

Implementing Linear Regression using KNIME

In this demonstration, we will use the KNIME Regression Learner node to see how to predict insurance claim amount

Data

In this practical, we will use the data file *datasets/InsuranceClaim.csv*, which contains 293 records based on patient admissions to a hospital. All patients belong to a single diagnosis related group (DRG). Four fields are included:

	A	B	C	D
1	ASG	AGE	LOS	CLAIM
2	0	23	2	3619
3	1	20	3	4590.6
4	1	29	1	3850
5	1	27	2	11187.4
6	1	20	2	4368
7	0	25	3	5763.8
8	0	26	3	5108.6
9	1	37	2	4582.2
10	0	37	2	4677.4
11	0	30	2	3904.6
12	1	22	1	5170.2
13	0	41	2	3362.8
14	0	26	2	3785.6

The fields are described below:

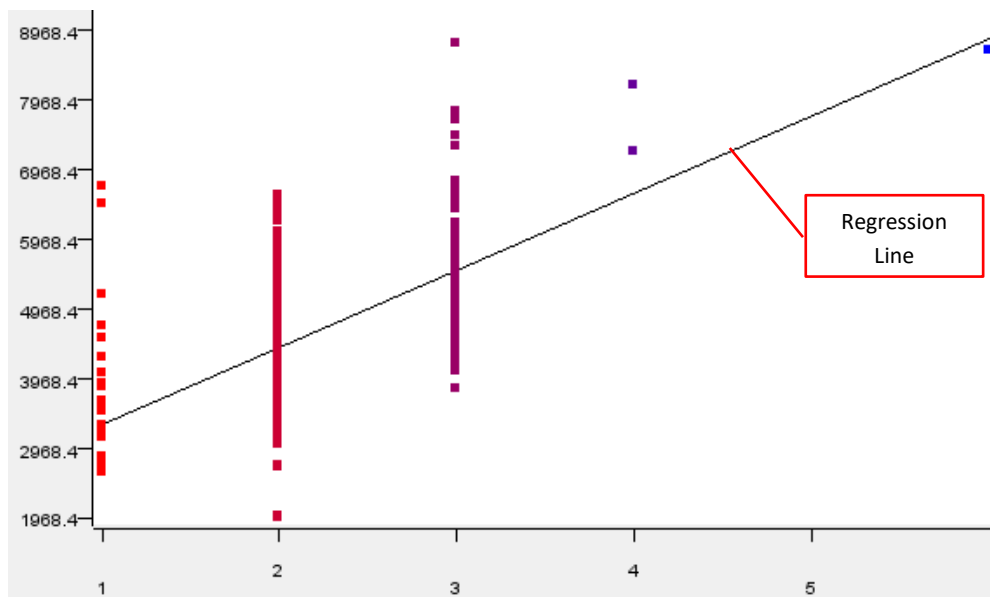
- ASG - Admission Severity Group
- AGE – Age of Patient
- LOS – Length of Stay
- CLAIM - Insurance claim amount

The goal is to build a predictive model for the insurance claim amount (CLAIM) and use this model to identify outliers (patients with claim values far from what the model predicts), which might be instances of errors or fraud made in the claims.

Introduction to Linear Regression

Linear regression is a method familiar to just about everyone these days. It is used to predict a target that is interval or ratio in scale (measurement level continuous) with predictors that are also interval or ratio. In addition, categorical input fields can be included by creating dummy variables.

Linear regression assumes that the data can be modelled with a linear relationship. To illustrate, the figure below contains a scatterplot depicting the relationship between the length of stay for hospital patients and the dollar amount claimed for insurance. Superimposed on the plot is the best-fit regression line. The linear relationship is represented by a **straight** line on the graph.



SCATTERPLOT OF HOSPITAL LENGTH OF STAY AND INSURANCE CLAIM AMOUNT

The plot may look a bit unusual in that there are only a few values for length of stay (LOS), which is recorded in whole days, and few patients stayed more than three days. Although there is a lot of spread around the regression line and a few outliers, it is clear that there is a positive trend in the data such that longer stays are associated with greater insurance claims.

Linear regression is normally used with several input attributes, however, it is difficult to visualize the complete solution with all input attributes in convenient graphical form, but it is useful to look at scatterplots with just two variables.

Basic Concepts of Regression

In the plot above, there seems to be a positive relation between length of stay and the amount of a health insurance claim. It would be more useful to have some form of *prediction equation*. In other words, if we can find a simple function that can approximate the pattern shown in the plot, then the equation for the function would describe the relation and can be used to predict values of one field if we know the values of the others.

A straight line is a very simple function and is usually what we start with. However, the value of the straight line equation would be linked to how well it actually describes or fits the data, and so part of the regression output includes fit measures.

The Regression Equation

In the plot above, insurance claim amount is placed on the Y (vertical) axis and the length of stay appears along the X (horizontal) axis. If we are interested in insurance claim as a function of the length of stay, we consider insurance claim to be the output field and length of stay as the input or predictor field. A straight line is superimposed on the scatterplot along with the general form of the equation:

$$y_i = A + Bx_i + e_i$$

Where

- B is the slope (the change in y per one unit change in x)
- A is the intercept (the value of Y when X is zero)
- e_i is the model residual or error for the i^{th} observation

Given this, how do we go about finding a best-fitting straight line? The best-fitting straight line is the one that minimizes the sum of the squared deviation of each point about the line. Referring to the plot of insurance claim amount and length of stay, we might need an indication of how well a straight line fits the data.

One of the most often used measure is the *r-square* measure. The r-square measure (which is the correlation squared, or r^2) is on a scale from 0 (no linear association) to 1 (perfect prediction). It can be interpreted as the proportion of variation in one field that can be predicted from the other. Thus an r-square of 0.5 indicates that we can account for 50% of the variation in one field if we know values of the other. You can think of this value as a measure of the improvement in your ability to predict one field from the other (or others if there is more than one input field).

Multiple regression is a direct extension of simple regression. Instead of a single input field $y_i = A + Bx_i + e_i$, multiple regression allows for more than one input field in the prediction equation.

$$y_i = A + B_1x_{1i} + B_2x_{2i} + B_3x_{3i} + \dots + e_i$$

For multiple regression, besides concerns about how well the equation fits the data we are interested in the relative importance of the independent fields (predictor) in predicting the output field (outcome).

Residuals and Outliers

Viewing the plot, we see that many points fall near the line but some are farther away it. For each point, the difference between the value of the output field and the value predicted by the equation is called the residual (e_i). Points above the line have positive residuals (they were under predicted), those below the line have negative residuals (they were over predicted), and a point falling on the line has a residual of zero (perfect prediction).

We pay more attention to points having large residuals because they represent cases where the prediction line performs poorly. As we will see later, we can large residuals to identify data errors or possible cases of fraud. Fraud detection is important in insurance claims, invoice submission, or telephone and credit card usage.

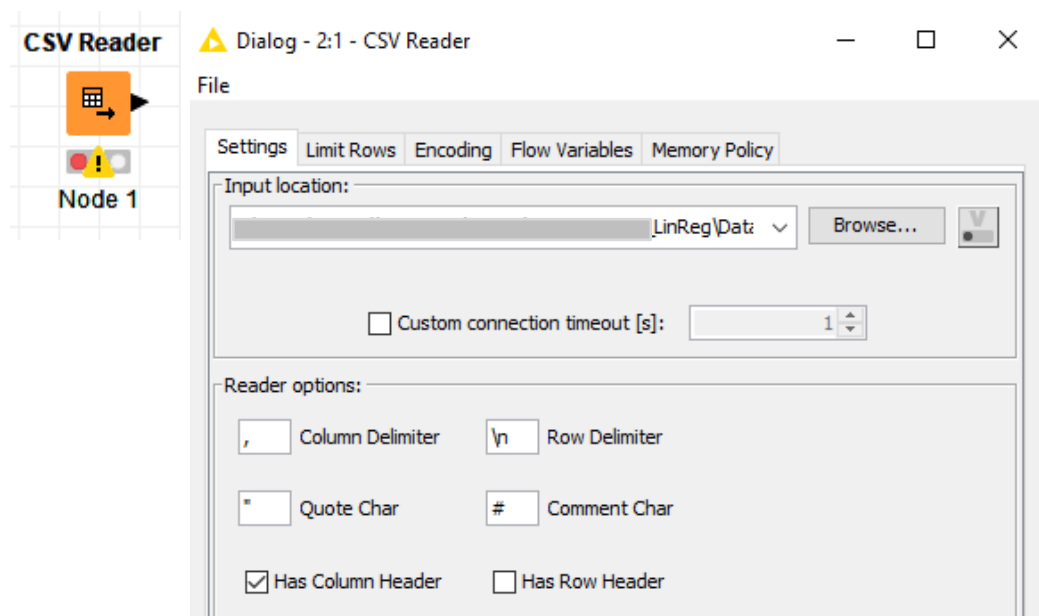
An Example: Error or Fraud Detection in Claims

To illustrate linear regression we use a dataset containing insurance claims (CLAIM) for a single medical treatment performed in a hospital. In addition to the claim amount, the data file also contains patient age (AGE), length of hospital stay (LOS) and a severity of illness category (ASG). The last field is based on several health measures and higher scores indicate greater severity of the illness.

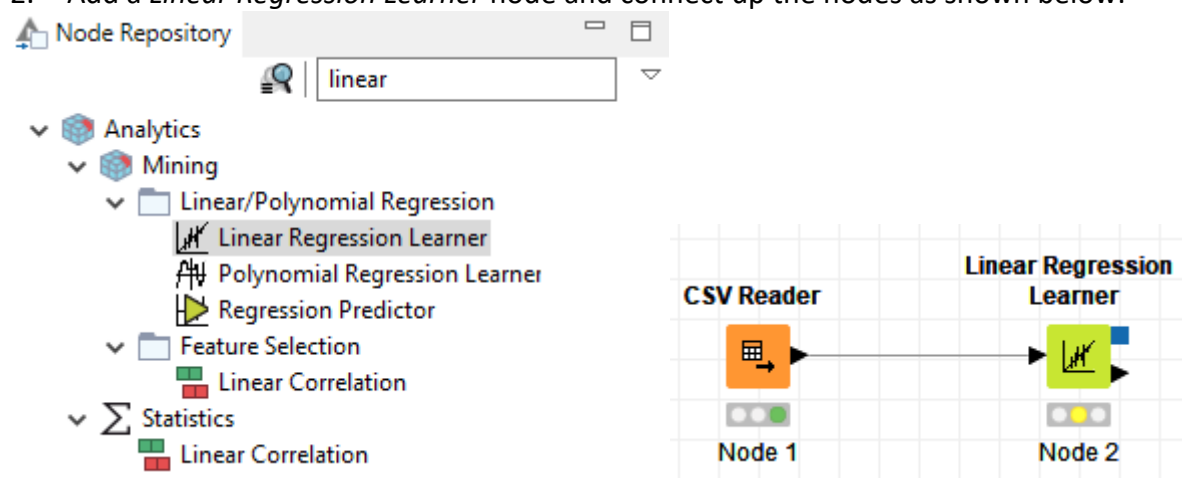
In this demonstration, we will build a regression model that predicts the total claim amount for a patient based on his/her length of stay, severity of illness and patient age. Assuming the model fits well enough, we will look closely at those patients that the model predicts poorly. Such cases can simply be due to poor model fit but they also might be due to errors on the claims form or fraud.

We approach the problem of fraud detection by identifying exceptions (cases that are very different) to the prediction model. Such exceptions are not necessarily instances of fraud, but since they are inconsistent with the model, they may be more likely to be fraudulent or contain errors. Some organizations perform random audits on claims applications and then classify them as fraudulent or not.

1. Import the file *InsuranceClaim.csv* using the *CSV Reader* node.

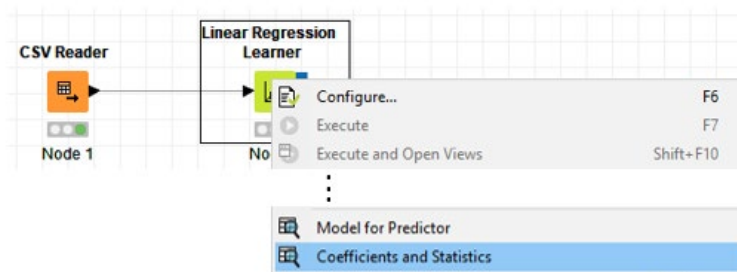


2. Add a *Linear Regression Learner* node and connect up the nodes as shown below:



3. Configure the Learner node as such:

4. Execute the Learner node and look at the Coefficients and Statistics of the model.



5. You can think of the model as an equation that helps us predict the CLAIM value based on the other input values (ASG, AGE and LOS). Besides the model, the node will also generate some figures that give us some idea of how good the equation will be in predicting. Of course, the more data there is and the more closely the data relates to each other (in a linear fashion), the better will be the model.

▲ Coefficients and Statistics - 2:2 - Linear Regression Learner

File Hilite Navigation View

Table "Coefficients and Statistics" - Rows: 4 Spec - Columns: 5 Properties Flow Variables						
Row ID	Variable	Coeff.	Std. Err.	t-value	P> t	
Row1	ASG	417.194	114.657	3.639	0	
Row2	AGE	-33.406	11.616	-2.876	0.004	
Row3	LOS	1,105.646	103.6	10.672	0	
Row4	Intercept	3,026.754	408.59	7.408	0	

Referring to the above figure, the column “Variable” of the table shows the various attributes and a constant value (intercept). The “Coeff.” column lists the coefficients of the equation. So the prediction equation for predicting CLAIM will be

$$\text{CLAIM} = 417.194 * \text{ASG} - 33.406 * \text{AGE} + 1105.646 * \text{LOS} + 3026.754$$

Given values of ASG, AGE and LOS, we can calculate (or predict) the CLAIM value. According to the equation, the coefficient for length of stay indicates that on average, each additional day spent in the hospital was associated with a claim increase of about \$1,106. The coefficient for admission severity group (ASG) tells us that each one-unit increase in the severity code is associated with a claim increase of \$417. Finally, the age coefficient of about -\$33 suggests that claims decrease, on average, by \$33 as patient age increases one year. Here the length of stay is the most important predictor of claim amount, followed by severity group and age.

The decrease in claim amount for increasing age is counterintuitive and should be examined by a domain expert (here a physician). Perhaps the youngest patients are at greater risk or perhaps the type of insurance policy, which is linked somehow to age, influences the claim amount. If there is no convincing reason for this negative association, the data values for age and claims should be examined more carefully (perhaps data errors or outliers are influencing the results). Such oddities may have shown up in the original data exploration. We will not pursue this issue here, but it certainly would be done in practice.

The constant or intercept of \$3,027 indicates that the amount of predicted claim of someone with 0 days in the hospital, in the least severe illness category (0) and with age 0. This is clearly impossible. We get this odd result because no one in the sample had less than 1 day in the hospital (it was an inpatient procedure) and the patients were adults (no ages of 0), so the intercept projects well beyond where there are any data. Thus the intercept cannot represent an actual patient, but still is needed to fit the data. Also, note that when using regression, it can be risky to extrapolate beyond where the data are observed, since the assumption that the same pattern continues may not be valid.

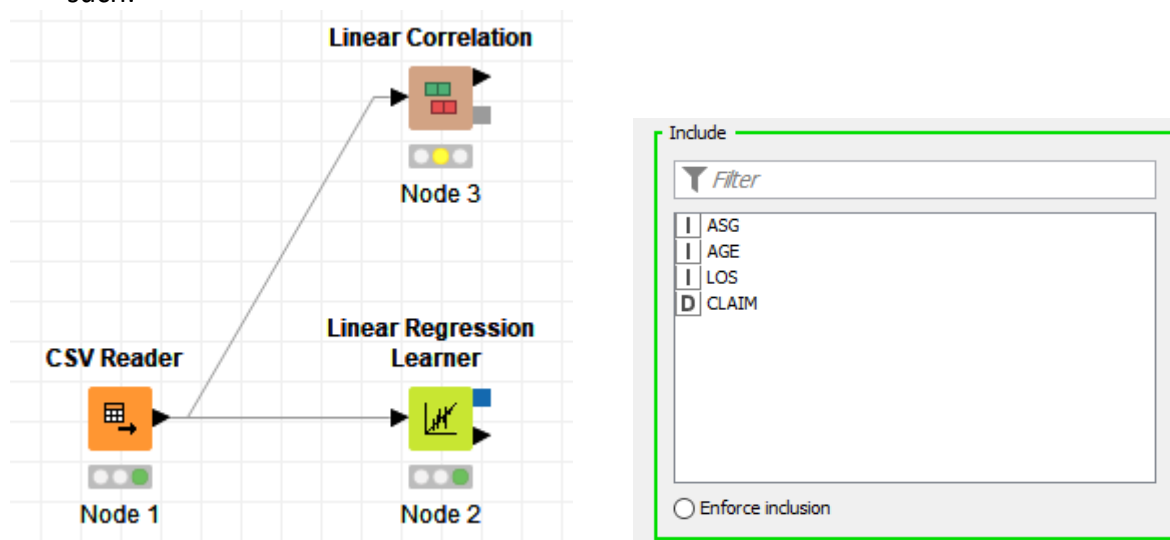
Again, the prediction is dependent on the coefficients and that is generated based on quality of the data. Note that if there is no underlying relationship between CLAIM and ASG, AGE and LOS, the model will be of poor quality as well. How significant or how well the input attributes are related to the CLAIM output field can be determined from the t-Stat and p-Value values.

Generally, the smaller the p-Value (or the further away the t-Stat value is from 0), the more likely the input attribute is related to the targeted output attribute. Looking at the p-Value, we see that all three attributes are highly significant (p-Values are .005 or less). If any of the attributes were found to be not significant, we would typically remove the attributes and rerun the regression.

The Std. Error column contains standard errors of the estimated regression coefficients. These gives us an idea of the precision of our estimation of the coefficients. In our example, the regression coefficient for length of stay is \$1,106 and the standard error is about \$104. Thus we would not be surprised if in the population the true regression coefficient were \$1,000 or \$1,200 (within two standard errors of our sample estimate), but it is very unlikely that the true population coefficient would be \$300 or \$2,000 (more than 2 standard errors).

We can also use the *Linear Correlation* node to see the relative importance of the attributes based on correlation of the attribute with the target or labelled attribute.

6. Connect a Linear Correlation node to the output of the CSV Reader and configure it as such:



7. Execute the node and you will see the following result:

▲ Correlation measure - 2:3 - Linear Correlation

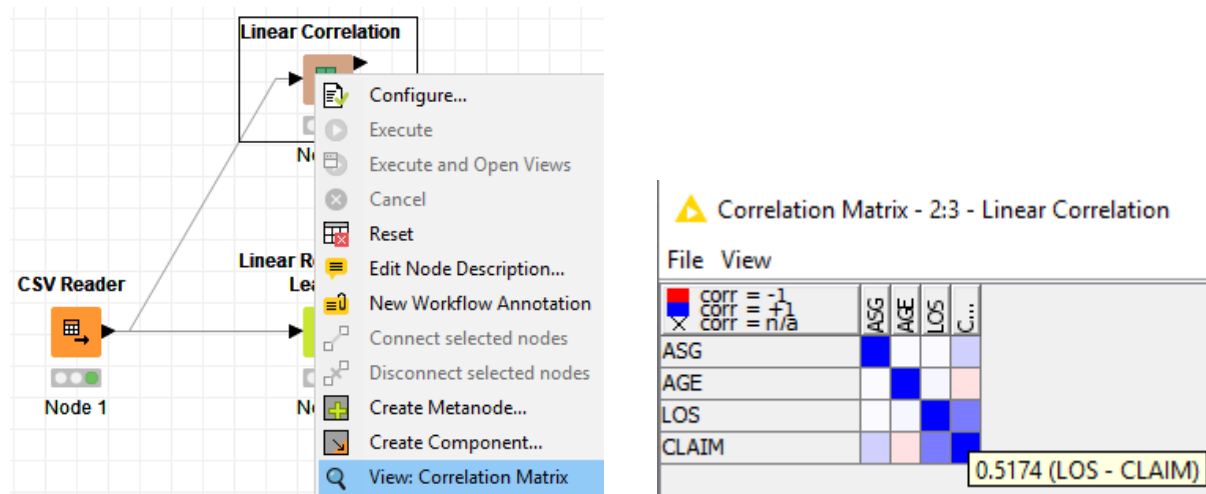
File Hilite Navigation View

Table "Correlation values" - Rows: 4 Spec - Columns: 4 Properties Flow Variables

Row ID	D ASG	D AGE	D LOS	D CLAIM
ASG	1.0	0.016213964942368...	0.020050297642853...	0.18499324538098...
AGE	0.01621396...	1.0	0.03714071215293228	-0.11770631193204...
LOS	0.02005029...	0.03714071215293228	1.0	0.5173628621312206
CLAIM	0.18499324...	-0.11770631193204789	0.5173628621312206	1.0

The higher the weight value, the more relevant the attribute is to the target attribute. As can be seen from the table shown in the figure above, the LOS is the more relevant following by ASG and finally the AGE.

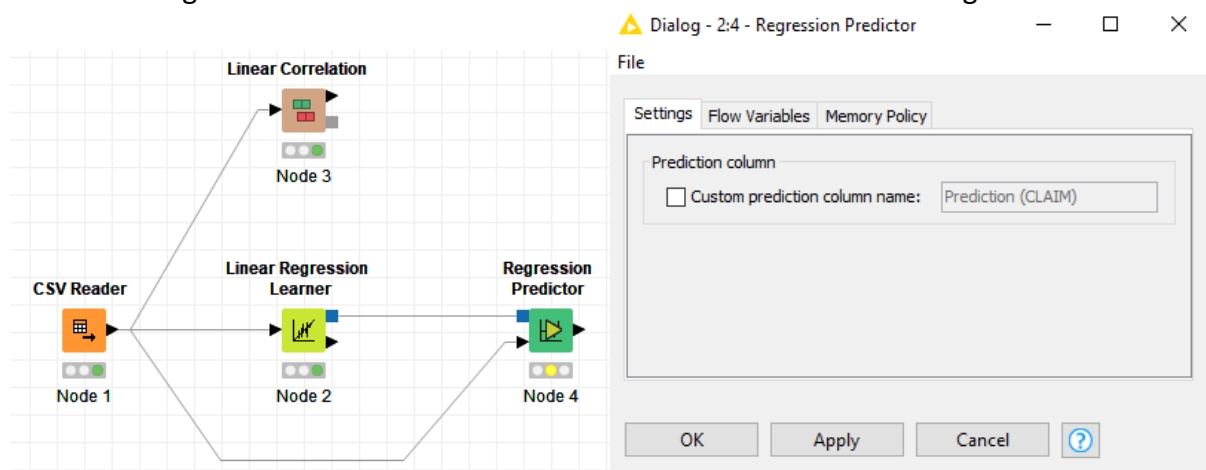
You can also use view the Correlation Matrix in a heat-map sort of visualization:



Points Poorly Fit by Model

We would like to detect errors or possible fraud by identifying cases that deviate substantially from the model. Even if these are not results of errors or fraud, they are inconsistent with the majority of cases and thus we should examine them closely.

8. Add a Regression Predictor node as shown to the workflow and configure it as such:



9. Execute the node and observe the "Prediction (CLAIM)" column.

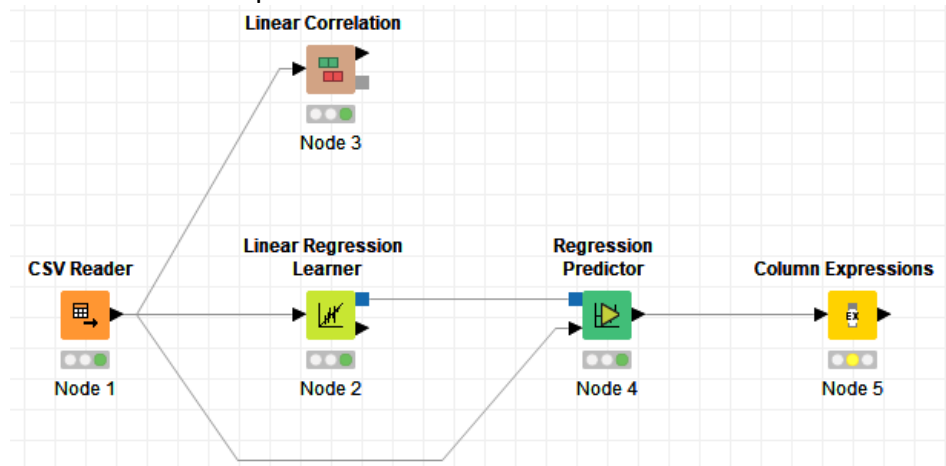
Predicted data - 2:4 - Regression Predictor

File Hilite Navigation View

Table "default" - Rows: 293 Spec - Columns: 5 Properties Flow Variables

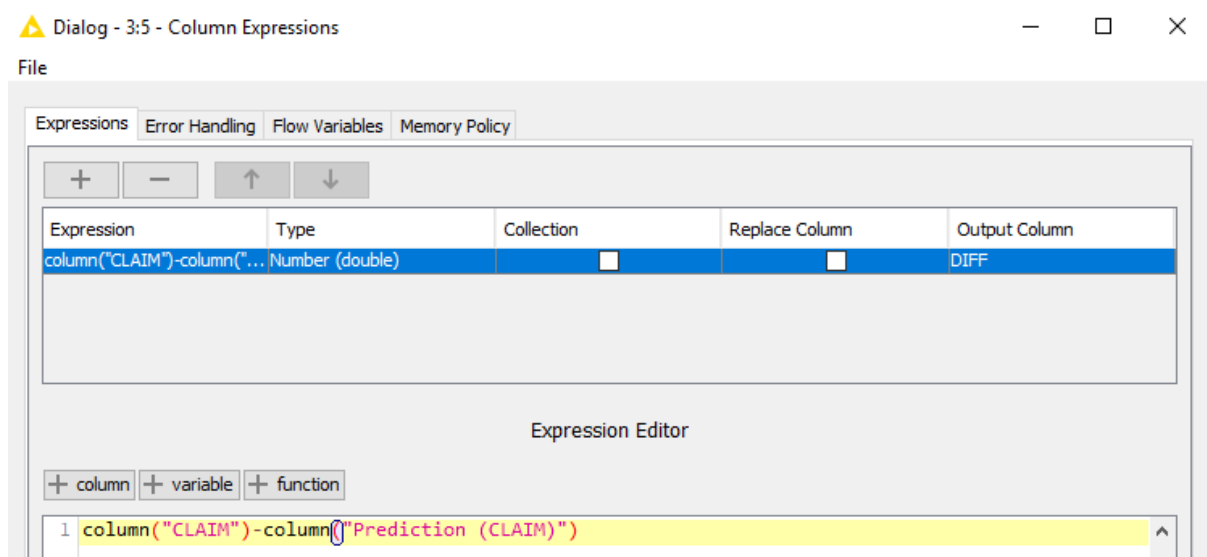
Row ID	ASG	AGE	LOS	CLAIM	Predict...
Row0	0	23	2	3,619	4,469.719
Row1	1	20	3	4,590.6	6,092.775
Row2	1	29	1	3,850	3,580.833
Row3	1	27	2	11,187.4	4,753.29
Row4	1	20	2	4,368	4,987.129
Row5	0	25	3	5,763.8	5,508.553
Row6	0	26	3	5,108.6	5,475.148
Row7	1	37	2	4,582.2	4,419.235
Row8	0	37	2	4,677.4	4,002.042
Row9	0	30	2	3,904.6	4,235.88
Row10	1	22	1	5,170.2	3,814.672
Row11	0	41	2	3,362.8	3,868.419

10. Let's use another KNIME node, *Column Expressions*, to calculate the difference between the actual and the predicted CLAIM values.



11. Configure the *Column Expressions* node as such using a simple formula:

column("CLAIM") – column("Prediction (CLAIM)")



12. Check the output data of the node and you will see that a new column "DIFF" has been created for you based on the formula you specified:

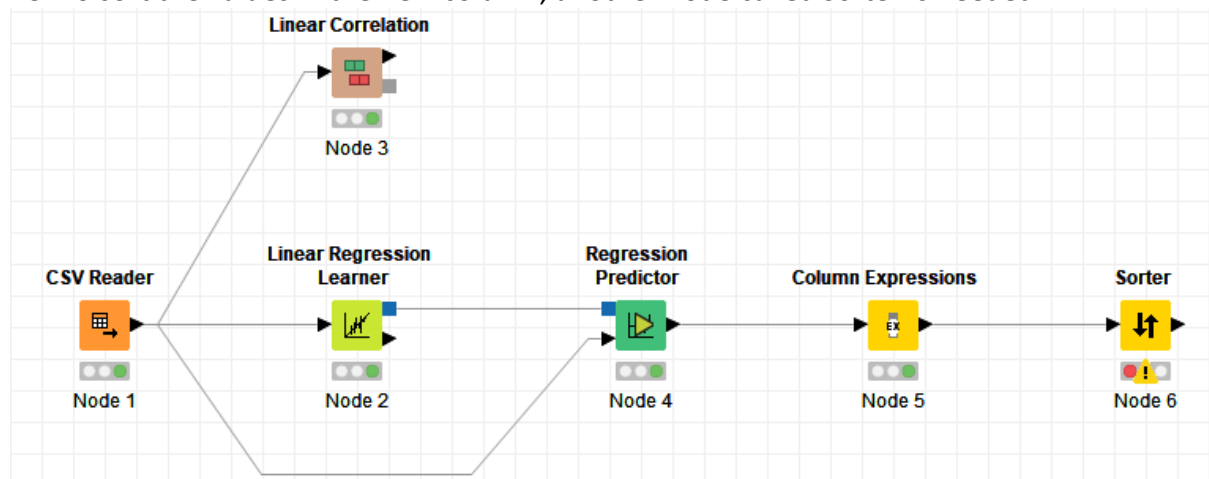
Output table - 3:5 - Column Expressions

File Hilite Navigation View

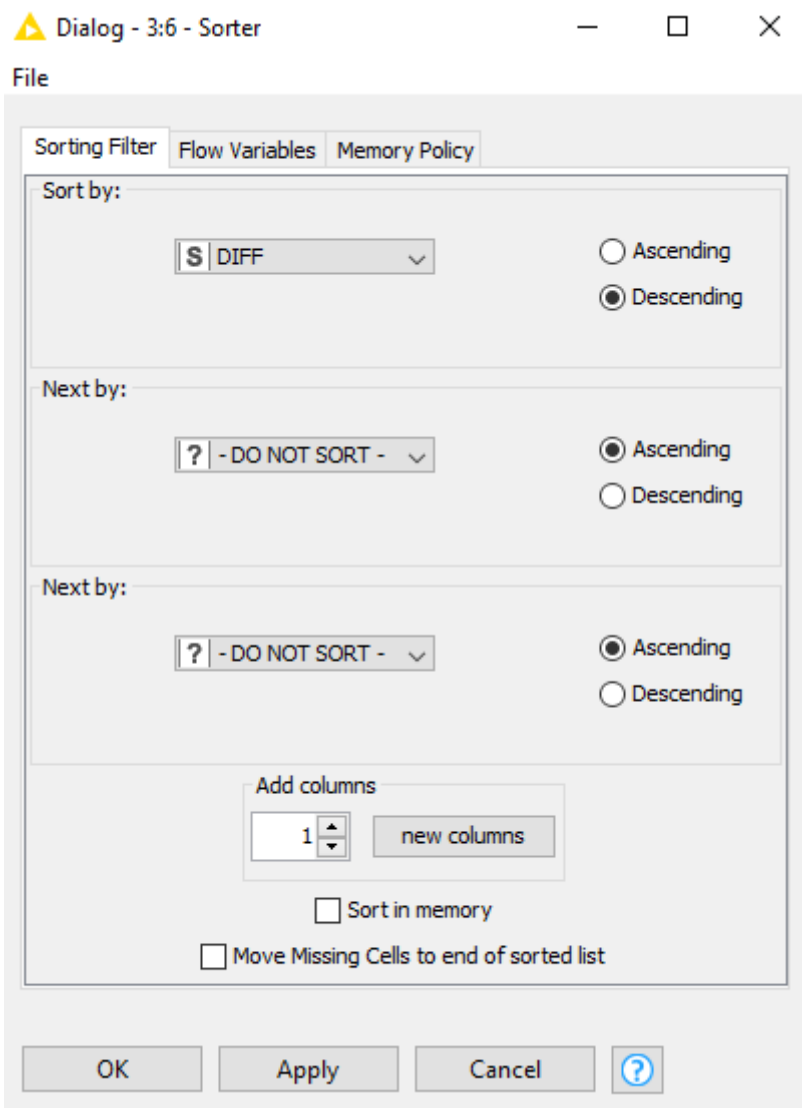
Table "default" - Rows: 293 Spec - Columns: 6 Properties Flow Variables

Row ID	I ASG	I AGE	I LOS	D CLAIM	D Predict...	S DIFF
Row0	0	23	2	3,619	4,469.719	-850.7186143453782
Row1	1	20	3	4,590.6	6,092.775	-1502.1746222536...
Row2	1	29	1	3,850	3,580.833	269.16668862102506
Row3	1	27	2	11,187.4	4,753.29	6434.109800300079
Row4	1	20	2	4,368	4,987.129	-619.1287476258603
Row5	0	25	3	5,763.8	5,508.553	255.24652471992795
Row6	0	26	3	5,108.6	5,475.148	-366.5479684335087
Row7	1	37	2	4,582.2	4,419.235	162.96486876570907


13. To sort the values in the new column, another node called *Sorter* is needed.



14. Configure the Sorter node as such:



15. Execute the node and look at the output data:

 Sorted Table - 3:6 - Sorter

File Hilite Navigation View

Table "default" - Rows: 293 Spec - Columns: 6 Properties Flow Variables						
Row ID	I ASG	I AGE	I LOS	D CLAIM	D Predict...	D DIFF
Row3	1	27	2	11,187.4	4,753.29	6,434.11
Row158	1	25	3	11,715.2	5,925.747	5,789.453
Row161	1	26	1	6,727	3,681.05	3,045.95
Row214	1	28	3	8,782.2	5,825.531	2,956.669
Row102	2	35	1	6,489	3,797.594	2,691.406
Row87	0	38	2	6,606.6	3,968.636	2,637.964
Row241	0	34	3	7,679	5,207.904	2,471.096
Row132	0	29	3	7,798	5,374.931	2,423.069
Row231	0	37	3	7,452.2	5,107.687	2,344.513

Notice that there are two records for which the claim values are much higher than the regression prediction. Both are about \$6,000 more than expected from the model. These claims should be examined more carefully.

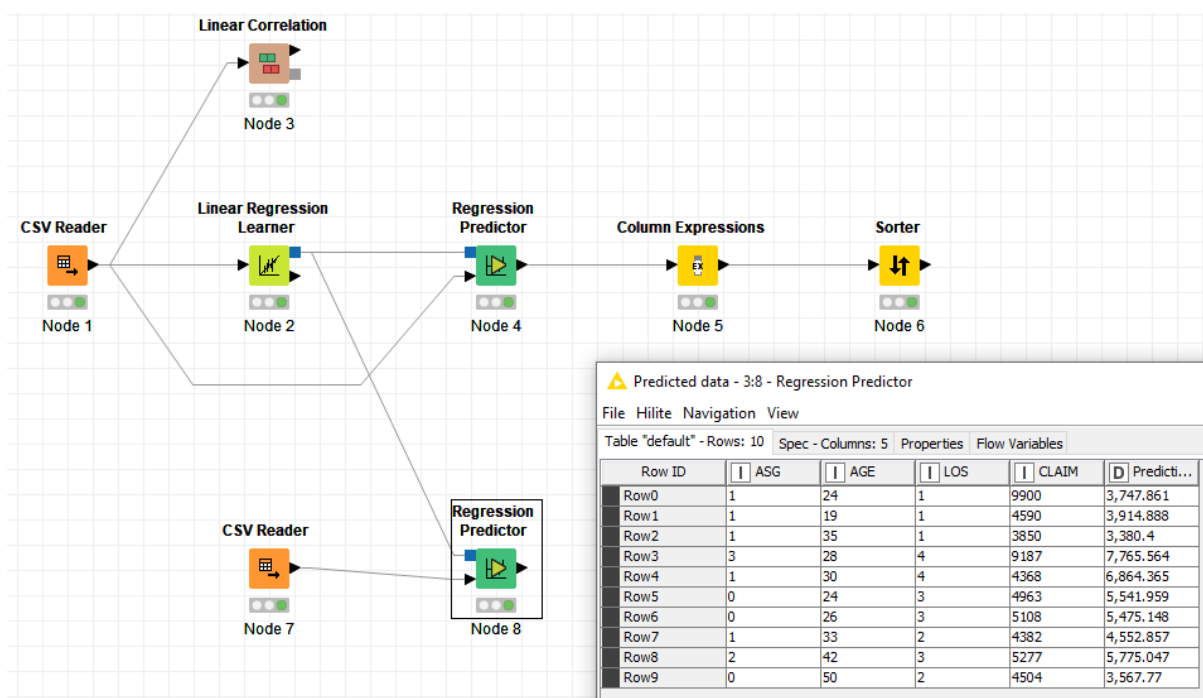
We could also examine the last few records for large over-predictions, unlikely to be fraud but might be errors.

Applying the model to new data

We are given the following details about some submission of claims:

ASG	AGE	LOS	CLAIM
1	24	1	9900
1	19	1	4590
1	35	1	3850
3	28	4	9187
1	30	4	4368
0	24	3	4963
0	26	3	5108
1	33	2	4382
2	42	3	5277
0	50	2	4504

We will import the new data into KNIME, and use the model that we have created to predict the claim values.



Based on the model built, Row0 is predicted to only claim \$3747.86, but \$9900 was actually claimed. More investigation may be warranted.

Conclusion

We have seen how to use KNIME's Regression Learner and Prediction nodes to build and apply a linear regression model to predict insurance claim payouts.