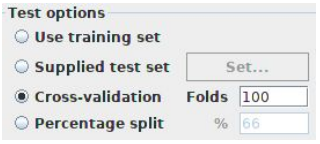
After receiving the Financial Times dataset for MBA Rankings, we have to apply some little corrections to the data in order to analyze it with Weka's tool. This basic corrections consist on format all the text of our dataset as well as separating the employment percentages into a correct CSV file.

Once load it, we can proceed to apply all those machine learnings methods in which the regression methodology is applicable, as far as we know. We have selected the following algorithms:

- Multilinear Regression
- k-Nearest Neighbors
- Support Vector Machines
- Neural Networks

The criterion we will follow in order to do the evaluation will be based on the Correlation Coefficient combined with leave one out cross validation. The leave one out behaviour can be emulated with Weka by setting the number of folds equal to the number of instances of our dataset. This will be the setup of the analysis:





Note that the dataset consists on 100 instances on Current relation label and that we are using 23 attributes. The number of folds is adjusted to 100 to do the leave one out validation. We have eliminated both the first rank column and School's name in order to not add noise to the data.

Correlation Coefficient Comparison

For your understanding, we have included a brief explanation of all the algorithms we have used as well as a table which collects all the relevant results after trying every model.

Multilinear Regression [Functions → Linear Regression]

The strategy followed in linear regression consists on generate a plane in order to fit as well as possible the training data. As we can have seen in Weka, it is an algorithm really fast to train and the accuracy is quite good which make us think that the input attributes are a good linear combination with each other.

k-Nearest Neighbors [Lazy → IBk]

The strategy followed in k-Nearest neighbors, when applied in regression, consists on taking the mean of the k most similar instances through the training dataset. As we can observe in the results, it has the worst accuracy based on the correlation coefficient since the instances are little different between them.

Support Vector Machines [Functions → SMOReg]

The regression version of SVM works by finding which is the line that fits at same time that minimizes the error of the cost function. For that, the algorithm only takes into account those instance which are closer to the line with minimum cost (support vectors).

Neural Networks [Functions → Multilayer Perceptron]

The neural networks methodology consists on approximate the underlying function in order to take the best decision while discriminating classes, so the main objective in the regression case will be approximate a function that fits better the real prediction value.

Algorithm	Build Model Time	Training Speed	MAE	RMSE	RAE	RRSE	Correlation Coeff.
Linear	0s	V. Fast	9.2814	11.395	36.7984%	39.1142%	0.9193
k-NN	0s	V. Fast	15.88	20.548	62.9604%	70.5324%	0.7296
SVM	0.09s	Medium	10.1095	12.3187	40.0816%	42.2847%	0.9063
NN	0.3s	Slow	5.621	7.4528	22.2859%	25.8911%	0.9653

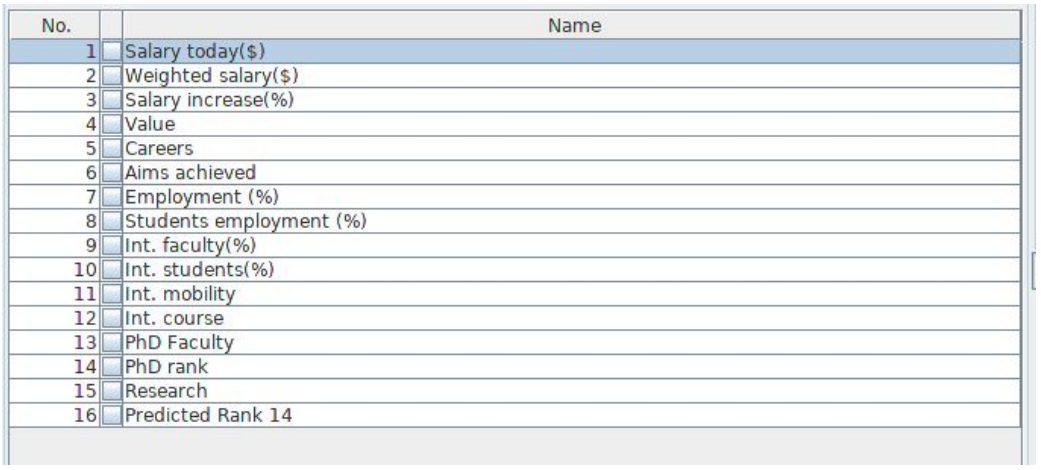
Since we are doing the evaluation based on the correlation coefficient, we expect values between -1 and 1, and having in mind the above table, we can say that all the accuracies can be considered quite good, since the closer we are to 1, the better the data is fitted. In the first instance, we can say that the first k-NN algorithm is the worst candidate we could select, so in this part of the analysis, we will focus on Linear, SVM and NN algorithms. With the Mean Average Error, we can see how good are the Neural Network model since the difference between the predictions and the true values is little, which is an indicator of how good the algorithm works. The values for this fact in both regression and SVM are not as good as for NN. The same case occurs with RMSE, but in this case we have to think that all the errors are squared, so this is an indicator more reliable in case of having large errors. Both RAE and RRSE are percentually better in the NN algorithm. Despite these values, the training time of the Neural Network could increase substantially as the number of dataset values goes up. So, we

think that the values that the linear regression method offers are quite good and having in mind all these considerations, at this point of the study we will go for a multilinear regression model.

Improvements for the models

The difference between the previous algorithms stands out just by looking the table above, but just to not be hurried, we will try to make some improvements to the previous algorithms before taking a final decision.

In order to try to improve our models, we have made a feature selection in which we have removed all those attributes which have less weight when computing the final ranking score, according to the statement's document, so we have removed 7 attributes and the results have been the following:



No.	Name
1	Salary today($)
2	Weighted salary($)
3	Salary increase(%)
4	Value
5	Careers
6	Aims achieved
7	Employment (%)
8	Students employment (%)
9	Int. faculty(%)
10	Int. students(%)
11	Int. mobility
12	Int. course
13	PhD Faculty
14	PhD rank
15	Research
16	Predicted Rank 14

Algorithm	MAE	RMSE	RAE	RRSE	Correlation Coeff.
Linear	9.0462	11.0565	35.8661%	37.9523%	0.9239
k-NN	16.17	21.4642	64.1101%	73.6771%	0.712
SVM	9.4607	11.4476	37.5096%	39.2947%	0.9181
NN	9.4133	12.9652	37.3213 %	44.504 %	0.9106

As we can observe on the table, we have improved slightly both Linear and SVM models by removing some irrelevant attirbutes. As we can observe, the accuracy of k-NN and NN have decreased. The explanation for k-NN is that the data is not normalized, and what's more, precisely the data with higher contribution (salary type) is that one which differs the more. In the case of Neural Networks, the lesser values the algorithm has to train, the worst results, so a neural network is known to work better as the number of samples is higher. By having this two tables in front, we think that both SVM and Linear are algorithms with more robustness and as we expected previously, we think that the linear regression model would be the best candidate for this case of studio.