University of Minho
School of Engineering

Machine Learning and Decision-Making

ADI @ LEI/3º, MiEI/4º - 2º Semestre

Filipe Gonçalves, Inês Alves, Cesar Analide

Part IV – March 2022

Contents

- Linear Regression
- Hands On

Linear Regression

Linear Regression (LR) is used when we want to predict the value of a variable (independent variable) based on the value of another variable (dependent variable).

It helps to determine if an independent variable does a good job in predicting the dependent variable or which independent variable plays a significant role in predicting the dependent variable.

- **Dependent Variable:** target variable that will be estimated and predicted (y);
- **Independent Variable:** predictor variable that is used to estimate and predict (x);
- **Slope:** angle of the line, denoted as *m* or $\beta 1$;
- **Intercept:** where the function crosses the y-axis, denoted as *b* or $\beta 0$.

$$y = \beta 0 + \beta 1 X + \epsilon$$

Linear regression finds the best fit line through your data by searching for the regression coefficient ($\beta 1$) that minimizes the total error (ε) of the model.

Linear Regression

- Exercise:
  - **Problem:** Development of a Machine Learning Model able to predict the price of a given LEGO set
  - **Regression Approach:** Linear Regression approach to solve this problem
  - **Dataset:** Table with information regarding LEGO sets, containing:
    - 'age': Which age category it belongs to
    - 'list_price': Price of the set (in $)
    - 'num_reviews': Number of reviews per set
    - 'piece_count': Number of pieces in that LEGO set
    - 'play_star_ratings': Ratings
    - 'review_difficulty': difficulty level of the set
    - 'star_rating': Ratings
    - 'theme_name': Which theme it belongs
    - 'val_star_rating': Ratings
    - 'country': Country name

Linear Regression

CHECK OUT THE DATA

File Table - 3:1 - CSV Reader

File Edit Hilite Navigation View

Table "default" – Rows: 12261 | Spec – Columns: 11 | Properties | Flow Variables

Row ID	S ages	D list_pr...	I num_r...	I piece_...	D play_s...	I prod_id	S review...	D star_r...	S theme_...	D val_st...	S country
Row0	6-12	29.99	2	277	4	75823	Average	4.5	Angry Birds™	4	US
Row1	6-12	19.99	2	168	4	75822	Easy	5	Angry Birds™	4	US
Row2	6-12	12.99	11	74	4.3	75821	Easy	4.3	Angry Birds™	4.1	US
Row3	12+	99.99	23	1032	3.6	21030	Average	4.6	Architecture	4.3	US
Row4	12+	79.99	14	744	3.2	21035	Challenging	4.6	Architecture	4.1	US
Row5	12+	59.99	7	597	3.7	21039	Average	4.9	Architecture	4.4	US
Row6	12+	59.99	37	598	3.7	21028	Average	4.2	Architecture	4.1	US
Row7	12+	49.99	24	780	4.4	21029	Average	4.7	Architecture	4.3	US
Row8	12+	39.99	23	468	3.6	21034	Average	4.7	Architecture	4.1	US
Row9	12+	39.99	11	444	3.6	21033	Average	4.8	Architecture	4.5	US
Row10	12+	39.99	14	386	4.1	21036	Average	4.4	Architecture	3.6	US
Row11	12+	34.99	53	321	3.2	21019	Average	4.6	Architecture	4.4	US
Row12	12+	29.99	7	361	4.2	21032	Easy	4.6	Architecture	4.2	US
Row13	7-12	159.99	63	847	3.8	17101	Average	3.4	BOOST	3.5	US
Row14	10+	29.99	13	708	4.7	41597	Average	4.8	BrickHeadz	4.8	US
Row15	10+	19.99	1	234	3	41614	Easy	5	BrickHeadz	5	US
Row16	10+	19.99	1	160	5	41613	Very Easy	5	BrickHeadz	5	US
Row17	10+	9.99	1	149	2	41609	Very Easy	3	BrickHeadz	4	US
Row18	10+	9.99	1	141	2	41608	Very Easy	4	BrickHeadz	4	US
Row19	10+	9.99	3	101	4	41604	Average	4.7	BrickHeadz	4.5	US
Row20	10+	9.99	2	105	3	41605	Easy	5	BrickHeadz	5	US
Row21	10+	9.99	1	113	5	41606	Easy	5	BrickHeadz	5	US
Row22	10+	9.99	1	136	?	41607	?	5	BrickHeadz	?	US
Row23	10+	9.99	2	91	3	41485	Easy	4.5	BrickHeadz	4	US
Row24	10+	9.99	7	140	3.2	40270	Easy	4.9	BrickHeadz	4.7	US
Row25	10+	9.99	5	143	4.6	41599	Easy	5	BrickHeadz	5	US
Row26	10+	9.99	3	122	2.7	41598	Very Easy	4	BrickHeadz	3	US
Row27	10+	9.99	5	130	4.3	41603	Easy	5	BrickHeadz	4.8	US
Row28	10+	9.99	3	119	4.5	41602	Easy	5	BrickHeadz	4.5	US
Row29	10+	9.99	1	135	1	41600	Very Easy	4	BrickHeadz	3	US

Linear Regression

CHECK OUT THE DATA

Statistics View - 4:3 - Statistics

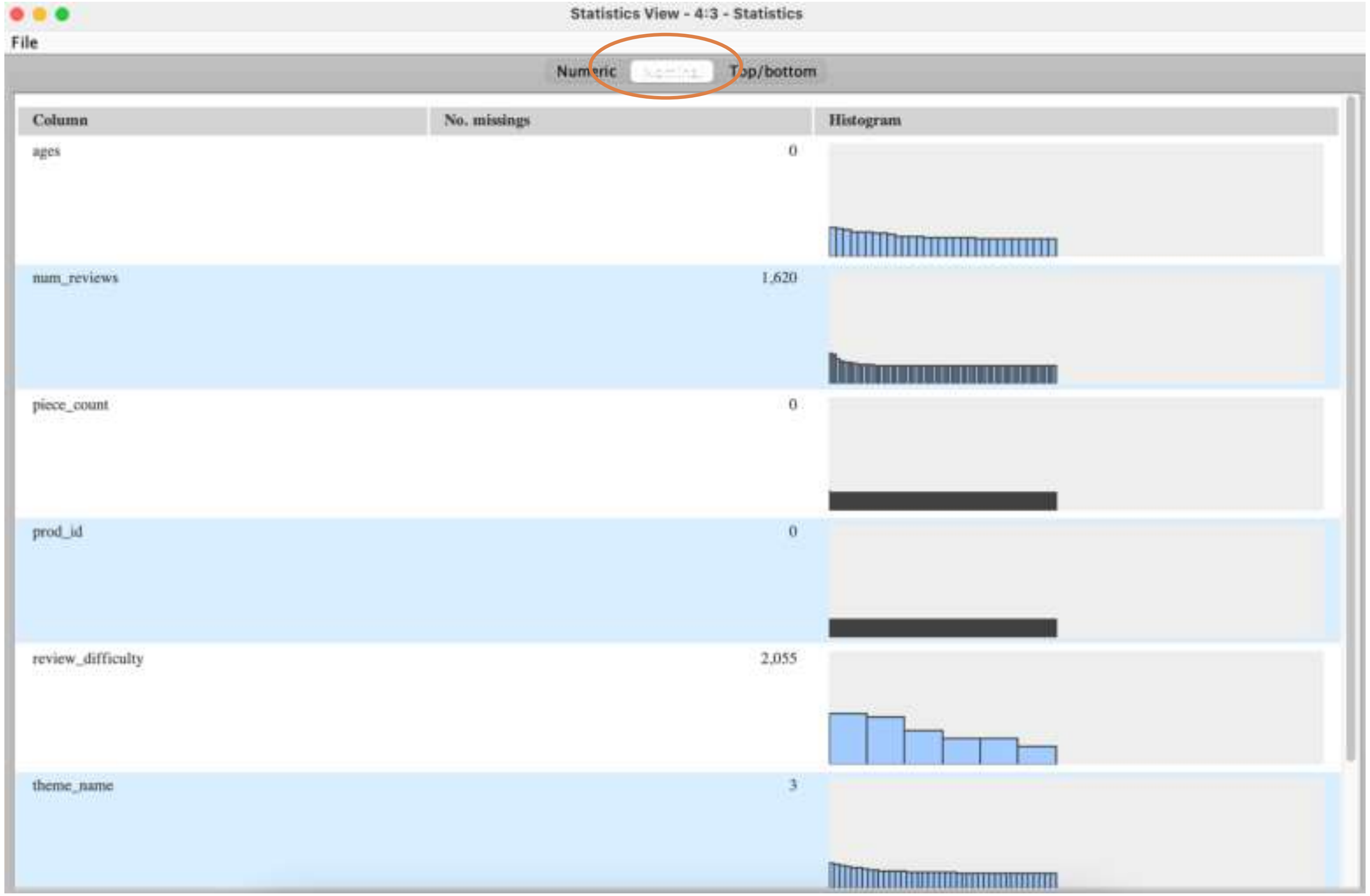
File

Nominal Top/bottom

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
list_price	2.2724	65.142	?	1,104.87	91.9804	4.6807	31.6525	0	0	0	2 1.10
num_reviews	1	16.8262	?	367	36.369	5.2629	35.535	1,620	0	0	1 36
piece_count	1	493.4059	?	7,541	825.3646	3.9681	19.8627	0	0	0	1 7.5
play_star_rating	1	4.3376	?	5	0.6521	-1.6728	3.9361	1,775	0	0	1
prod_id	630	59,836.7523	?	2,000,431	163,811.4523	11.4223	132.1878	0	0	0	630 2.000.431
star_rating	1.8	4.5141	?	5	0.5189	-1.6045	3.3166	1,620	0	0	2

Linear Regression

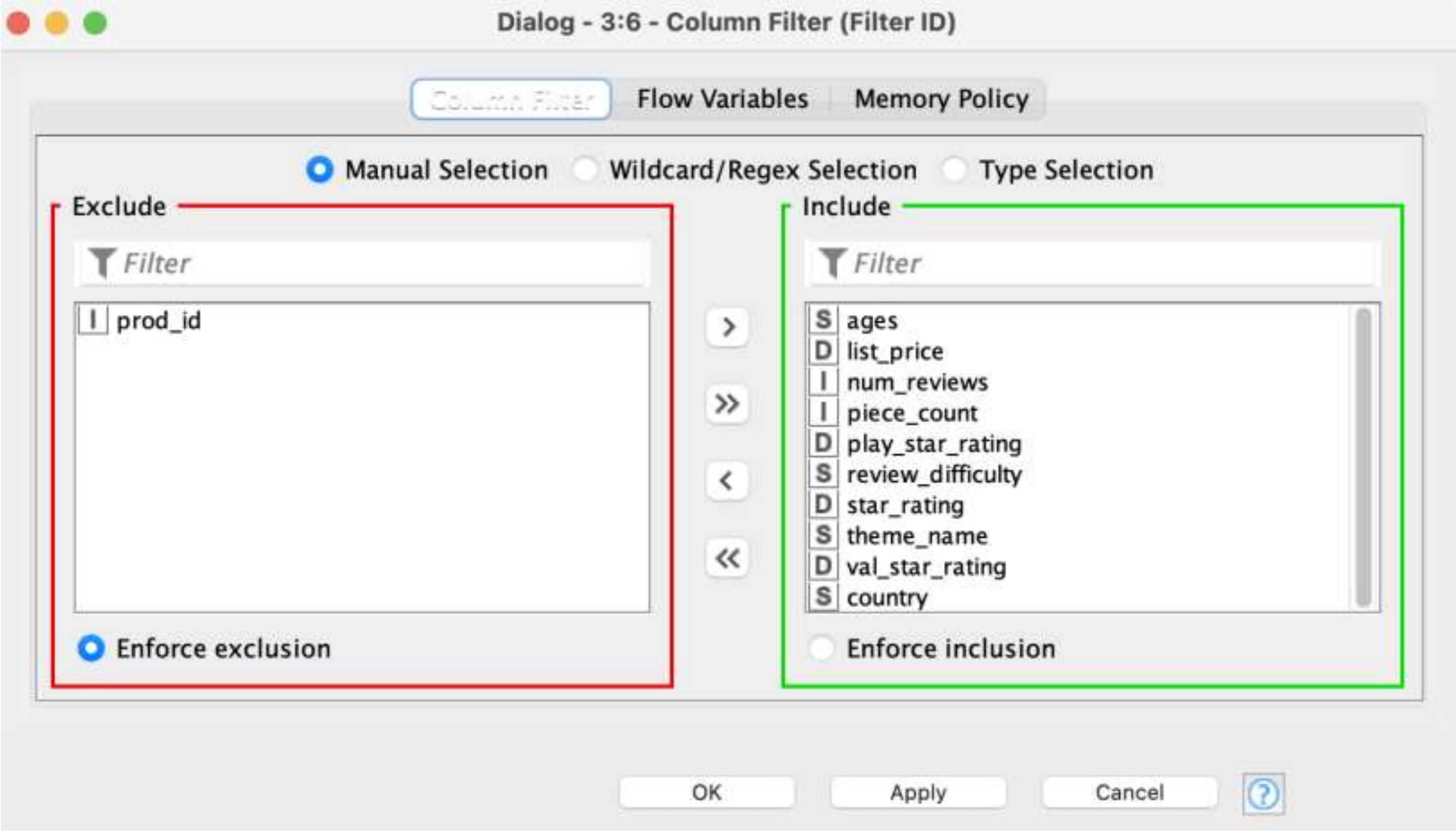
CHECK OUT THE DATA



Linear Regression

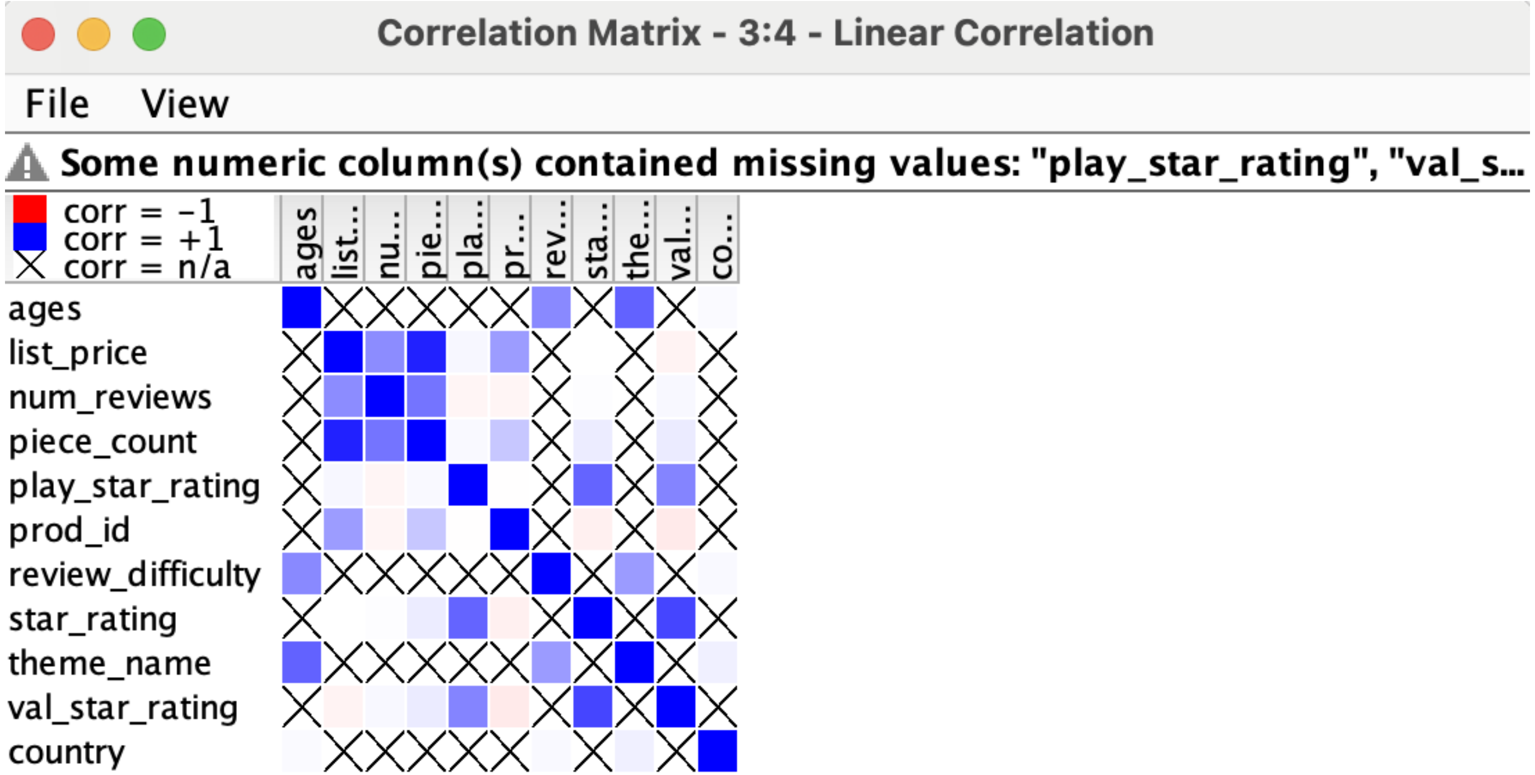
COLUMN FILTER

We can drop the product ID from our data



Linear Regression

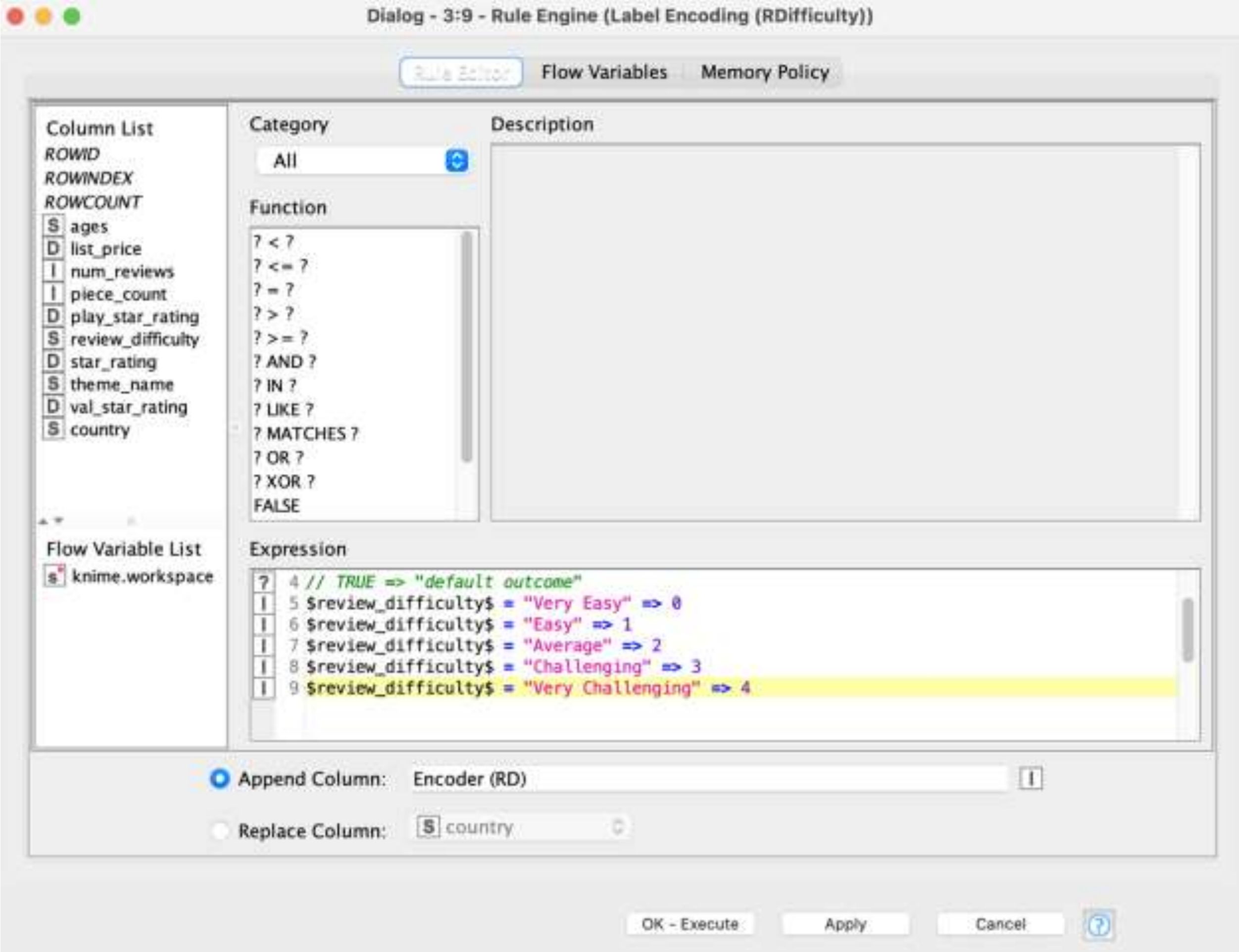
CORRELATION STUDY



Linear Regression

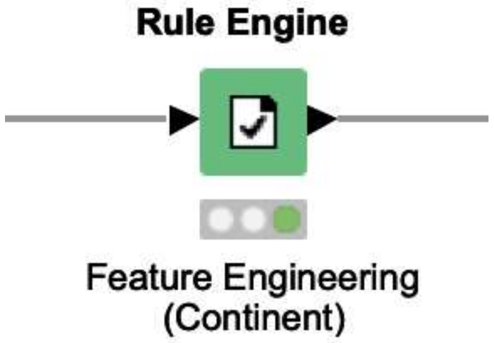
ENCODING

Label Encoding



- Can we do one-hot encoding to all our left columns? Does it make sense?
- Remember: encoding a huge number of categories has a very high cost…

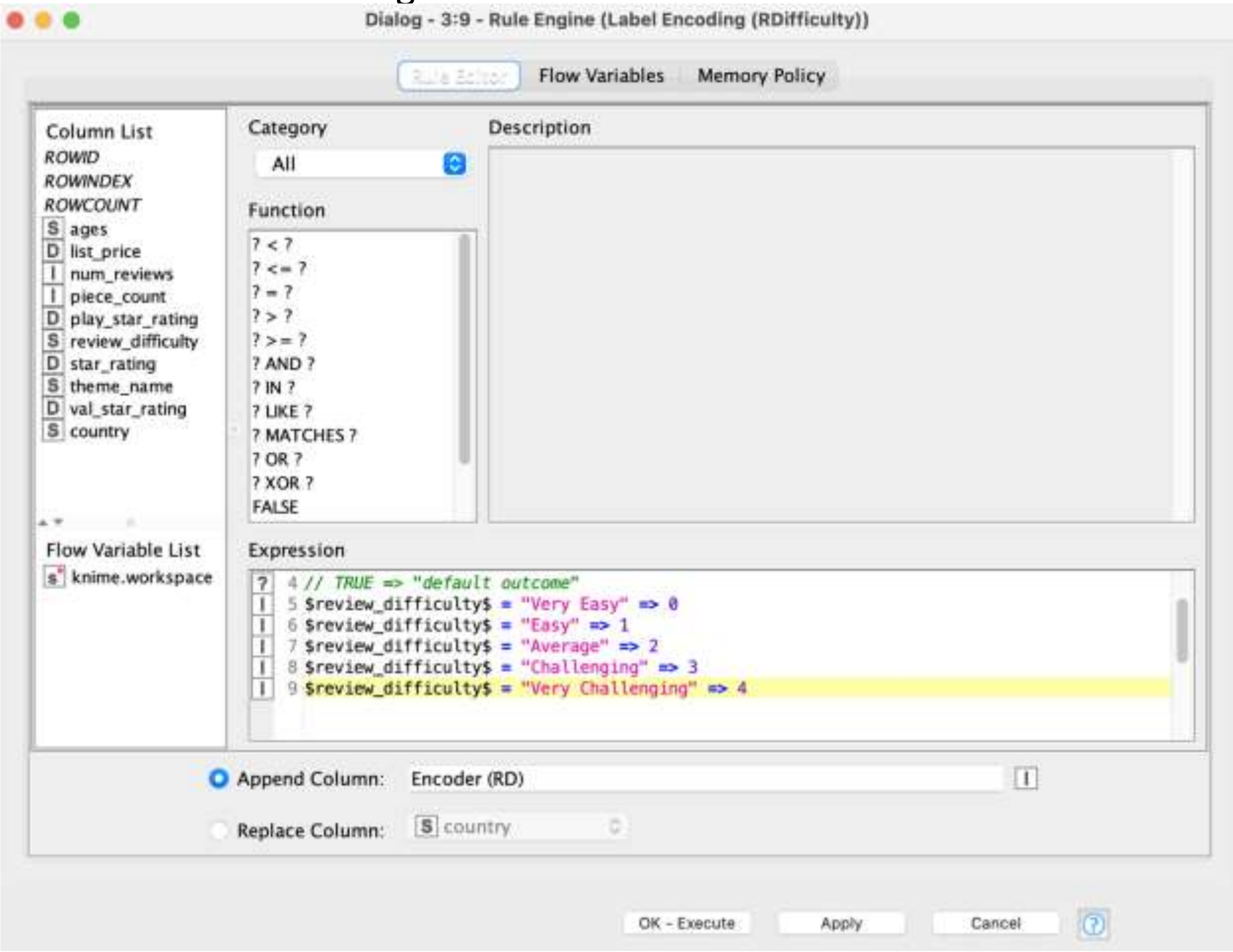
FEATURE ENGINEERING

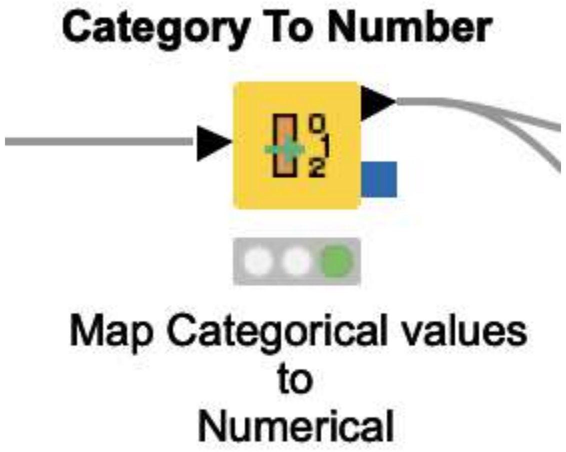


Linear Regression

ENCODING

Label Encoding





Linear Regression

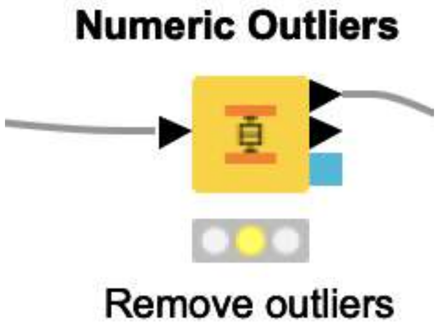
OUTLIERS

Now, our dataset is in a numerical format 👐

Next step: treat any **numeric** outliers that may exist

Remember

Outliers are extreme values in a feature that deviate from other observations on data. They need to be treated as they may have an effect on the statistics involved in the data.



Outlier Treatment

- Replace them
- Remove them

It mainly depends on the quantity of outliers.

Linear Regression

MISSING VALUES

We still have a lot of missing values in our data. What are we going to do about it?

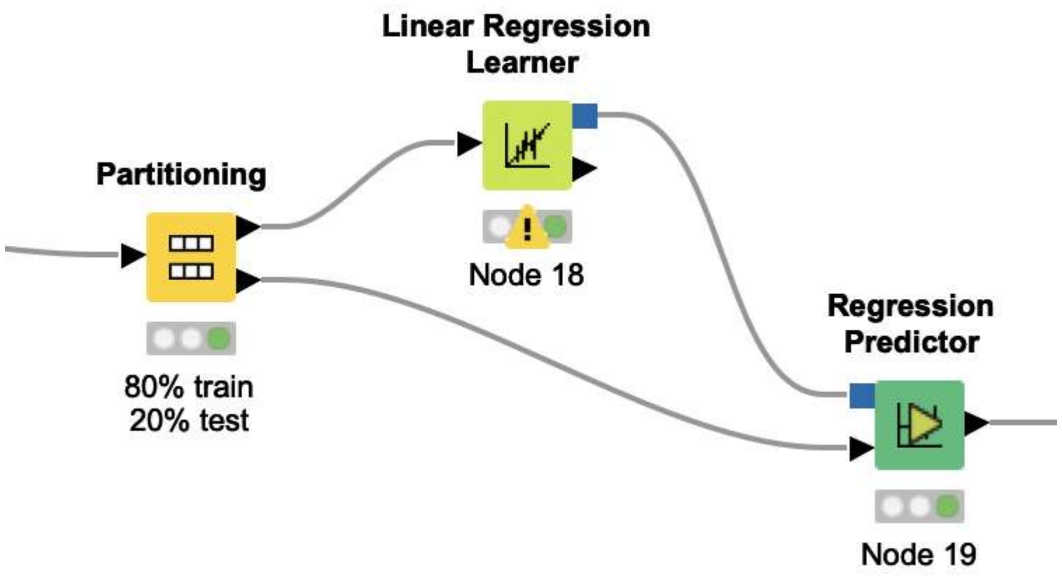
1) Mean
2) Median
3) Most used Value
4) Business knowledge and Knowledge extraction

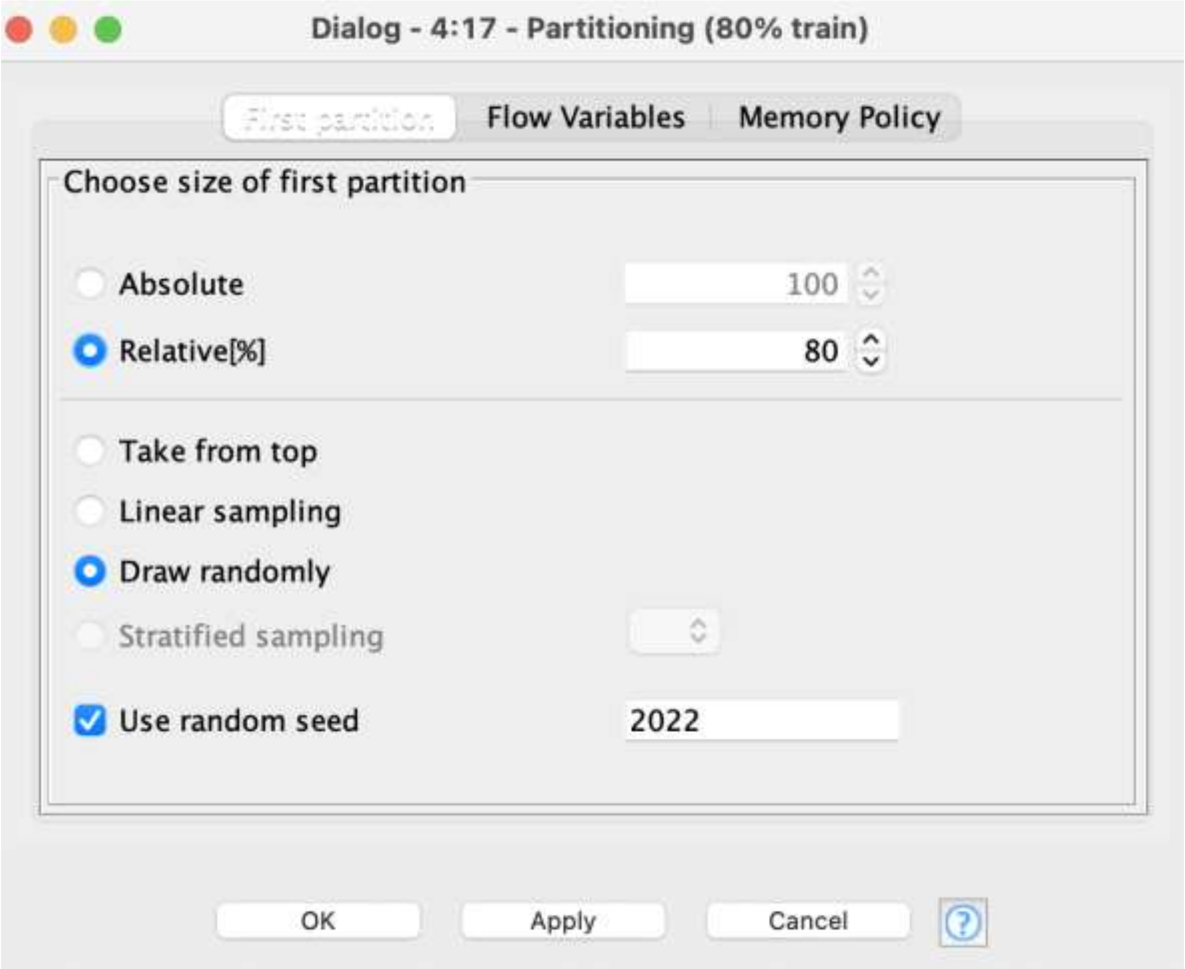
Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
list_price	2.2724	65.142	?	1,104.87	91.9804	4.6807	31.6525	0	0	0	
num_reviews	1	16.8262	?	367	36.369	5.2629	35.535	1,620	0	0	
piece_count	1	493.4059	?	7,541	825.3646	3.9681	19.8627	0	0	0	
play_star_rating	1	4.3376	?	5	0.6521	-1.6728	3.9361	1,775	0	0	
prod_id	630	59,836.7523	?	2,000,431	163,811.4523	11.4223	132.1878	0	0	0	
star_rating	1.8	4.5141	?	5	0.5189	-1.6045	3.3166	1,620	0	0	

Linear Regression

TRAINING A LINEAR REGRESSION MODEL

Label: LEGO set price

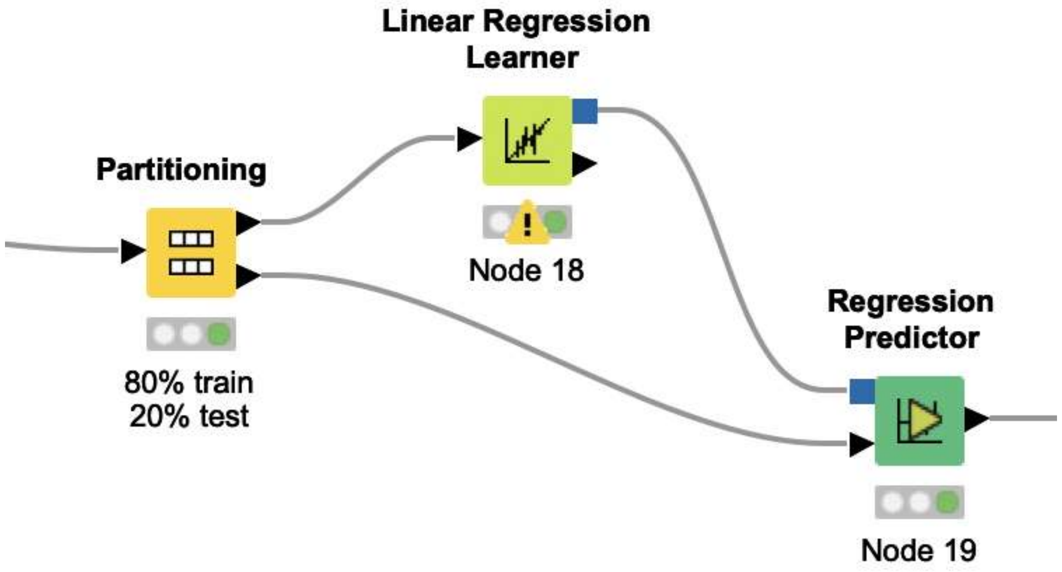


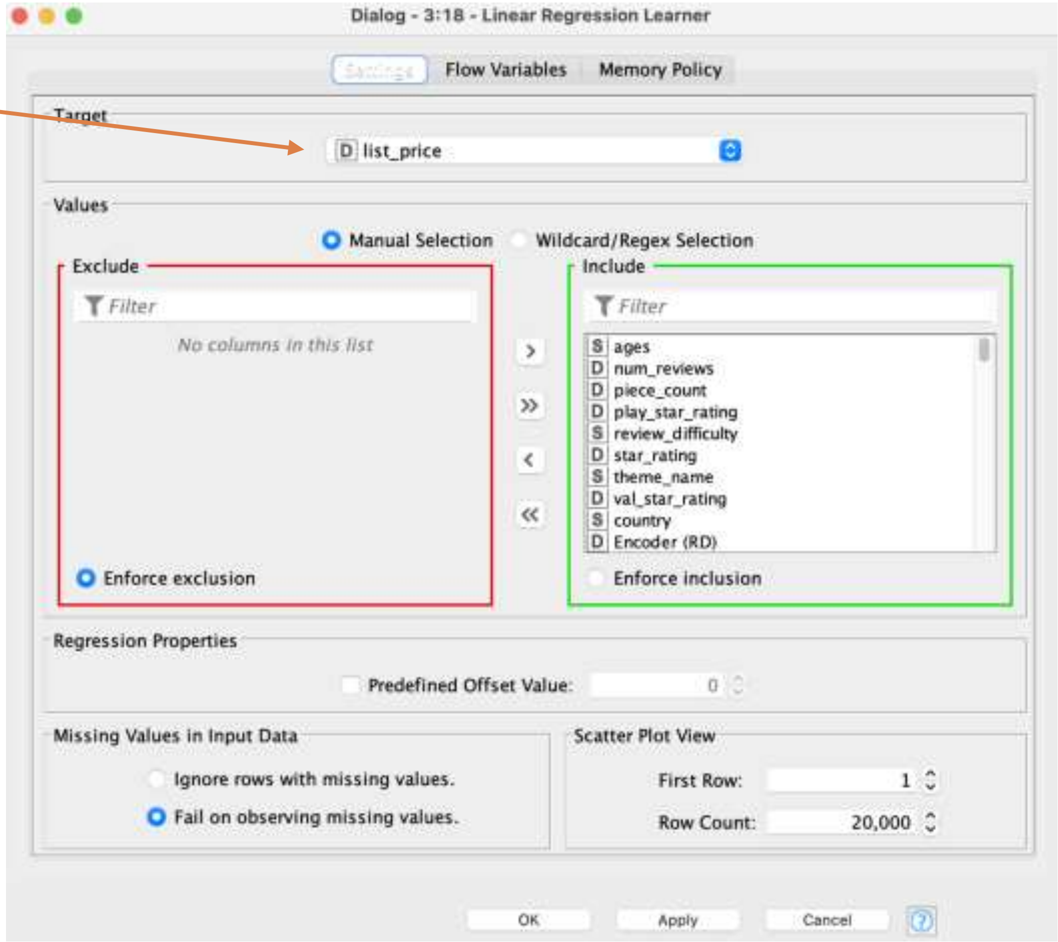


Linear Regression

TRAINING A LINEAR REGRESSION MODEL

Label: LEGO set price





Linear Regression

EVALUATING THE MODEL

1. R-Square value
2. Mean Absolute Error (MAE)
3. Mean Square Error (MSE)
4. Root Mean Square Error (RMSE)

R-Squared: proportion of variation in the outcome that is explained by the predictor variables.

$$\frac{\textit{total variance explained by model}}{\textit{total variance}}$$

Here, the higher the value, the better the model. 1 means that the model explains 100% of the variance of the labels. 0 means that the model doesn't understand how the labels vary.

MAE: mean of the absolute value of the errors $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

MSE: mean of the squared errors $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Linear Regression

EVALUATING THE MODEL

1. R-Square value
2. Mean Absolute Error (MAE)
3. Mean Square Error (MSE)
4. Root Mean Square Error (RMSE)

RMSE: calculates the average of the square roots of the error between values (actual) and predictions (hypotheses). It has a range from 0 to infinity and returns the magnitude of errors. The scores are negatively-oriented, so **lower values are better.** A score of 0 means that, on average, the predictions are great, that is, 100% effective.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Comparing the MAE, MSE and RMSE metrics:

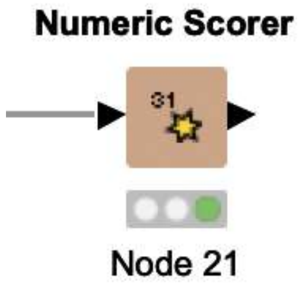
- **MAE** it's the easiest to understand because it's the average error;
- **MSE** it's most popular than MAE because MSE "punishes" larger errors, which tends to be useful in real world problems;
- **RMSE** is even more popular than MSE because RMSE is interpretable in the "y" units.

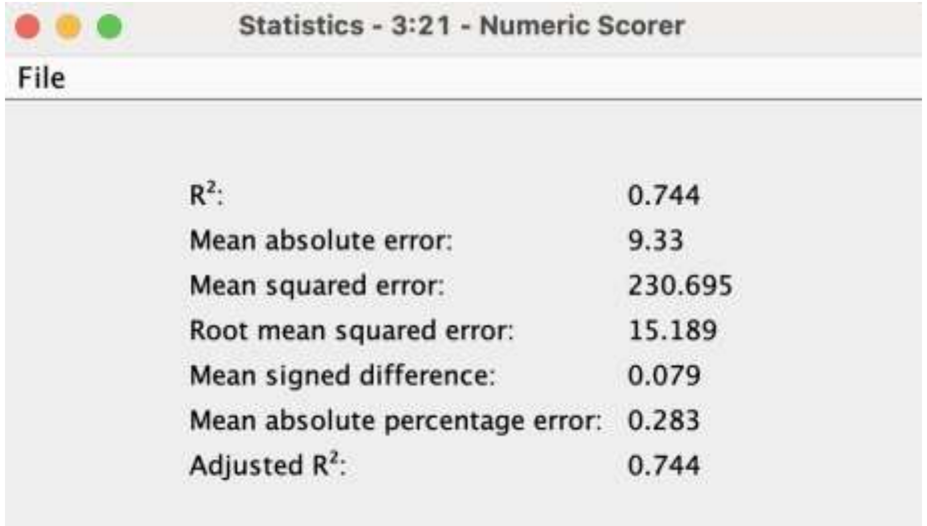
All of these are **loss functions**, so we want to minimize them.

Linear Regression

EVALUATING THE MODEL

1. R-Square value
2. Mean Absolute Error
3. Mean Square Error
4. Root Mean Square Error (RMSE)



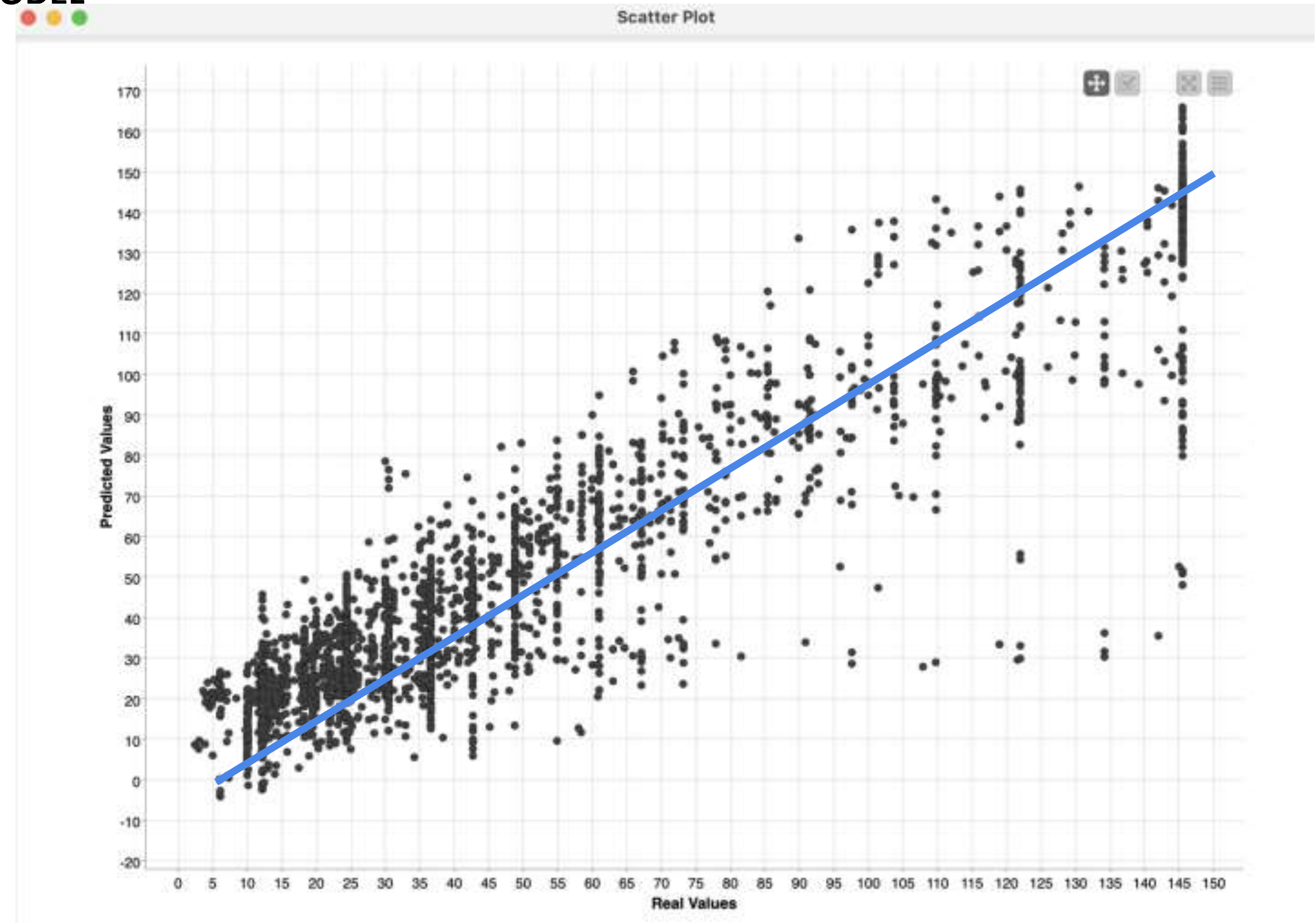


Knime's **Numeric Scorer** takes the predicted feature values and actual feature values as input and produces the metrics.

Our model has an R-square value of 74.4 % which means that 74.4% of our lego dataset falls around the regression line created by our model.

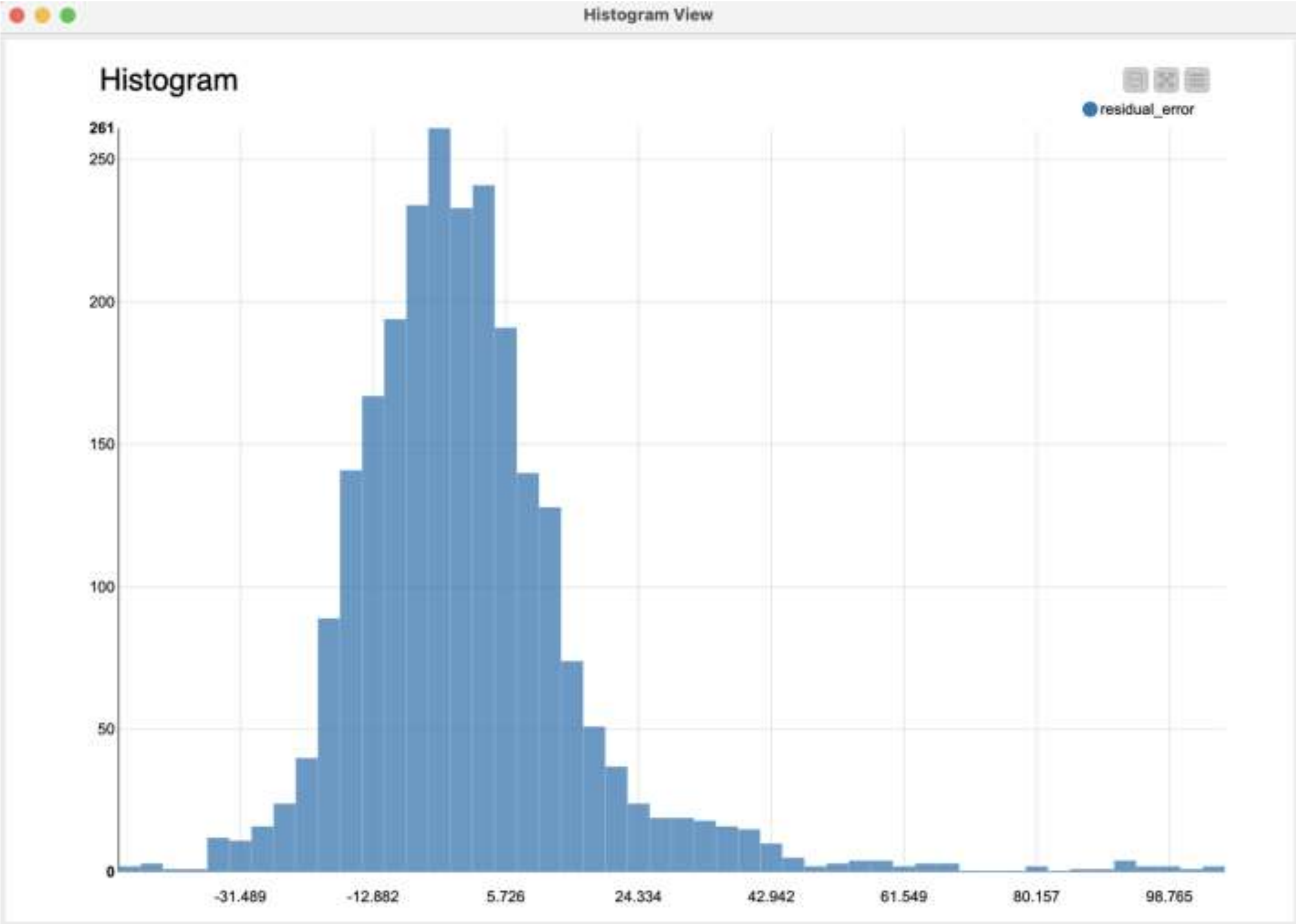
Linear Regression

EVALUATING THE MODEL



Linear Regression

EVALUATING THE MODEL



Linear Regression