# Implementing One Hot Encoding using KNIME

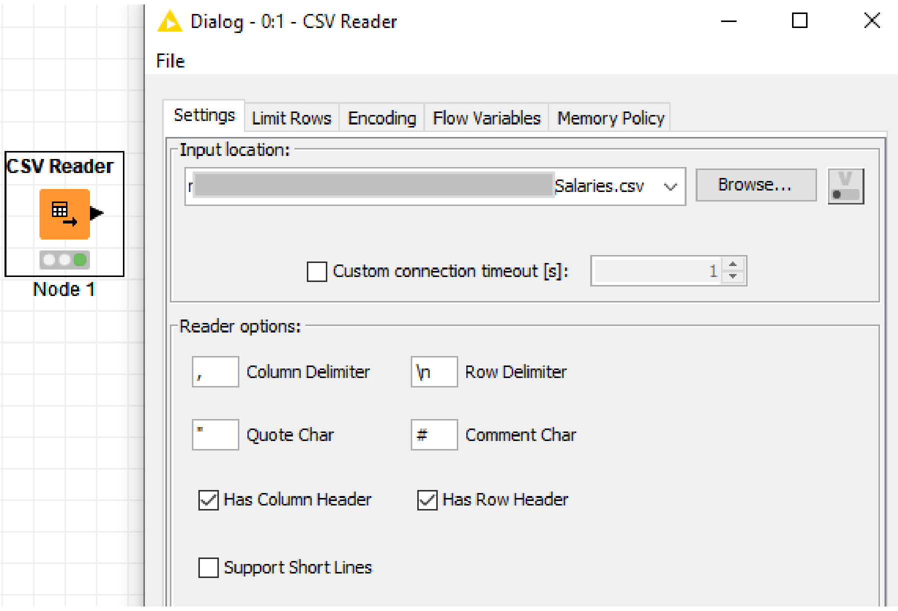
In this demonstration, we will use One-Hot Encoding to transform a dataset with categorical variables into one with only quantitative variables to allow modelling via linear regression.

One-Hot Encoding results in a Dummy Variable Trap as the outcome of one variable can easily be predicted with the help of the remaining variables. The Dummy Variable Trap is a scenario in which variables are highly correlated to each other. It leads to the problem known as multicollinearity. Multicollinearity occurs where there is a dependency between the independent features. It is a serious issue in machine learning models like Linear Regression and Logistic Regression.

In order to overcome the problem of multicollinearity, one of the dummy variables has to be dropped. Hence, for a categorical variable with ***n*** unique values, we use one-hot encoding for ***n-1*** variables.

### Using the One To Many node in KNIME for One-Hot Encoding

1. Use a *CSV Reader* node to read in the datasets/Salaries.csv file. Remember to check the "Has Row Header" checkbox.

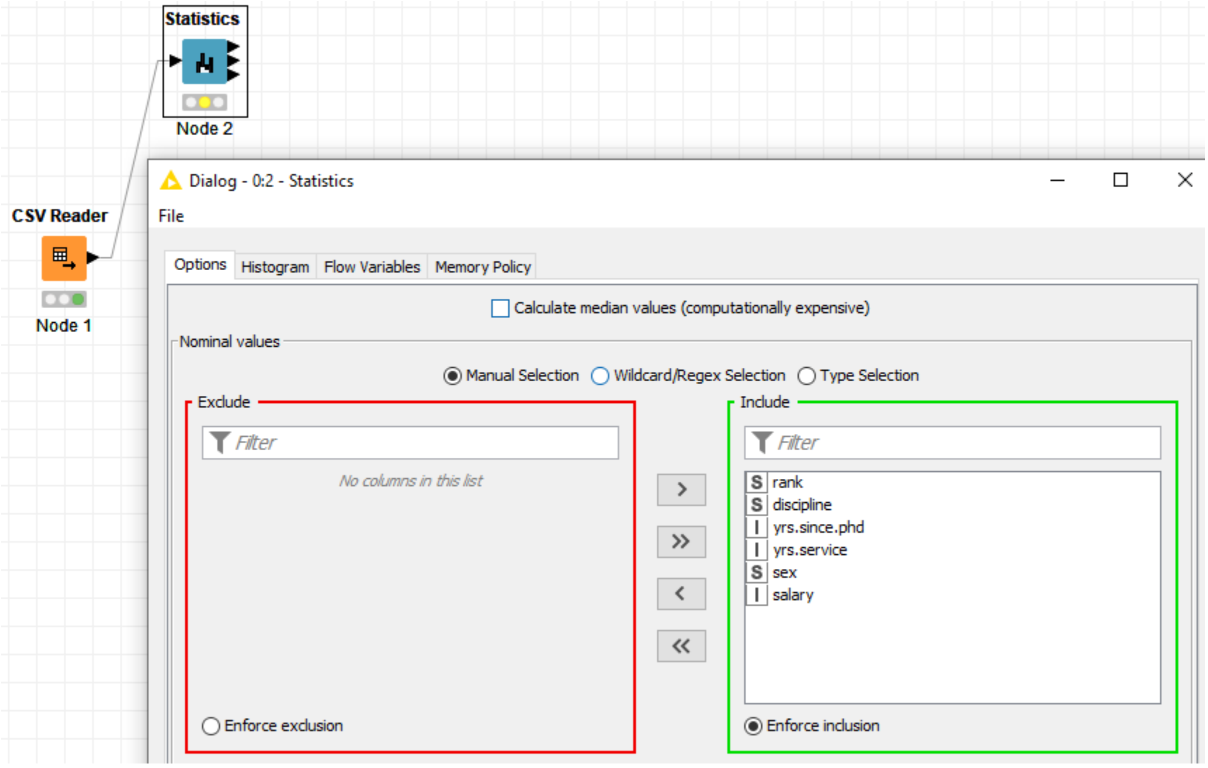



The data looks like this:

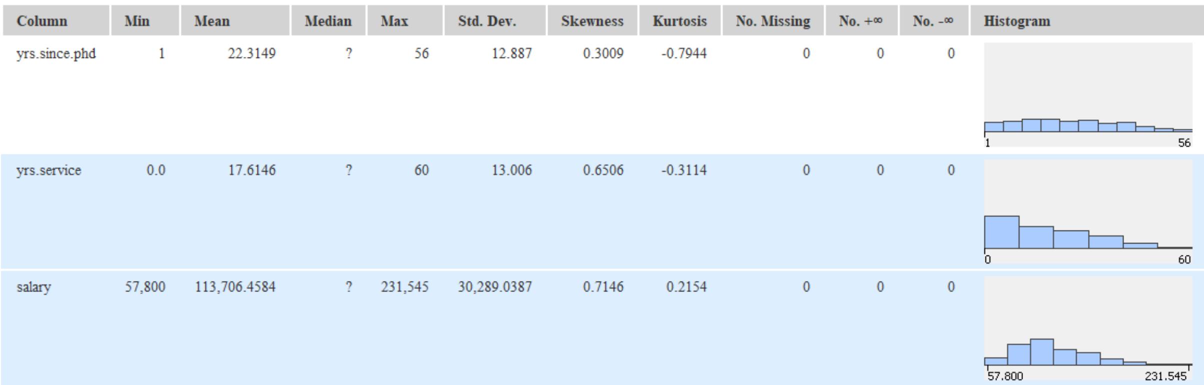
Table "Salaries.csv" - Rows: 397 | Spec - Columns: 6 | Properties | Flow Variables

| Row ID | S rank | S discipline | I yrs.sinc... | I yrs.ser... | S sex | I salary |
|---|---|---|---|---|---|---|
| 1 | Prof | B | 19 | 18 | Male | 139750 |
| 2 | Prof | B | 20 | 16 | Male | 173200 |
| 3 | AsstProf | B | 4 | 3 | Male | 79750 |
| 14 | AsstProf | B | 2 | 0 | Male | 78000 |
| 20 | Prof | A | 39 | 36 | Female | 137000 |

2. Use a *Statistics* node to explore the data:



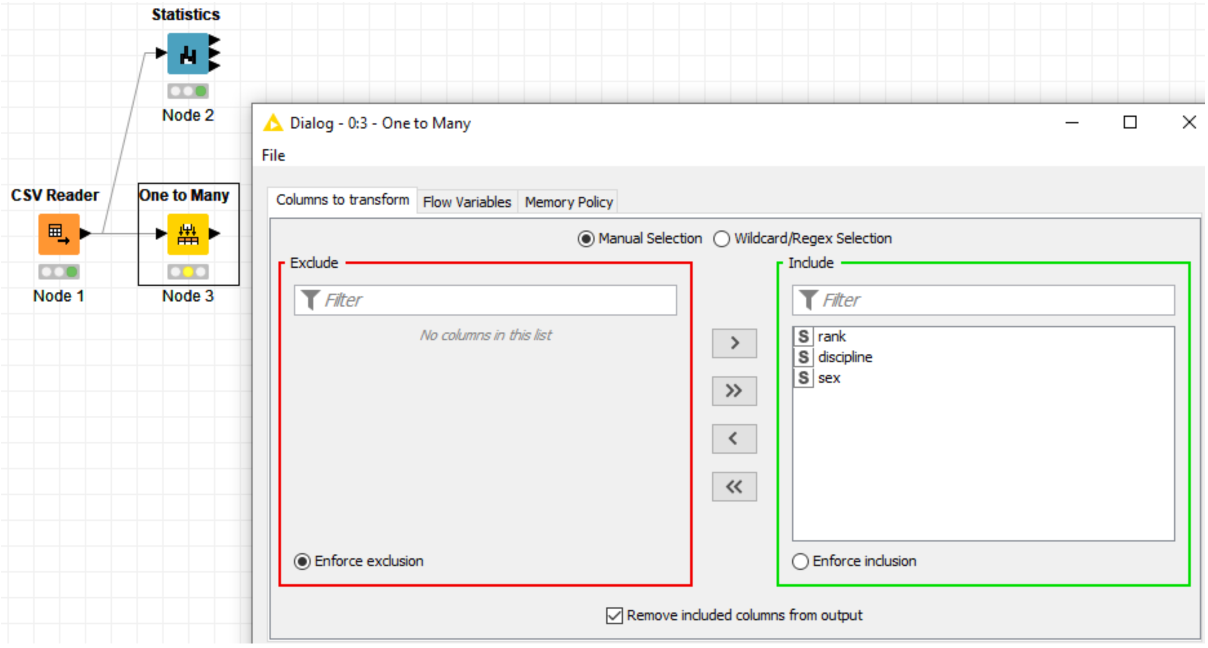

The quantitative variables have statistics as such. Most importantly, the distributions do not seem to have anomalies, and there are no missing data.



| Column | Min | Mean | Median | Max | Std. Dev. | Skewness | Kurtosis | No. Missing | No. +∞ | No. -∞ | Histogram |
|---|---|---|---|---|---|---|---|---|---|---|---|
| yrs.since.phd | 1 | 22.3149 | ? | 56 | 12.887 | 0.3009 | -0.7944 | 0 | 0 | 0 | 1 56 |
| yrs.service | 0.0 | 17.6146 | ? | 60 | 13.006 | 0.6506 | -0.3114 | 0 | 0 | 0 | 0 60 |
| salary | 57,800 | 113,706.4584 | ? | 231,545 | 30,289.0387 | 0.7146 | 0.2154 | 0 | 0 | 0 | 57.800 231.545 |

The statistics of the qualitative variables show that there is no missing data also.

3. Connect a *One to Many* node to transform the categorical variables of rank, discipline and sex.




The processed data looks like this:

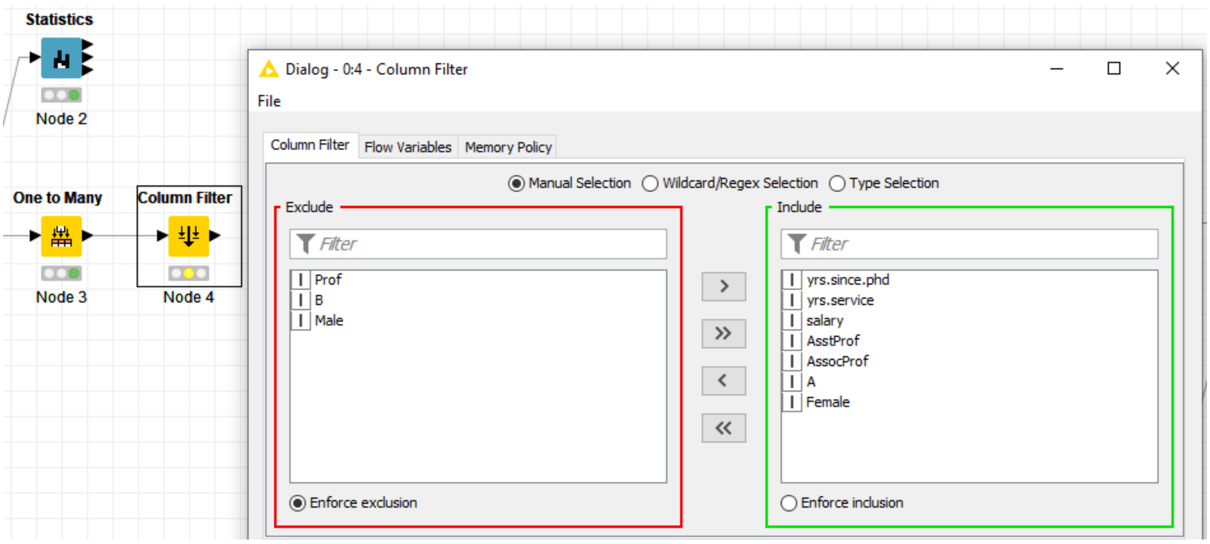
Processed data - 0:3 - One to Many

File Hilite Navigation View

Table "default" - Rows: 397 | Spec - Columns: 10 | Properties | Flow Variables

| Row ID | yrs.sinc... | yrs.ser... | salary | Prof | AsstProf | AssocProf | B | A | Male | Female |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 19 | 18 | 139750 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2 | 20 | 16 | 173200 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 3 | 4 | 3 | 79750 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 4 | 45 | 39 | 115000 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 5 | 40 | 41 | 141500 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 6 | 6 | 6 | 97000 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 7 | 30 | 23 | 175000 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 8 | 45 | 45 | 147765 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 9 | 21 | 20 | 119250 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 10 | 18 | 18 | 129000 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 11 | 12 | 8 | 119800 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 12 | 7 | 2 | 79800 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 13 | 1 | 1 | 77700 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 14 | 2 | 0 | 78000 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 15 | 20 | 18 | 104800 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 16 | 12 | 3 | 117150 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 17 | 19 | 20 | 101000 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 18 | 38 | 34 | 103450 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 19 | 37 | 23 | 124750 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 20 | 39 | 36 | 137000 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

4. Since we only should have *n-1* dummy coding for *n* unique values, we shall choose (arbitrarily) the values “Prof”, “B” and “Male” as the base values on which to base the other values on. We can use a *Column Filter* node to perform this step:

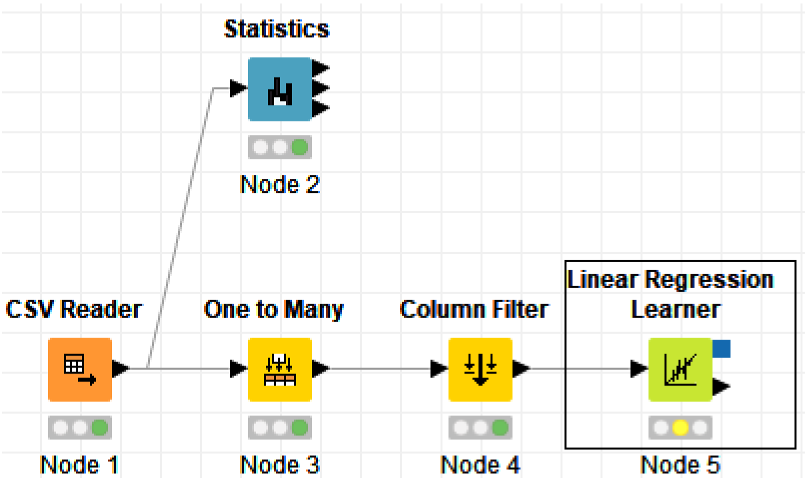



The filtered table will look like this:

Table "default" - Rows: 397 | Spec - Columns: 7 | Properties | Flow Variables

| Row ID | yrs.sinc... | yrs.ser... | salary | AsstProf | AssocProf | A | Female |
|---|---|---|---|---|---|---|---|
| 1 | 19 | 18 | 139750 | 0 | 0 | 0 | 0 |
| 2 | 20 | 16 | 173200 | 0 | 0 | 0 | 0 |
| 3 | 4 | 3 | 79750 | 1 | 0 | 0 | 0 |
| 4 | 45 | 39 | 115000 | 0 | 0 | 0 | 0 |
| 5 | 40 | 41 | 141500 | 0 | 0 | 0 | 0 |
| 6 | 6 | 6 | 97000 | 0 | 1 | 0 | 0 |
| 7 | 30 | 23 | 175000 | 0 | 0 | 0 | 0 |
| 8 | 45 | 45 | 147765 | 0 | 0 | 0 | 0 |
| 9 | 21 | 20 | 119250 | 0 | 0 | 0 | 0 |
| 10 | 18 | 18 | 129000 | 0 | 0 | 0 | 1 |
| 11 | 12 | 8 | 119800 | 0 | 1 | 0 | 0 |

5. Use a *Linear Regression Learner* to build the model to predict the salary based on the other attributes:



The regression model has the following coefficients and statistics:

Table "Coefficients and Statistics" - Rows: 7 | Spec - Columns: 5 | Properties | Flow Variables

| Row ID | S Variable | D Coeff. | D Std. Err. | D t-value | D P>\|t\| |
|---|---|---|---|---|---|
| Row1 | yrs.since.phd | 535.058 | 240.994 | 2.22 | 0.027 |
| Row2 | yrs.service | -489.516 | 211.938 | -2.31 | 0.021 |
| Row3 | AsstProf | -45,065.999 | 4,237.523 | -10.635 | 0 |
| Row4 | AssocProf | -32,158.411 | 3,540.647 | -9.083 | 0 |
| Row5 | A | -14,417.626 | 2,342.875 | -6.154 | 0 |
| Row6 | Female | -4,783.493 | 3,858.668 | -1.24 | 0.216 |
| Row7 | Intercept | 130,222.349 | 3,907.327 | 33.328 | 0 |

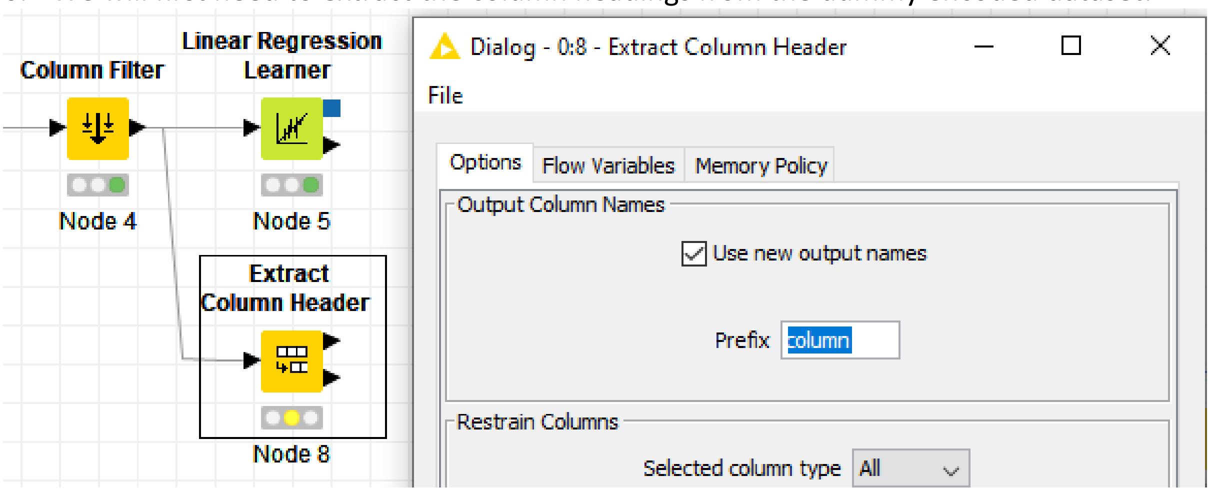
**Results discussion:**

- All variables are significant (based on p-value) except for Female. It seems that gender does not play an important role in determining an academic's salary.
- The more years one has a PHD, the higher the salary.
- However, the more years one is in service, the lower the salary? This actually makes sense when you consider two different academics, both who had received their PHD 15 years ago. One has been at work in the same university for 10 years, while the other was head-hunted to join the university 2 years ago. For the former academic, the combined effect of "yrs.since.phd" and "yrs.service" will yield 15*535.058 + 10*(-489.516) = 3130.17, while the latter will have 15*535.058 + 2*(-489.516) = 7046.838. The latter academic has a higher premium as he was deemed valuable enough to be head-hunted to the new university.
- Prof being the base value, it is not surprising to see AsstProf and AssocProf having negative coefficients.
- Discipline B commands a higher salary than Discipline A. This could be because Discipline B is in the hard science domain while Discipline A could be in the social sciences domain.

**Applying model to new data**

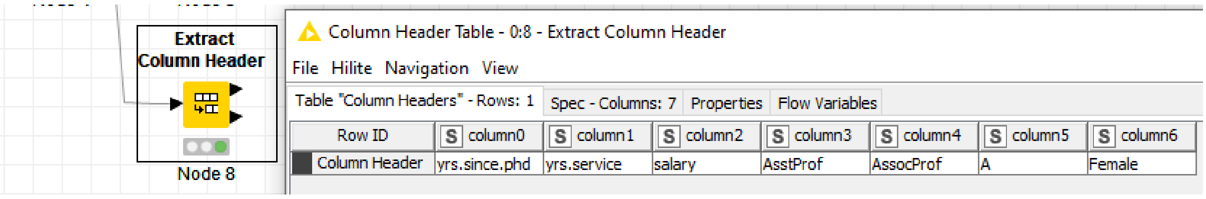
We will next attempt to apply the model to a new set of data to predict the salary of the new data point. To do that, we need to perform a number of steps of data transformation.

6. We will first need to extract the column headings from the dummy encoded dataset:



Take note to change the output column names to start with “column” with no space behind the word.

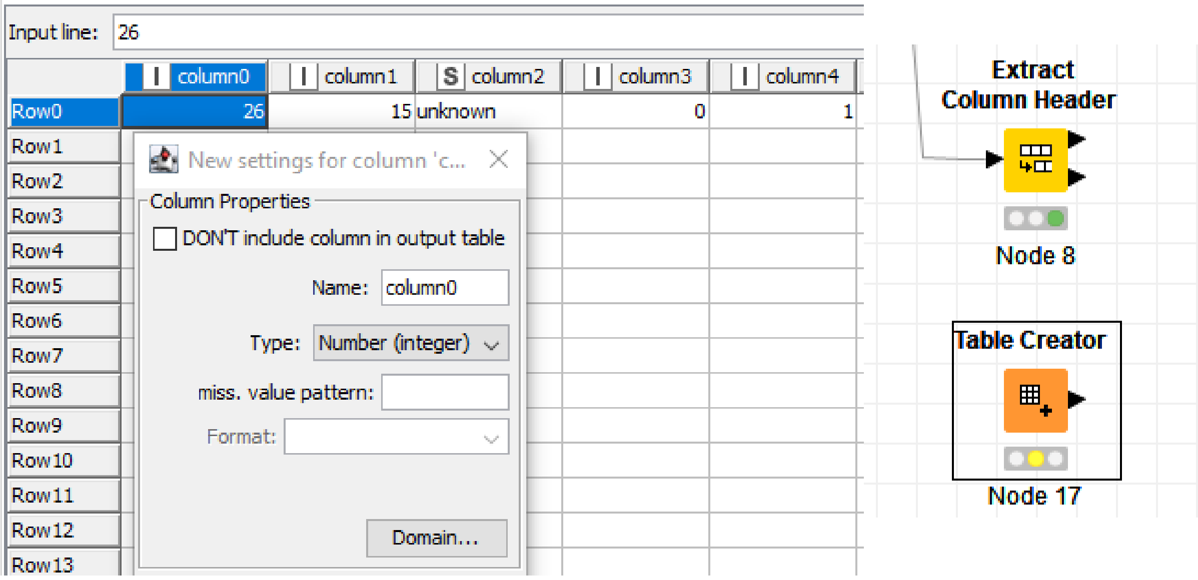
The result of the extraction is as such:



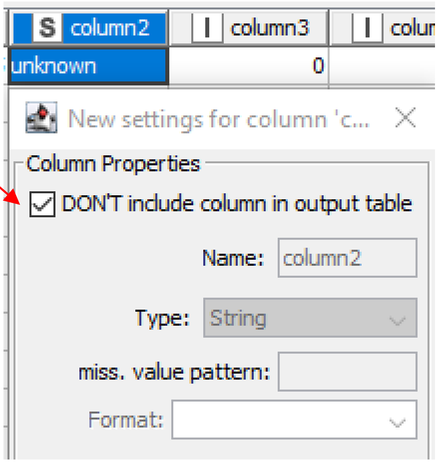

| Row ID | S column0 | S column1 | S column2 | S column3 | S column4 | S column5 | S column6 |
|---|---|---|---|---|---|---|---|
| Column Header | yrs.since.phd | yrs.service | salary | AsstProf | AssocProf | A | Female |

7. We will use a *Table Creator* node to create a data point:

Configure this node like this:

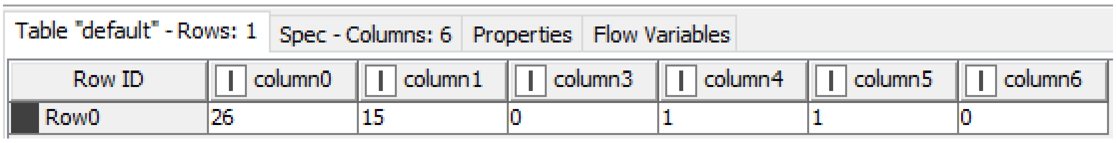



where column0 = 26* refers to yrs.since.phd column,
column1 = 15* refers to yrs.service column,
column2 = unknown# refers to salary column,
column3 = 0* refers to AsstProf column,
column4 = 1* refers to AssocProf column,
column5 = 1* refers to A column, and
column6 = 0* refers to Female column.



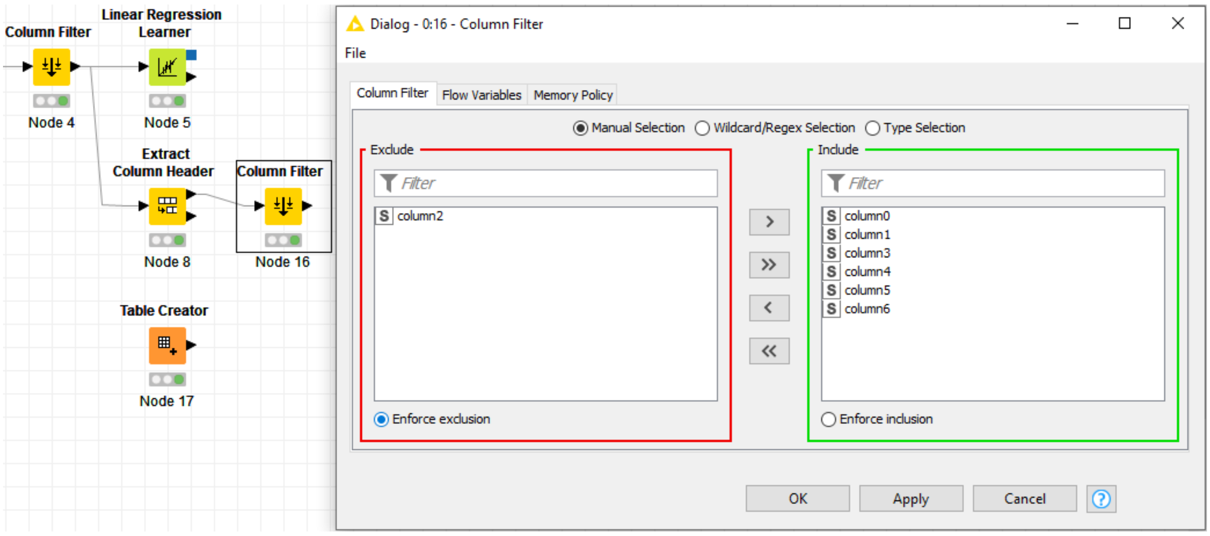

*: change type of variable to “Number (integer)”
#: exclude column from output table, as this is the column we wish to predict

Output table:

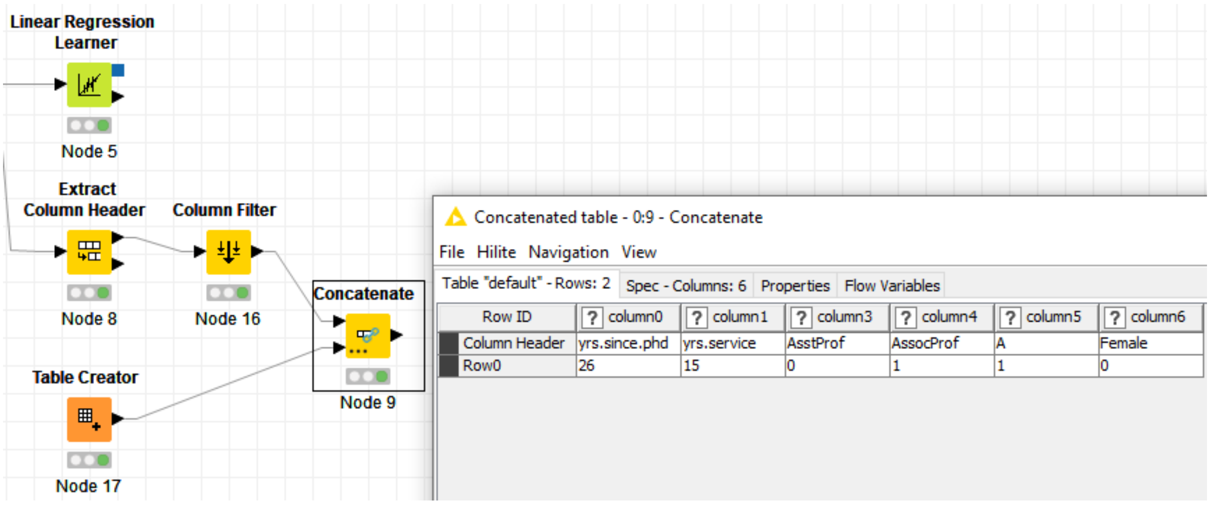



| Row ID | I column0 | I column1 | I column3 | I column4 | I column5 | I column6 |
|---|---|---|---|---|---|---|
| Row0 | 26 | 15 | 0 | 1 | 1 | 0 |

8. Since in the output of the *Table Creator* node we excluded the salary column, likewise we have to exclude the salary header from the output of *Extract Column Header*. To do that we can use the *Column Filter* node.

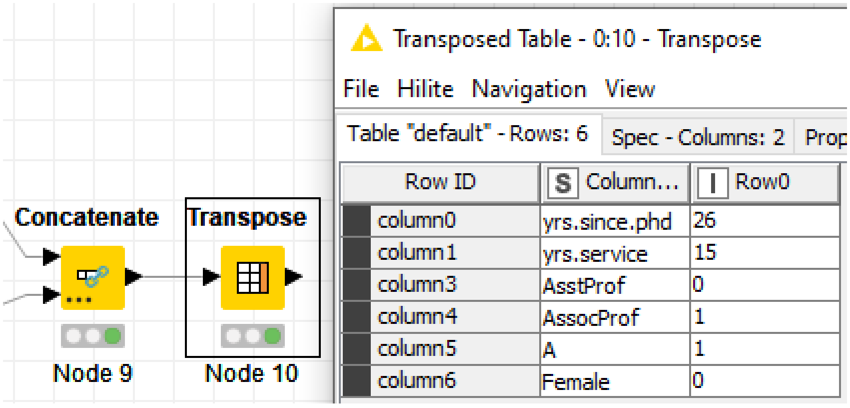



9. We are now ready to concatenate the headers with the data we created for prediction.



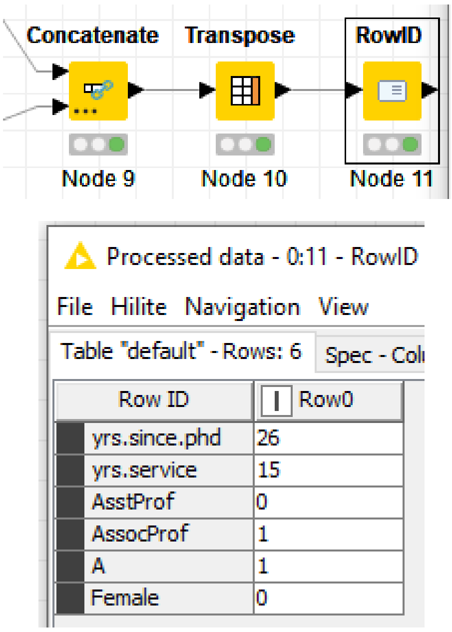

| Row ID | column0 | column1 | column3 | column4 | column5 | column6 |
|---|---|---|---|---|---|---|
| Column Header | yrs.since.phd | yrs.service | AsstProf | AssocProf | A | Female |
| Row0 | 26 | 15 | 0 | 1 | 1 | 0 |

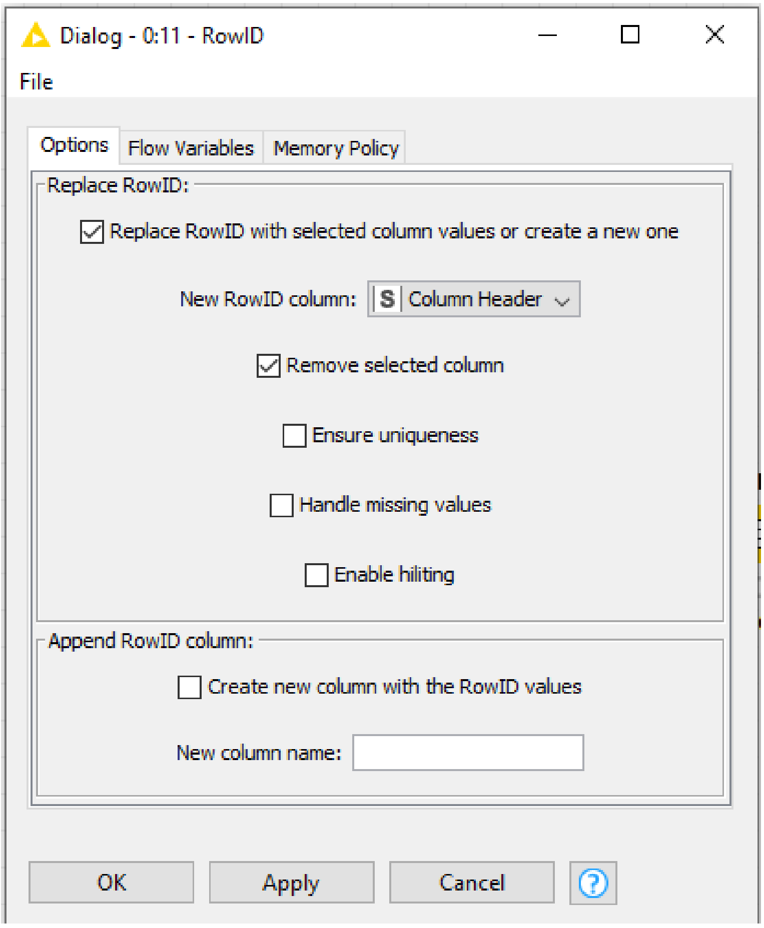
10. To make the "Column Header" row the actual column headers, a little trick is necessary. First we transpose the data so that "Column Header" becomes a column instead:



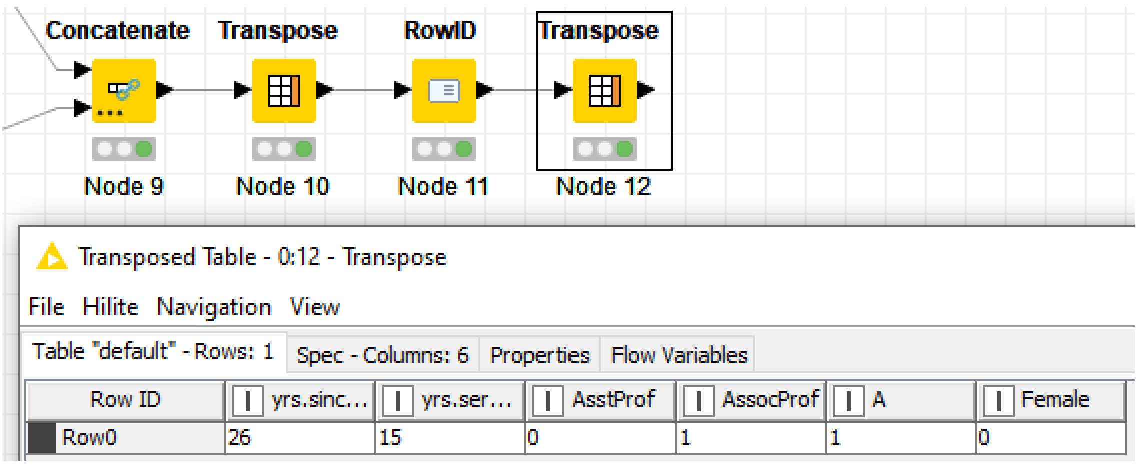

| Row ID | Column... | Row0 |
|---|---|---|
| column0 | yrs.since.phd | 26 |
| column1 | yrs.service | 15 |
| column3 | AsstProf | 0 |
| column4 | AssocProf | 1 |
| column5 | A | 1 |
| column6 | Female | 0 |

11. Then we make the “Column Header” column the new id column using the *RowID* node (all this work is needed as there isn’t an equivalent *ColumnID* node).

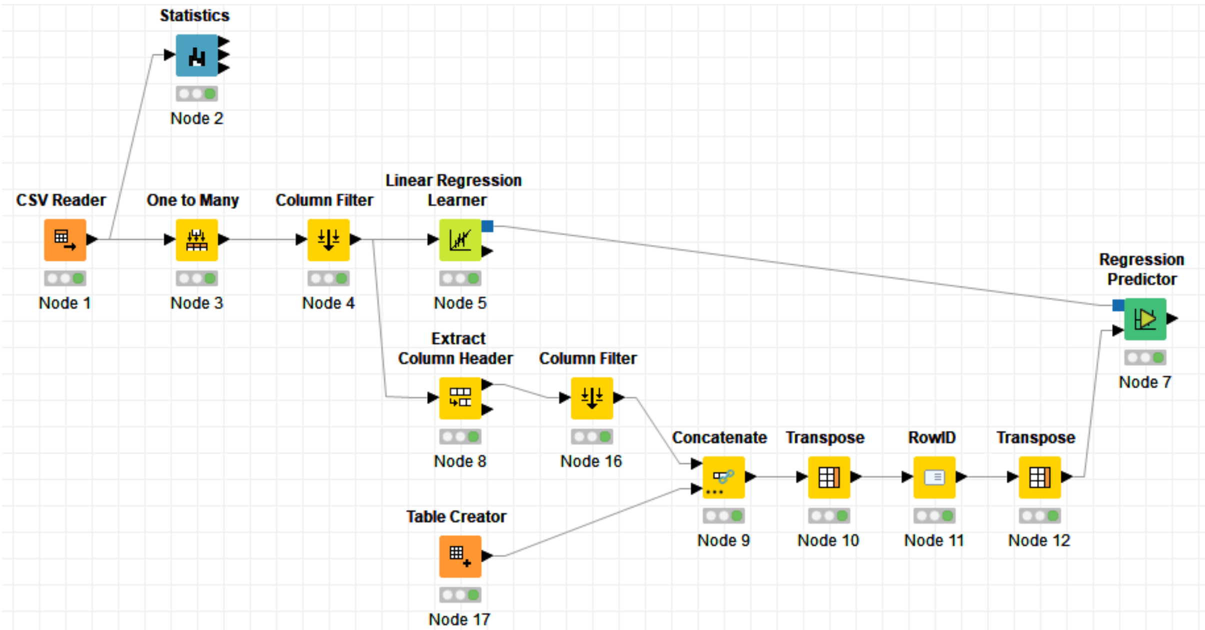



12. Now we can transpose the data back:



13. At last, we are ready to apply the regression model trained to the data point we created:




14. The predicted salary of the data point is as shown i.e. $90215.093:

Predicted data - 0:7 - Regression Predictor

File Hilite Navigation View

Table "default" - Rows: 1 | Spec - Columns: 7 | Properties | Flow Variables

| Row ID | I yrs.sinc... | I yrs.ser... | I AsstProf | I AssocProf | I A | I Female | D Predicti... |
|---|---|---|---|---|---|---|---|
| Row0 | 26 | 15 | 0 | 1 | 1 | 0 | 90,215.093 |

A check with similar data in the original dataset tells us that the prediction is not too far-fetched:

| | | | | | | |
|---|---|---|---|---|---|---|
| 216 | Prof | B | 16 | 11 | Male | 145350 |
| 217 | Prof | B | 15 | 11 | Male | 146000 |
| 218 | AssocProf | B | 29 | 22 | Male | 105350 |
| 219 | AssocProf | B | 14 | 7 | Female | 109650 |
| 220 | Prof | B | 13 | 11 | Male | 119500 |
| 221 | Prof | B | 21 | 21 | Male | 170000 |
| 222 | Prof | B | 23 | 10 | Male | 145200 |

**Conclusion**

We have seen how to perform regression on data with nominal variables by performing one-hot encoding. To prevent the problem of multicollinearity, we have to drop one dummy variable for every nominal feature converted using one-hot encoding.