AIML PROJECT

g

Real Estate Value Price Predictor using Knime Analytics Work flow.

DEPARTMENT OF ELECTRONICS AND COMMNICATION ENGINEERING 2023-24
Done by: 22ECB0F23- Rishit Shetty
Under Supervision of: Professor Dr.K.Ravi Kishore
Department of ECE NITW (2023)

ABSTRACT

The Real Estate Price Predictor is a comprehensive machine learning workflow designed to forecast property prices based on diverse features and data sources. This report provides an in-depth exploration of the workflow, methodologies employed, and insights gained through the development and evaluation of the predictive model.The workflow begins with data collection, encompassing a wide array of variables such as property characteristics, location-based features, economic indicators, and historical pricing trends. The dataset is preprocessed to handle missing values, outliers, and to ensure compatibility with machine learning algorithms.Feature engineering plays a crucial role in enhancing the model's predictive capabilities. Novel features, derived from existing variables, are created to capture intricate relationships within the data. Dimensionality reduction techniques are applied to streamline the feature set while retaining critical information.The report delves into the selection and fine-tuning of machine learning algorithms, including regression models, ensemble methods, and deep learning approaches. The model training process involves partitioning the dataset into training and testing sets, with rigorous validation procedures to prevent overfitting and ensure generalizability.The report also addresses challenges encountered during the workflow, such as data quality issues, model interpretability concerns, and potential biases. Strategies for mitigating these challenges are discussed to enhance the robustness and reliability of the predictive model.

REFRENCES:

Open for Innovation | KNIME:For node information.
Kaggle: Your Home for Data Science:For Database used for workflow.

BRIEF INTRODUCTION:

1.What is Knime: KNIME Analytics Platform is an open source software that allows users to access,
blend, analyze, and visualize data, without any coding. Its low-code, no-code interface offers an easy
introduction for beginners, and an advanced data science set of tools for experienced users.

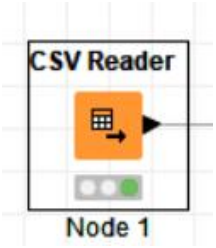
About the Dataset:

1. Number of Instances: 506
2. Number of Attributes: 13 continuous attributes (including "class"
   attribute "MEDV"), 1 binary-valued attribute.
3. Attribute Information:

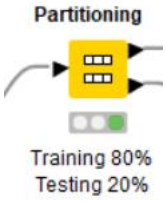
a) CRIM per capita crime rate by town
b) ZN proportion of residential land zoned for lots over 25,000 sq.ft.
c) INDUS proportion of non-retail business acres per town
d) CHAS Charles River dummy variable (= 1 if tract bounds
   river; 0 otherwise)
e) NOX nitric oxides concentration (parts per 10 million)

f) RM average number of rooms per dwelling
g) AGE proportion of owner-occupied units built prior to 1940
h) DIS weighted distances to five Boston employment centres
i) RAD index of accessibility to radial highways
j) TAX full-value property-tax rate per $10,000
k) PTRATIO pupil-teacher ratio by town
l) B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks
by town
m) LSTAT % lower status of the population
n) MEDV Median value of owner-occupied homes in $1000's

Nodes Used



● The csv module's reader and writer objects read and write sequences.
● Drag and drop the . csv file from the file system explorer to the workspace.
● The CSV Reader node will be created on the workspace automatically and it will be configured to read the dropped file. To read the file properly, the appropriate column
delimiter must be chosen in the configuration dialog.

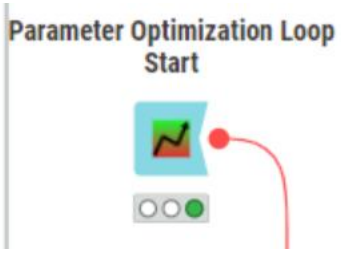


Input port

1. Type: Table
Table to partition

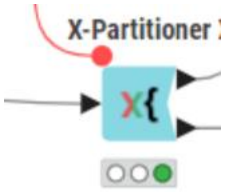
Output port

1. Type: Table
First partition (as defined in dialog)
First partition (as defined in dialog).
2. Type: Table
Second partition (remaining rows)
Second partition (remaining rows)



Output ports

Type: Flow Variable
Parameters
A parameter combination as flow variables
This loop starts a parameter optimization loop. In the dialog you can enter several parameters with an interval and a step size. The loop will vary these parameters following a certain search strategy. Each parameter is output as a flow variable. The parameters can then be used inside the loop body either directly or by converting them with a Variable to Table node into a data table.



Input ports

Type: Table

Any datatable

The datatable that is to be split

Output ports

Type: Table

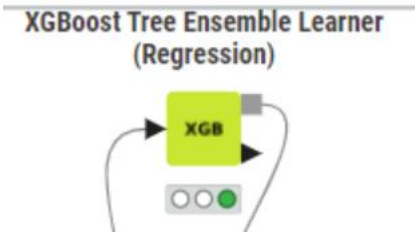
Training data

The data table with the training data

Type: Table

Test data

The data table with the test data



Input ports

Type: Table

Input Data

The data to learn from.

Output ports

Type: XGBoostModel

XGBoost Model

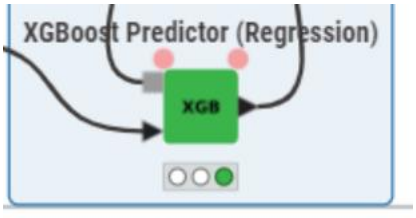
The trained model.

Type: Table

XGBoost Feature Importance

The feature importance measures for the training features. If the values are missing, then this indicates that the feature isn't used by the model at all.

I. Feature name column: The column containing feature names.
II. Weight column: The weight of a feature is the number of times a feature is used to split the data across all trees.
III. Gain column: The gain implies the average gain across all splits the feature is used in. A higher value of this metric when compared to another feature implies it is more important for generating a prediction.
IV. Cover column: The cover of a feature is the average coverage across all splits the feature is used in.
V. Total gain column: The total gain sums up the gain across all splits the feature is used in.
VI. Total cover column: The total cover sums up the total coverage across all splits the feature is used in.



Input ports

Type: XGBoostModel

XGBoost Model

The XGBoost model.

Type: Table

Input Data

The data to predict.

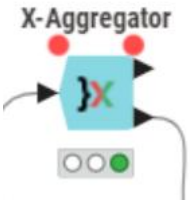
Output ports

Type: Table

Predicted Data

The data with the appended prediction.

.



Input ports

Type: Table

Outport from predictor

Contains the class column and the prediction column to compare

Output ports

Type: Table

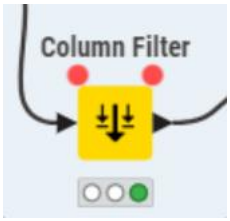
Prediction table

Collected output tables from the predictor

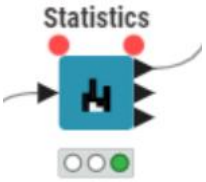
Type: Table

Error rates

Error rates for all iterations



The node offers 3 different filtering modes: manually, by name, and by type. - manually you decide which columns to keep and which to leave out, through Add and Remove buttons. - by name you decide which columns to keep based on their name through wild cards and Reg Ex. - by type you decide the columns to keep based on their type, like all Strings or all Integers.



Input ports

Type: Table

Table

Table from which to compute statistics.

Output ports

Type: Table

Statistics Table

Table with numeric values.

Type: Table

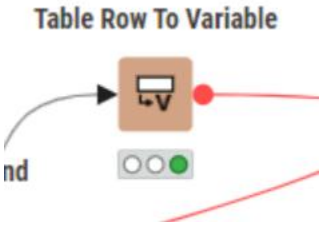
Nominal Histogram Table

Table with all nominal value histograms.

Type: Table

Occurrences Table

Table with all nominal values and their counts.



Input ports

Type: Table

Parameters table

The table whose first row will be converted to variables.

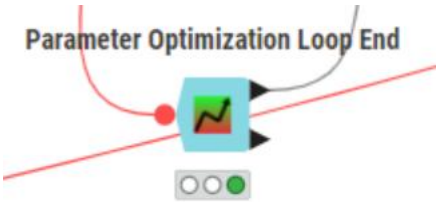
Output ports

Type: Flow Variable

Variables Connection

Holds created flow variables.

.



Input ports

Type: Flow Variable

Objective function value

A flowvariable that contains the objective function value

Output ports

Type: Table

Best parameters

Best Parameters found during the loop

Type: Table

All parameters

All tested parameter combinations in the loop

.

Input ports

Type: Table

Display data

Data table with data to display.

Output ports

Type: Image

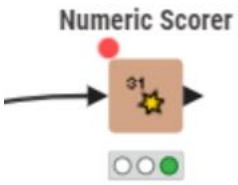
Image

SVG image rendered by the JavaScript implementation of the scatter plot.

Type: Table

Input data and view selection

Data table containing the input data appended with a column, that represents the selection made in the scatter plot view.



Input ports

Type: Table

Table

Table with predicted and reference numerical data

Output ports

Type: Table

Statistics

The computed statistical measures:

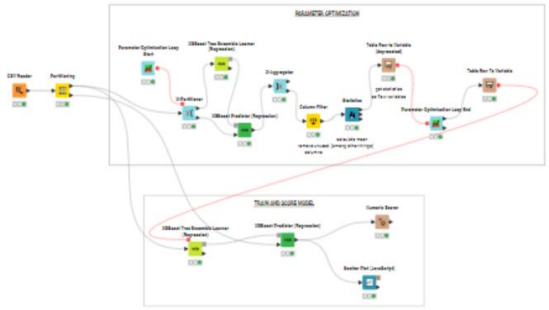
- R^2 - coefficient of determination: 1-SS_res/SS_tot
- Mean squared error: - $1/n*\Sigma((p_i-r_i)^2)$
- Mean absolute error: - $1/n*\Sigma|p_i-r_i|$
- Root mean squared error: - $Sqrt(1/n*\Sigma((p_i-r_i)^2))$
- Mean signed difference: - $1/n*\Sigma(p_i - r_i)$
- Mean absolute percentage error: $1/n * \Sigma((|r_i - p_i|)/|r_i|)$
- Adjusted R^2: $1-(1-R^2)(n-1)/(n-p-1)$

Knime Workflow :

Real Estate Value Price Predictor using XGBoost for Regression

This workflow shows how the XGBoost nodes can be used for regression tasks.
It also demonstrates a combination of parameter optimization with cross validation to find the optimal value for the number of boosting rounds.
The data used in this workflow is taken from https://www.kaggle.com/datasets/arslanali4343/real-estate-dataset

Workflow Requirements
KNIME Analytics Platform 3.7.0 or higher
KNIME XGBoost Integration
KNIME Optimization Extension
KNIME JavaScript Views



ADVANCED METRICS:

R^2:	0.836
Mean absolute error:	2.31
Mean squared error:	12.233
Root mean squared error:	3.498
Mean signed difference:	-0.565
Mean absolute percentage error:	0.113
Adjusted R^2:	0.836

Scatter Plot Graph