HR Analytics: Job Change of Data Scientists

Ammar Lakho 18055

Ayesha Atique 17927

Zunaira Mairaj 17631

TABLE OF CONTENTS

Problem Description

An organization desperately wants to expand their Data Science department. When the candidates are selected, one of the important factors that needs to be taken into consideration is that they are able to carry out their job duties in accordance to the expectations of their employers. For this purpose, the organization is willing to recruit capable individuals who currently do not have advanced training or experience in this specific area since they want people working for them on an urgent basis. Candidates will go through comprehensive Data Science courses, and they will be offered a job once they successfully complete these courses. However, it is up to the candidates to choose whether they wish to work with this organization or to move to another company once they have undergone all the required courses.

To anticipate that, a model has been built that predicts if a person is likely to work for the same organization from which they have completed their training or not. The model will learn from the history of candidates who've gone through the courses for the organization.

The model can help the organization in the future because the organization would like to spend their resources on a candidate that is likely to work for them. So they will use our model to predict whether a candidate, who has just signed up for their courses, will work for them in the future or not and then only accept the student if the model tells them to. This will also reduce the manpower required in selecting only the appropriate candidates for the organization.

Data has been collected about the candidates by making them fill a sign up form. The data collected is described in more detail in the Data Description section.

Data Description

The data collected consists of 19158 rows and 14 columns. Each row contains information about a candidate who has passed the courses in the past.

Attributes

1. enrollee_id: Unique ID given to each candidate.
2. city: Code of the city that the candidate belongs to. There are 123 unique values,

implying candidates have been from 123 cities.

3. city_development_index: Numeric attribute. An economic metric which measures the level of development in a city.
4. gender: A categorical variable with 3 possible values: ‘Male’, ‘Female’ and ‘Other’.
5. relevant_experience: A binary attribute which has only 2 possible values: ‘Has relevant experience” and “no relevant experience”.
6. enrolled_university: A categorical attribute with 3 possible values depending on the candidate’s current educational situation: ‘no_enrollment’, ‘Part time course’ and ‘Full time course’.
7. education_level: A categorical attribute with 5 possible values depending on the candidate’s highest education level: ‘Primary School’, ‘High School’, ‘Graduate’, ‘Masters’ and ‘Phd’.
8. major_discipline: A categorical attribute with 6 possible values depending on the major that the candidate has graduated in: ‘STEM’, ‘Humanities’, ‘Business Degree’, ‘Arts’, ‘No Major’ and ‘Other’.
9. experience: A categorical attribute with 21 possible values depending on the amount of years of professional experience the candidate has: <1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, >21.
10. company_size: A categorical attribute with 8 possible values depending on the size of the company that the candidate is currently working in: ‘<10’, ‘10-49’, ‘50-99’, ‘100-500’, ‘500-999’, ‘1000-4999’, ‘5000-9999’.
11. company_type: A categorical attribute with 6 possible values depending on the type of the company that the candidate is currently working in: ‘Pvt Ltd’, ‘Early Stage Startup’, ‘Funded Startup, ‘Public Sector’, ‘NGO’, ‘Other’.
12. last_new_job: : A categorical attribute with 6 possible values depending on the number of years between the candidate’s current job and previous job: ‘never’, ‘1’, ‘2’, ‘3’, ‘4’, ‘>4’.
13. training_hours: Numeric attribute. Number of training hours that a candidate has completed for the courses.
14. target: Binary attribute with 2 possible values: 0 and 1. 0 means that the candidate does not accept the job offer after training. 1 means that the candidate accepts the job offer after training. This is the class label which needs to be predicted for the company to make better decisions in the future.

Data PreProcessing

Column Filtering

1. enrollee_id: This is filtered for obvious reasons.
2. City: There are 123 unique city values. Each city has a city_development_index(CDI) value. The CDI is dependent on the city code since the same city code would also mean the same CDI. So we should drop one of the 2 columns. City is a nominal categorical attribute where the code doesn't have a specific meaning. On the other hand, CDI is a numeric attribute and could help us analyze our results in an economic context.

Handling Missing Values

Column	%	Replaced With
gender	23%	No reason to assume the candidate's gender so missing values were replaced with 'Other' (existing category for this column).
enrolled_university	2%	High likelihood that candidates who aren't currently enrolled in a university left this blank so missing values were replaced with 'no_enrollment' (existing category for this column).
education_level	2%	No reason to assume why any candidate left this blank so missing values are replaced with 'Other'(new category for this column).
major_discipline	14%	High likelihood that people who had No Major left this blank so missing values are replaced with 'No Major' (existing category for this column).
experience	0.3%	High likelihood that people who had no experience left this blank so missing values are replaced with <1 (existing category for this column).
company_size	31%	No reason to assume why any candidate left this blank so missing values are replaced with 'Other'(new category for this column).

company_type	32%	No reason to assume why any candidate left this blank so missing values are replaced with 'Other' (existing category for this column).
last_new_job	2%	High likelihood that people who never had a job left this blank so missing values are replaced with 'never' (existing category for this column).

Experience

The experience column had too many categories(21) so tried a couple of things to solve that problem:

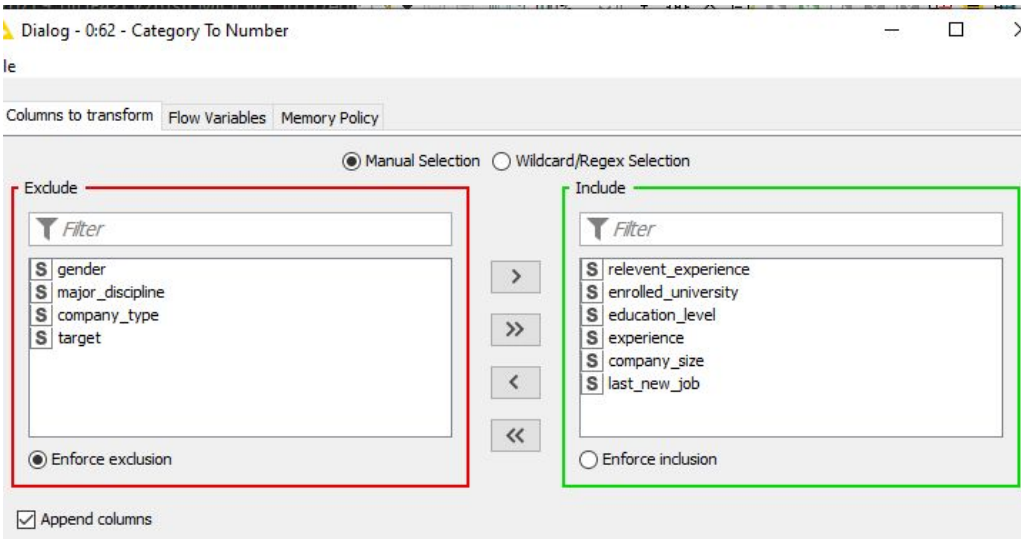
1. Joined a few categories to make fewer categories. '<1', '1' and '2' were combined to make 1 category '0-2' and so on.
2. Convert Experience into a numeric attribute.

The 1st option gave better ROC values so that was chosen.

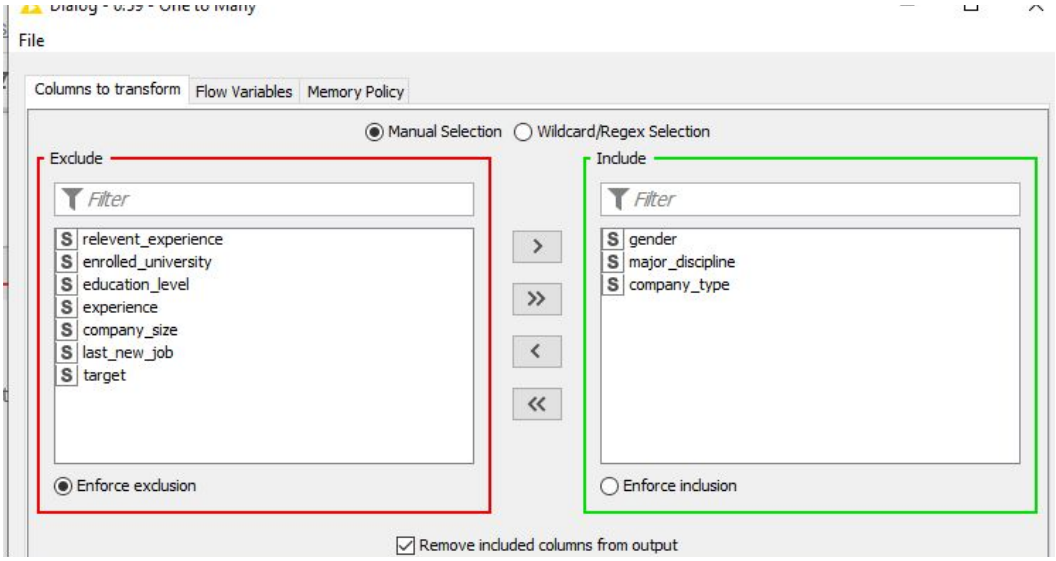
Encoding Categorical Variables

Since most of the attributes are categorical, some form of encoding is required. There are 2 common choices:

1. **Label Encoding**: This involves converting each unique category in a column into a unique number. For example in the enrolled_university column , 'no_enrollment' will be replaced with 0, 'Part time course' will be replaced with 1, and 'Full time course' will be replaced with 2.
2. **One-Hot Encoding**: This involves creating a different column for each category in For e.g. in the gender column, there will be a column for Male, Female and Other. A male candidate will have a 1 in the Male column and 0 in the other columns.



Label Encoding is applied to ordinal attributes so in our case it is applied to the 6 columns on the left. (Category To Number node is used to do Label Encoding).



One-Hot Encoding is applied to nominal attributes so in our case it is applied to the 3 columns on the right(One To Many node is used to do One-Hot Encoding in Knime).

Model Building + Evaluation

Evaluation Metric

The problem is an imbalanced class problem since the class label has 75% '0' values and 25% '1' values. Hence, the evaluation metric that needs to be optimized is **ROC**.

Feature Selection

To perform Feature Selection, the Backward Feature Elimination and Forward Feature Selection techniques were used with the help of nodes with the same name on Knime. Backward Feature Elimination did a better job of optimizing the evaluation metric and suggested 6 columns(out of 11) to be used for best performance: city_development_index, education_level, major_discipline, experience, company_size, training_hours.

Algorithms

Classification algorithms like Naive Bayes, Decision Trees, Tree Ensemble, Random Forest and Gradient Boosted Trees were used and the results achieved by each of them on different parameters are described in detail below.

Naive Bayes

- Naive Bayes did not have any parameters that could be tuned but the Naive Bayes

model resulted in a decent ROC value of 0.7780.

Decision Tree

- The attributes other than the ones mentioned (in the first row) did not bring any significant change to the evaluation metric therefore we did not record it.
- Enabling and increasing the max Binary Nominal splits had a positive effect on the evaluation metric.
- Disabling the Average Split Node had a positive effect on the evaluation metric.
- For Quality measure, Gini Index had a positive effect on the evaluation metric.
- Highest ROC value recorded for Decision Tree was 0.7428.

Binary nominal splits (max)	Min no. of records per node	Average split point	Pruning method	Quality measure	P(target=1.0)
disabled	2	Enabled	No pruning	Gini index	0.6709
Disabled	2	disabled	No pruning	Gini index	0.6716
Disabled	2	disabled	MDL	Gini index	0.6692
Disabled	2	disabled	No pruning	Gain Ratio	0.6924
Disabled	5	disabled	No pruning	Gain Ratio	0.7142
Disabled	10	disabled	No pruning	Gain Ratio	0.7347
5	10	Disabled	No pruning	Gain Ratio	0.7425
10	10	disabled	No pruning	Gain Ratio	0.7428

Tree Ensemble

- It learns multiple decision trees and the output model describes an ensemble of these decision tree models and is applied in the corresponding predictor node to aggregate the votes of the individual decision trees.
- Tree Ensemble model had parameters to tune but most of them did not bring about improvement in the evaluation metric.
- In the split criterion, Information Gain and gini index made few to no changes to the result whatsoever.
- Increasing number of models and patterns to record did not contribute much either.
- The tree depth and split size improved a lot to an extent and then it started decreasing the score after increasing it's value. The optimum value recorded was (depth=30, split size=10).

- Therefore we have only shown the record of attributes that improved the result.
- The highest score recorded was 0.7989

Ensemble Configuration		Tree Options					
No. of models	Attribute sampling	Split criterion	Tree depth	Min split node size	Min child node size	Highlighting (#patterns to store)	Target=1.0)
100	square root	Information Gain Ratio	disabled	disabled	disabled	disabled	0.7943
100	square root	Information Gain Ratio	30	disabled	disabled	3000	0.7968
100	square root	Information Gain Ratio	30	10	5	3000	0.7989

Random Forest

- Each parameter of Random Forest was tuned to optimize the evaluation metric.
- When it came to Split, both information Gain Ratio and Gini Index had a similar performance. However, Gini Index is known to be biased towards attributes with more categories. The dataset has some columns with quite a lot of categories so to ensure fairness, Information Gain Ratio was used.
- Increasing the depth led to an increase in performance uptil 17. Increasing depth more than 17 affected the model negatively.
- Increasing no. of models led to an increase in performance uptil 100. Increasing # of models more than 100 had no fixed pattern.
- So max ROC was achieved with no. of models=100, depth=4, Split=Information Gain Ratio.

# of Models	**Depth**	**Split**	**ROC**
100	10	Information Gain Ratio	0.7960
100	10	Information Gain	0.7946
100	10	Gini Index	0.7960
100	15	Information Gain Ratio	0.8028
100	20	Information Gain Ratio	0.8015

100	17	Information Gain Ratio	**0.8037**
100	18	Information Gain Ratio	0.8025
100	25	Information Gain Ratio	0.8001
150	15	Information Gain Ratio	0.8027
150	17	Information Gain Ratio	0.8028
150	25	Information Gain Ratio	0.7995
200	15	Information Gain Ratio	0.8021
200	17	Information Gain Ratio	0.8030
200	25	Information Gain Ratio	0.8003
250	15	Information Gain Ratio	0.8021
250	17	Information Gain Ratio	0.8018
250	25	Information Gain Ratio	0.8007

Gradient Boosted Trees

Gradient Boosted Trees performed the best out of all algorithms used.

# of Models	**Depth**	**ROC**
100	4	**0.8045**
100	10	0.7840
100	15	0.7780
150	4	0.8015
150	10	0.7812
150	15	0.7815
200	4	0.8012

200	10	0.7799
200	15	0.7818
200	20	0.7543

Not doing any attribute sampling resulted in the above table. Best ROC value was achieved at a depth of 4 and 100 models. Increasing depth over 4 and increasing models over 100 negatively impacted the evaluation metric.

Attribute Sampling	**ROC**
All columns	0.8045
Sample(square root)	**0.8071**
Sample(linear fraction)	0.8039
Sample(abs value)	0.8029

Square Root Sampling turned out to be the best Attribute Sampling method for the evaluation metric.

Furthermore, changing the default setting and not using midpoints splits for numeric attributes increased the ROC by 0.0003.

Hence best **ROC=0.8073** was achieved using depth=4, #models=100, Sampling(Square root) and not using midpoint splits for numeric attributes.

Findings

Insight

- The most useful attributes for prediction were city_development_index, education_level, major_discipline, experience, company_size and training_hours.
- Algorithm performance for our problem on this dataset can be ranked in the following order: Gradient Boosted Trees > Random Forest > Tree Ensemble >

Naive Bayes Decision Tree.

Model Expiry

Our model can potentially be used forever. However, as the company organizes more courses and collects more data, they are advised to provide us with that data. More data can be used in 2 ways:

1. Train the same model with the same configuration with old + new data. More data will mean our model has a good chance of being better at classification with the same configuration as before.
2. Redo the complete procedure for the new+old data starting from Data Description. This will be a more expensive procedure but will ensure that we get the best model possible for our data.

Limitation

The biggest problem with the data was the amount of missing values. The columns with the greatest percentage of missing values were company_size and company_type. Both these columns give us insight about the type of company the candidate is currently a part of, which is crucial information for our desired prediction. We had to make some assumptions about the data to handle those missing values as described in the Data Preprocessing section.

Suggestions for Future Data Collection

1. If the organization could ensure that everyone fills the form completely, it would lead to the Machine Learning algorithm doing all the learning itself by finding patterns in data and not overfitting to the assumptions we have made about the missing data.
2. The city attribute could not provide any useful insight since city_development_index is already present. Maybe if the data collected included province or some other geographical attribute, that could provide some useful insight which would be helpful for classifying the candidate.