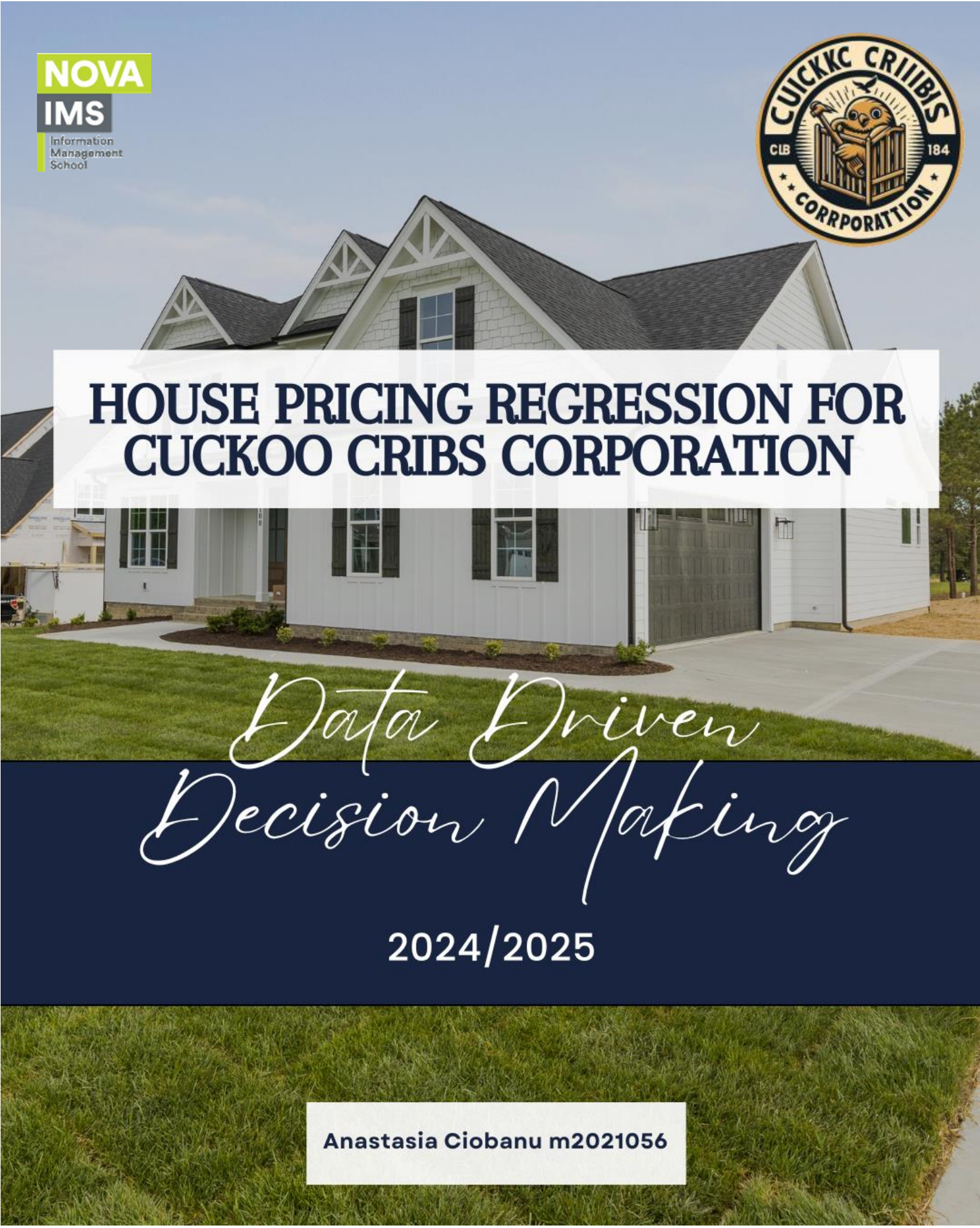


HOUSE PRICING REGRESSION FOR CUCKOO CRIBS CORPORATION

Data Driven Decision Making

2024/2025

Anastasia Ciobanu m2021056

Executive Summary

At Cuckoo Cribs, staying competitive in the real estate market depends on quickly spotting high-potential investments. To support this, a data-driven solution was developed to estimate the intrinsic value of residential properties, enabling early identification of undervalued listings.

Using KNIME Analytics Platform, a complete machine learning workflow was built to predict expected sales prices based on features such as construction year, area, renovation status, neighborhood and overall quality. This process included data cleaning, feature engineering, categorical encoding, outlier detection and, finally, model training.After testing several algorithms, an optimized XBoost Regressor was selected, achieving a strong performance on unseen data: an R^2 of 0,874, an RMSE of 28.294€ and a MAPE of 0,107, indicating high prediction accuracy.

This solution helps investment teams detect price anomalies, prioritize promising properties and make faster, data-driven decisions. Deliverables include a reusable PMML model file for easy integration and final predictions based on the latest dataset. Further sections detail the business context, modelling approach, evaluation and opportunities for ongoing improvement.

For illustration purposes, several visualisations and selected snippets of the model development workflow are included in the annex, offering a clear overview of key variables and methodological steps.

Available Data

The dataset provided by Cuckoo Cribs contains information on residential real estate transactions, where each property is described through It comprises 1.460 observations and 79 explanatory variables, spanning structural attributes, site conditions, neighbourhood location, and quality-related features. The target variable is *SalePrice*, representing the final transaction price of each house.

Data Types and Structure

The dataset included a variety of variable types, each of which added unique information about the housing market. Numerical features such as *LotArea, GrLivArea, GarageArea, and YearBuilt* offered continuous, quantifiable metrics fundamental to price estimation. Ordinal categorical variables like *ExterQual, BsmtQual*, and *KitchenQual* conveyed subjective quality assessments with a natural ranking, making them critical for capturing buyers' perceived value. In contrast, nominal categorical variables such as *MSZoning, Neighborhood, RoofStyle, and SaleType* provided segmentation without inherent order, allowing for the differentiation of property groups.

Additionally, binary indicators like *CentralAir* described the presence or absence of key comfort features, directly influencing the property's appeal and valuation. There was a comprehensive data understanding and cleaning phase to guarantee the data's dependability and integrity for predictive modelling.

As part of this, redundant variables were found and removed, inconsistent labels were standardized, and missing values were addressed. These pre-processing measures were essential for minimizing bias, lowering data noise, and creating a solid basis for modelling tasks that came after.

Feature Engineering

Many new features have been developed to improve model accuracy and interpretation, leveraging our knowledge and underlying data structure, to increase the added value of our consulting services:

Age: This variable is used to estimate the age of the property at the time of sale. It was calculated using the formula: *Age = YrSold - YearBuilt*

YearsSinceRemodel: This variable will indicate how many years have passed since the last remodel up to the year of sale. It helps assess whether a recent remodel might impact house price.

TotalHouseSF*:* Combines the surface areas of all usable spaces (1st & 2nd floor, basement, garage) to reflect the full size of the property. This composite variable is more informative than using each area individually.

TotalBathrooms*:* Summarises all bathrooms (full and half) in the house and basement. Weighting half baths acknowledges their lesser utility while preserving their contribution to price, using the formula: *TotalBathrooms = FullBath + BsmtFullBath + 0.5 * (HalfBath + BsmtHalfBath)*

RoomDensity*:* Captures how spacious or crowded a house feels by relating total rooms to living space, offering a measure of internal layout efficiency and is calculated as follows*: RoomDensity = (TotRmsAbvGrd / GrLivArea)*.

QualityIndex*:* Aggregates multiple quality-related ordinal features into a single numeric score. By converting *ExterQual, KitchenQual, BsmtQual* (and others) into numerical scales and summing them, we encapsulate overall construction and finish quality in one robust feature.

TotalPorchSF: This feature was engineered to represent the total porch surface area of the property by summing all types of porch-related spaces. It was calculated as: *TotalPorchSF = OpenPorchSF + EnclosedPorch + 3SsnPorch + ScreenPorch*

HasGarage, HasPool, HasBasement, HasFence, HasAlley, and ***IsRemodeled***: These binary flags were created using logic rules to indicate the presence or absence of specific features. Their simplicity improves interpretability and helps models focus on relevant distinctions.

This feature engineering ensures that critical information is captured in a compact, meaningful way, avoiding fragmentation and improving generalisation.

Data Preparation and Encoding

During the data preparation phase, particular focus was placed on columns with "NA" values that KNIME did not identify as missing values. These were prevalent in both numerical and nominal categorical variables. For numerical columns such as *GarageYrBlt, LotFrontage*, and *MasVnrArea*, a String Manipulation node was applied to replace "NA" text entries with zero, ensuring these could later be correctly interpreted as missing values and handled through the Missing Value node. There are 14 nominal categorical variables for which the node Column Expressions were used to replace "NA" strings with meaningful labels, such as *BsmtQual, BsmtCond,* the “NA” was replaced to "NoBasement", aligning with domain understanding. After cleaning, all 10 ordinal quality variables (e.g., *ExterQual, KitchenQual*) were converted from text levels (Ex, Gd, TA, FA and PO) to a numerical

scale (5 to 1) using Column Expressions to preserve their ordinal nature. In order to treat these new values as numerical features during model training, a String to Number node was applied because they were still stored as strings.

This pipeline improves the consistency and dependability of the dataset for predictive modelling by guaranteeing precise identification, conversion, and handling of missing and ordinal data.

Following the initial feature engineering and column filtering steps, the dataset was partitioned into training and testing sets using a 70/30 split. A Missing Value node was used to impute missing entries at the start of the structured transformation pipeline for the training data.

A One to Many (PMML) node was applied to encode nominal categorical variables through one-hot encoding, and a PMML Writer node stored the transformation logic. The test dataset was subjected to the same one-hot encoding logic and missing value application by PMML Reader and PMML Transformation Apply nodes in order to guarantee consistency between training and testing data. For both sets, Numerical Outliers and the associated Apply nodes were used to identify and manage numerical outliers.

The Normalizer node was then used for training data and the same scaling logic was applied to test data with the Normalizer Apply node. Finally, both streams were merged by the Joiner to bring together the datasets for the next modelling step. This pipeline ensures consistent and reproducible data transformations across both training and test partitions, supporting robust and reliable model training.

Model Details & Performance

To ensure our recommendations are both accurate and easily justifiable to the Cuckoo Cribs board, we rigorously tested three regression models: **Linear Regression**, **Random Forest**, and **XGBoost**—each offering a different balance between simplicity, interpretability, and predictive strength.

Model	R^2 Score	RMSE (Root Mean Squared Error)	MAPE (Mean Absolute Percentage Error)
Linear Regression	0,784	37.095 EUR	14,8%
Random Forest	0,773	38.038 EUR	13,3%
XGBoost	0,874	28.294 EUR	10,7%

Table 1. Results of the R^2 Score, RMSE and MAPE of each model

Model 1: Linear Regression

Linear Regression is a traditional and highly interpretable model. It provides a straightforward relationship between each input feature and the sale price, making it appealing to decision-makers who prefer transparency. In this case, the model achieved an **R^2 of 0,784**, which means the model explains 78,4% of the variation in property prices based on the selected predictors. However, the **RMSE of 37.095 EUR** indicates that its price estimates, on average, deviate by ~37.000 EUR from the true value. The **MAPE of 14,8%** suggests that predictions are, on average, 14,8% off from the actual selling price. Although acceptable, this margin could be improved.

Model 2: Random Forest

Based on the concept of creating several decision trees and averaging their predictions, the Random Forest model is an ensemble learning technique. This method works very well for complex datasets like real estate because it enhances robustness, decreases overfitting, and captures non-linear relationships and interactions among variables.

In our project, Random Forest achieved an **R^2 of 0,773**, meaning it explains 77% of the variance in house prices. The **Root Mean Squared Error (RMSE)**, a measure of the average deviation between predicted and actual prices, was **38.038 EUR**, showing it performs worse that the linear regression, likely because the relationship between the target and the features is mostly linear - causing the Random Forest to overcomplicate the model and underperform as a result. Additionally, the **Mean Absolute Percentage Error (MAPE)** was **13,3%**, indicating that predictions, on average, deviate about 13% from the true price, which is better than the previous model, but still could be improved.

Model 3: XGBoost (Extreme Gradient Boosting)

XGBoost (Extreme Gradient Boosting) emerged as the top-performing model in our predictive analysis. As a cutting-edge machine learning algorithm, XGBoost operates by building trees in sequence, with each new tree focused on correcting the prediction errors of the previous one. This iterative learning process enables the model to achieve high levels of precision and efficiency, especially when working with structured data such as real estate property attributes.

In the context of our study, XGBoost clearly outperformed the other models evaluated. It achieved the highest **R^2 score of 0,874**, indicating that it was able to explain 87,4% of the variability in house prices, demonstrating strong predictive power. Moreover, it recorded the **lowest Root Mean Squared Error (RMSE) at 28.294 EUR**, which reflects the smallest average deviation between predicted and actual prices. The **Mean Absolute Percentage Error (MAPE)** was also competitive at **10,7%**, signifying that, on average, the model's predictions were within 10,7% of the true sale price.

Out of the models tested, XGBoost proved to be the most appropriate for Cuckoo Cribs' objective of obtaining accurate, data-driven property valuations. Despite being straightforward and simple to understand, linear regression outperformed Random Forest in this case - likely due to the predominantly linear relationship in the data. Random Forest, although its ability to model complex interactions, may have overcomplicated the problem and, consequently, it underperformed. XGBoost, however, delivered the best results in terms of precision and optimization.

In our evaluation, XGBoost delivered the highest R^2 score (0,874), the lowest RMSE (28.294 EUR), and the lowest MAPE (0,107) demonstrating consistently strong generalisation. It has an advantage in structured data environments such as real estate because it can handle missing values natively, iteratively correct prediction errors, and avoid overfitting through built-in regularization. Its outputs, like feature importance and price predictions, are easily converted into useful business insights, even though the internal workings are more complicated.

As a result, XGBoost was chosen for its strategic alignment with Cuckoo Cribs' goal of becoming a leader in intelligent, evidence-based pricing in addition to its exceptional accuracy.

Business Outcome

With the implementation of the XGBoost model, Cuckoo Cribs has a reliable and expandable tool for making real estate investment decisions, signalling a substantial shift from intuition-driven to data-driven valuation. The model provides extremely accurate real-time price estimates by ingesting a small but potent set of property attributes, including location, quality indicators, and total surface area. This enables quicker and more assured investment decisions.

This predictive ability enables the acquisition team to spot undervalued or overvalued real estate assets at a glance. If, for example, a property is quoted for 270.000€ but the model estimates its fair market value at 310.000€, this gives a potential margin of 40.000€, an opportunity that might otherwise not be taken up by traditional methods. If applied systematically to multiple listings, such insights may lead to substantial financial gains through reclassification, optimizing rental income, or strategic acquisitions.

Additionally, operational efficiency results from the automation of property screening. Previously, subjective and time-consuming assessments that frequently relied on Excel spreadsheets or gut feelings could be replaced with immediate, objective forecasts. This can cut down on the amount of time spent on each property analysis by as much as 80%, freeing up the team to concentrate on deal execution and negotiation instead of manual valuation. Furthermore, having access to clear, data-supported appraisals gives Cuckoo Cribs more negotiating power and a solid foundation on which to discuss prices with brokers and sellers.

The model learns and adjusts continuously as it is exposed to more data over time, taking into account factors like seasonal effects, new market dynamics, or even variables like rental profitability or renovation costs. This makes it possible for Cuckoo Cribs to stay ahead of the competition in a market that is extremely competitive by enabling future improvements like automated alerts for high-potential listings or live monitoring dashboards.

Ultimately, this project not only increases pricing precision but also lessens human bias, speeds up decision-making, and enhances Cuckoo Cribs' standing as a cutting-edge, data-driven real estate company. Predictive analytics strategically incorporated into day-to-day operations enable the business to scale effectively, maximize long-term returns, and make more informed investment decisions.

Deployment

This model could be deployed through a web-based dashboard used internally by real estate agencies, built with tools like Streamlit, Flask, or Dash. The platform would allow agents to input property features such as location, usable area, number of rooms, and deposit type and instantly receive a predicted price or classification.

During client meetings, real estate agents could use the model to support property valuation and provide quick, data-driven insights into pricing. For both buyers and sellers, this would assist in establishing reasonable price expectations. Additionally, the model might help determine whether a property is overpriced or undervalued by using market trends discovered from the training data.

Furthermore, the model could be incorporated into platforms for property listings or CRM systems (e.g. A. both Imovirtual Pro and CASAFARI. The system may automatically recommend price ranges based on the model's predictions when new listings are added. The model would get better over time

with constant retraining using new market data, becoming more precise and in line with current market trends, which would increase agent productivity and customer satisfaction.

Monitoring & Retraining

As market conditions change, a regular monitoring and retraining strategy should be implemented to guarantee the predictive model stays accurate and applicable. Cuckoo Cribs ought to use current real estate sales data to assess model performance every month or every three months.
Metrics like RMSE, MAPE and R^2 can be tracked over time to identify any changes in the relationship between features and the target variable or a decline in predictive performance.

A discernible rise in error or a decrease in feature importance may be a sign that the market dynamics have changed as a result of factors like changing interest rates, construction costs, or regional demand. In these situations, it is advised to retrain the model once or twice a year, though there is room to do so more frequently during times of significant market shift.

In order to ensure that new data is integrated while preserving the current preprocessing and feature engineering logic, this retraining procedure can be automated within KNIME.
Cuckoo Cribs can sustain high predictive accuracy and market alignment by continuously monitoring and retraining the model, guaranteeing that agents and clients will continue to gain from trustworthy, data-driven insights.

Improvements & Next Steps

While the current predictive model already provides strong performance and tangible business value, its accuracy, scalability, and strategic relevance for Cuckoo Cribs could all be improved with a few tweaks.

One of the most impactful next steps would be to expand the dataset with additional property listings and actual sales transactions. A larger and more current sample would enhance the model's ability to generalize and better represent changing market conditions. Furthermore, incorporating data from outside sources, like infrastructure advancements, interest rate changes, and macroeconomic trends, would enhance context and enable more accurate pricing forecasts.

Another interesting approach is featuring enhancement. It may be possible to uncover hidden value indicators that aren't currently represented in numerical features by including renovation costs, rental yield estimates, or by using Natural Language Processing (NLP) to extract insights from unstructured data, such as listing descriptions and agent notes.
In terms of modelling, investigating more sophisticated methods, like ensemble stacking or deep learning, may improve accuracy, particularly for high-value or unusual properties. Over time, as more data becomes available, these strategies might prove especially successful.

Ultimately, the model's integration into a mobile app, web-based interface, or real-time alert system would facilitate quicker decision-making. For instance, the acquisition team could be notified immediately upon the listing of properties with high potential or low value, giving them a competitive edge in the market.

When combined, these upcoming enhancements would strengthen the model's position as a flexible, intelligent instrument at the heart of Cuckoo Cribs' data-driven investment approach, maximising profits while lowering risk and labour costs.

Annexes

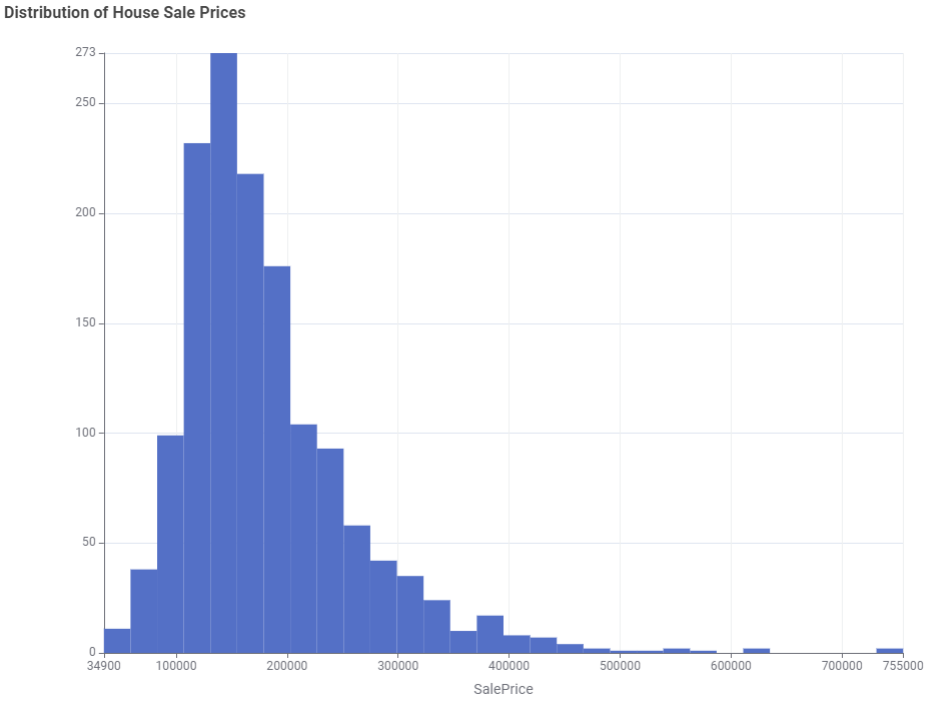


Fig.1 Distribution of House sale Prices

Fig.2 Average House Sale Price by Neighbourhood

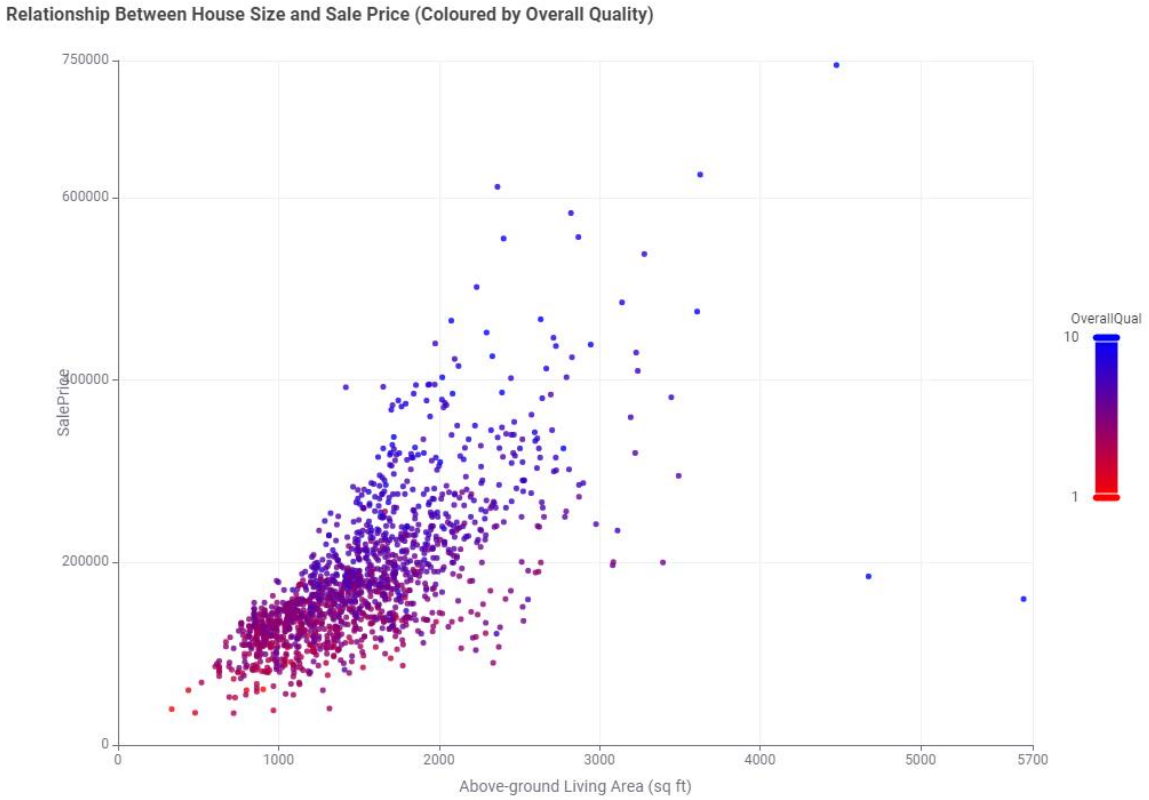


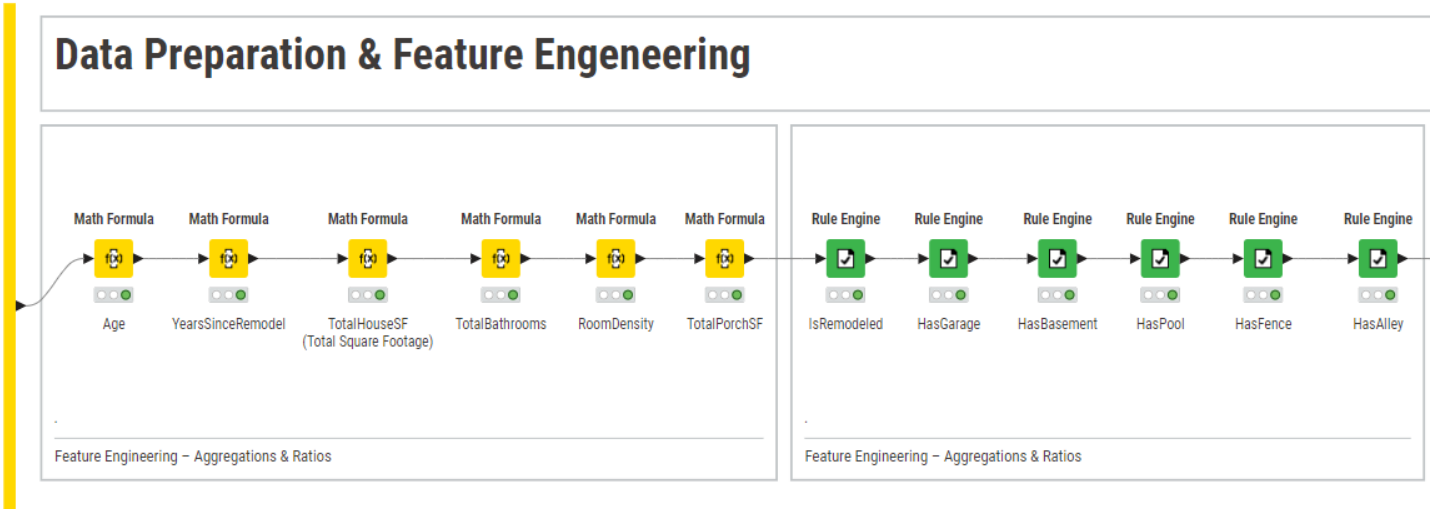
Fig.3 Relationship Between House Size and Sale Price (Coloured by Overall Quality)

MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1
No. missings: 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0	**No. missings:** 0
Top 100: 20 : 536 60 : 299 50 : 144 120 : 87 30 : 69 160 : 63 70 : 60 80 : 58 90 : 52 190 : 30 85 : 20 75 : 16 45 : 12 180 : 10 40 : 4	**Top 100:** RL : 1151 RM : 218 FV : 65 RH : 16 C (all) : 10	**Top 100:** NA : 259 60 : 143 70 : 70 80 : 69 50 : 57 75 : 53 65 : 44 85 : 40 78 : 25 21 : 23 90 : 23 68 : 19 24 : 19 64 : 19 73 : 18 72 : 17 63 : 17 55 : 17 79 : 17 100 : 16 51 : 15 66 : 15 74 : 15 52 : 14 59 : 13	**Top 100:** 7200 : 25 9600 : 24 6000 : 17 10800 : 14 9000 : 14 8400 : 14 1680 : 10 7500 : 9 6120 : 8 6240 : 8 9100 : 8 8125 : 8 3182 : 7 8450 : 6 7800 : 6 4500 : 5 10400 : 5 5400 : 5 5000 : 5 10140 : 5 4435 : 5 10000 : 5 9750 : 5 11250 : 4 8500 : 4	**Top 100:** Pave : 1454 Grvl : 6	**Top 100:** NA : 1369 Grvl : 50 Pave : 41	**Top 100:** Reg : 925 IR1 : 484 IR2 : 41 IR3 : 10	**Top 100:** Lvl : 1311 Bnk : 63 HLS : 50 Low : 36	**Top 100:** AllPub : 1459 NoSeWa : 1	**Top 100:** Inside : 1052 Corner : 263 CulDSac : 94 FR2 : 47 FR3 : 4	**Top 100:** Gtl : 1382 Mod : 65 Sev : 13	**Top 100:** NAmes : 225 CollgCr : 150 OldTown : 113 Edwards : 100 Somerst : 86 Gilbert : 79 NridgHt : 77 Sawyer : 74 NWAmes : 73 SawyerW : 59 BrkSide : 58 Crawfor : 51 Mitchel : 49 NoRidge : 41 Timber : 38 IDOTRR : 37 ClearCr : 28 StoneBr : 25 SWISU : 25 MeadowV : 17 Blmngtn : 17 BrDale : 16 Veenker : 11 NPkVill : 9 Blueste : 2	**Top 100:** Norm : 1260 Feedr : 81 Artery : 48 RRAn : 26 PosN : 19 RRAe : 11 PosA : 8 RRNn : 5 RRNe : 2

Fig.4 Statistics view of data before Feature Engineering

MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2
No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0	No. missings: 0
Top 100: 20 : 536 60 : 299 50 : 144 120 : 87 30 : 69 160 : 63 70 : 60 80 : 58 90 : 52 190 : 30 85 : 20 75 : 16 45 : 12 180 : 10 40 : 4	**Top 100:** RL : 1151 RM : 218 FV : 65 RH : 16 C (all) : 10	**Top 100:** 0.0 : 259 60.0 : 143 70.0 : 70 80.0 : 69 50.0 : 57 75.0 : 53 65.0 : 44 85.0 : 40 78.0 : 25 21.0 : 23 90.0 : 23 68.0 : 19 24.0 : 19 64.0 : 19 73.0 : 18 72.0 : 17 63.0 : 17 55.0 : 17 79.0 : 17 100.0 : 16 51.0 : 15 66.0 : 15 74.0 : 15 52.0 : 14 59.0 : 13	**Top 100:** 7200 : 25 9600 : 24 6000 : 17 10800 : 14 9000 : 14 8400 : 14 1680 : 10 7500 : 9 6120 : 8 6240 : 8 9100 : 8 8125 : 8 3182 : 7 8450 : 6 7800 : 6 4500 : 5 10400 : 5 5400 : 5 5000 : 5 10140 : 5 4435 : 5 10000 : 5 9750 : 5 11250 : 4 8500 : 4	**Top 100:** Pave : 1454 Grvl : 6	**Top 100:** Reg : 925 IR1 : 484 IR2 : 41 IR3 : 10	**Top 100:** Lvl : 1311 Bnk : 63 HLS : 50 Low : 36	**Top 100:** AllPub : 1459 NoSeWa : 1	**Top 100:** Inside : 1052 Corner : 263 CulDSac : 94 FR2 : 47 FR3 : 4	**Top 100:** Gtl : 1382 Mod : 65 Sev : 13	**Top 100:** NAmes : 225 CollgCr : 150 OldTown : 113 Edwards : 100 Somerst : 86 Gilbert : 79 NridgHt : 77 Sawyer : 74 NWAmes : 73 SawyerW : 59 BrkSide : 58 Crawfor : 51 Mitchel : 49 NoRidge : 41 Timber : 38 IDOTRR : 37 ClearCr : 28 StoneBr : 25 SWISU : 25 MeadowV : 17 Blmngtn : 17 BrDale : 16 Veenker : 11 NPkVill : 9 Blueste : 2	**Top 100:** Norm : 1260 Feedr : 81 Artery : 48 RRAn : 26 PosN : 19 RRAe : 11 PosA : 8 RRNn : 5 RRNe : 2	**Top 100:** Norm : 1445 Feedr : 6 Artery : 2 RRNn : 2 PosN : 2 PosA : 1 RRAn : 1 RRAe : 1

Fig.5 Statistics views after Feature Engineering Treatment



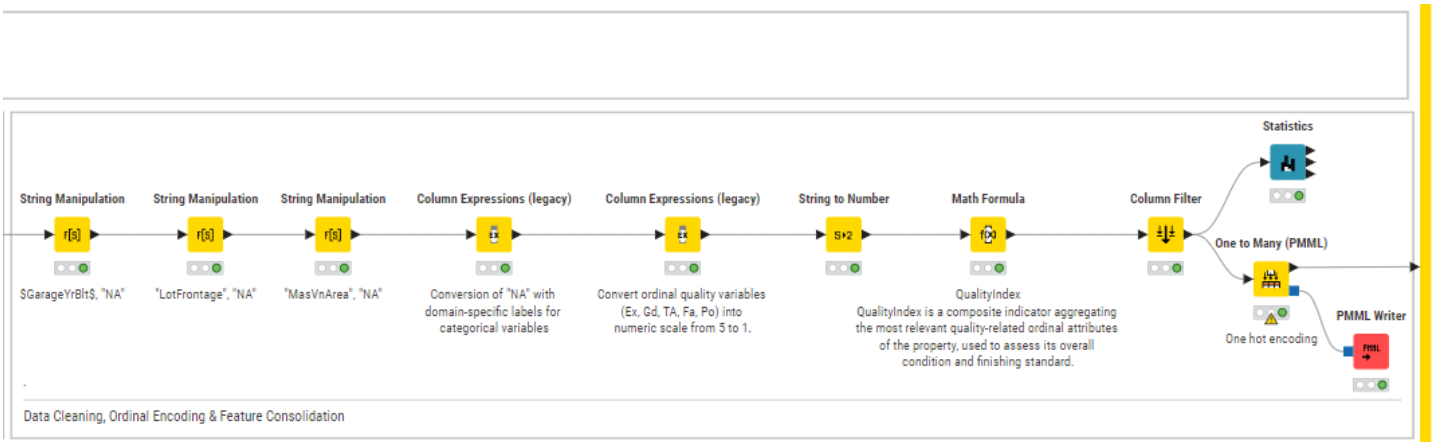


Fig.6 Feature Engineering

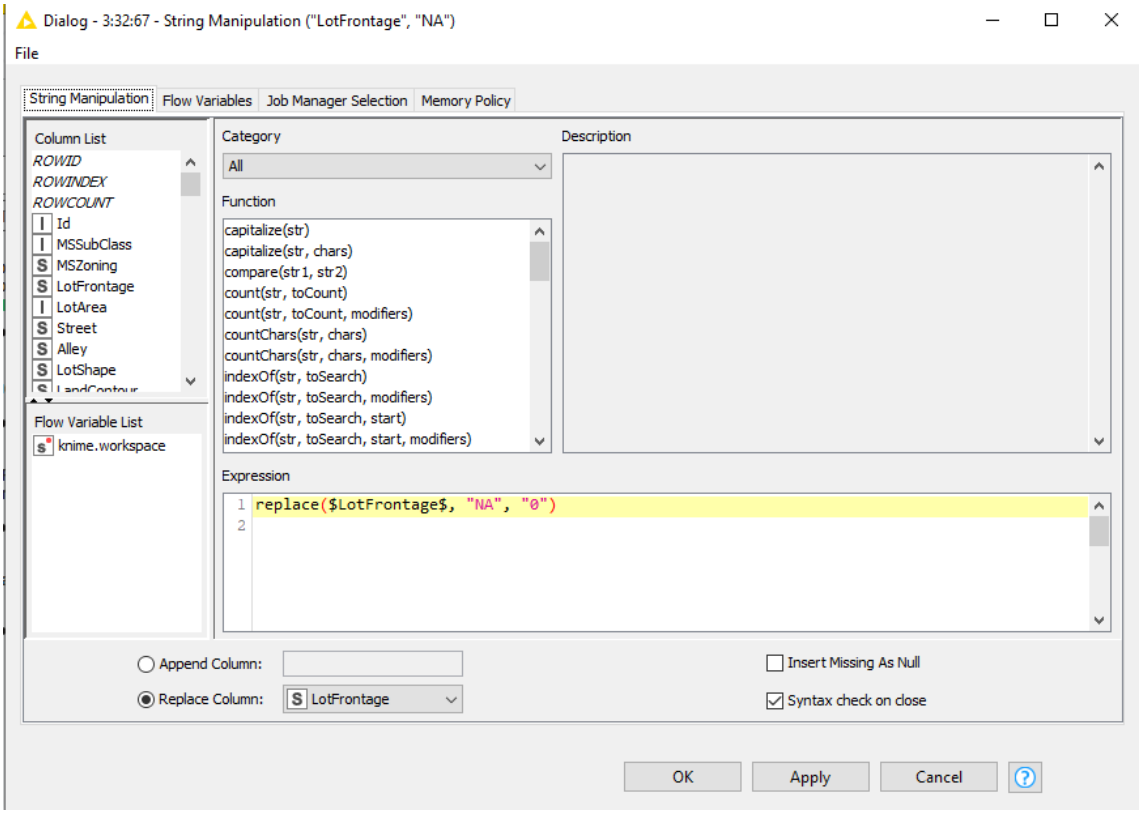


Fig.7 String Manipulation- "NA" replacement for the variable LotFrontage

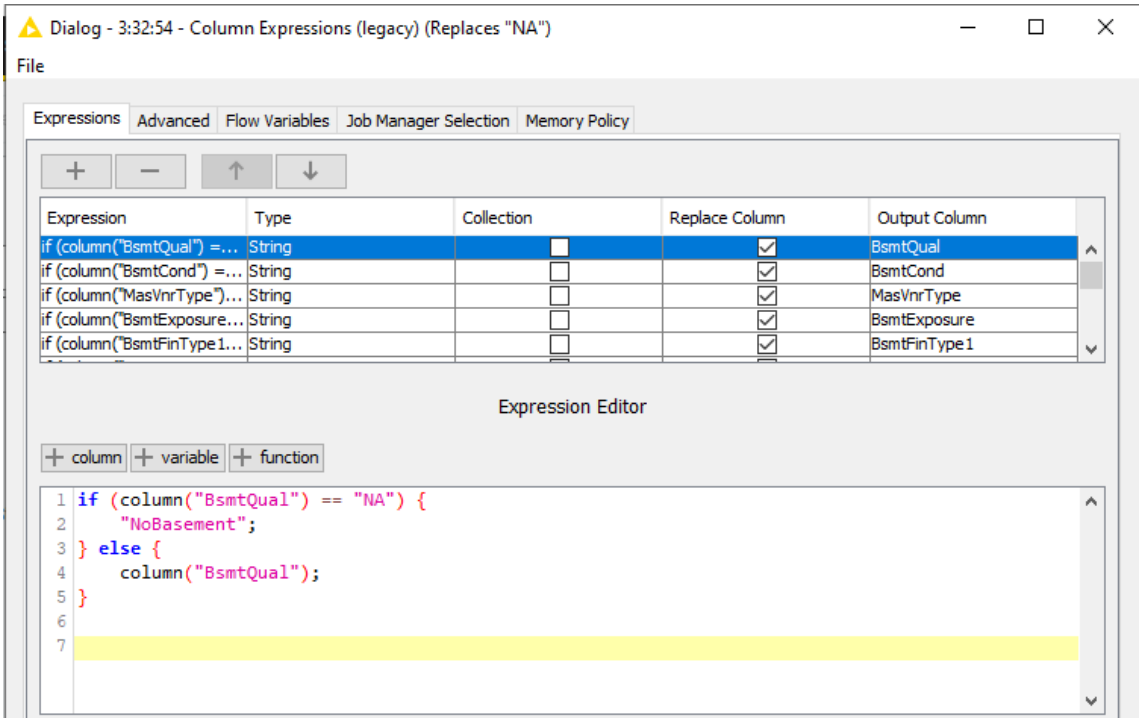


Fig.8 Replace NA with domain-specific labels for categorical variables where NA does not mean missing, but absence of the feature

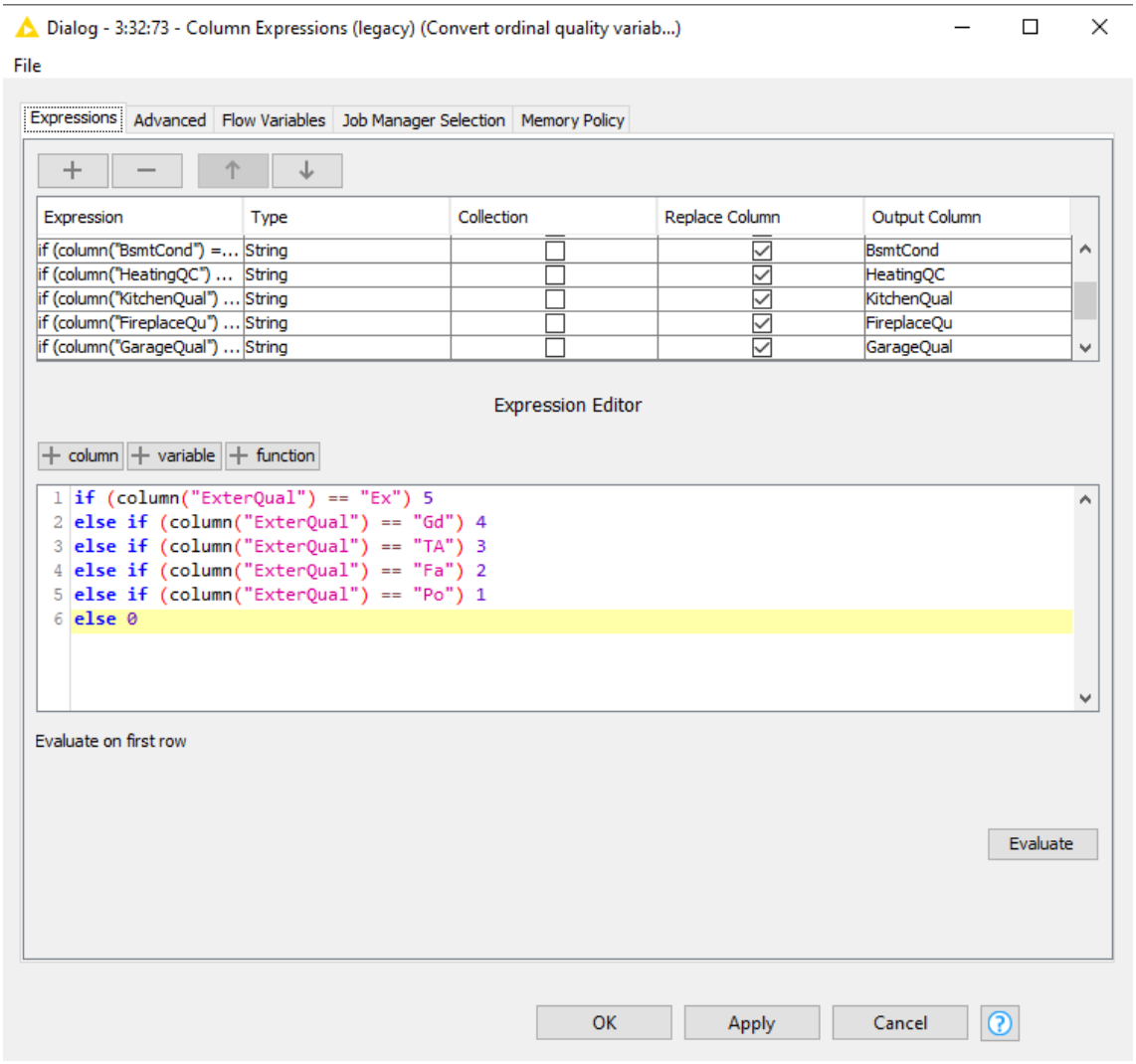


Fig.9 Column Expressions-Convert ordinal quality variables (Ex, Gd, TA, Fa, Po) into numeric scale from 5 to 1.

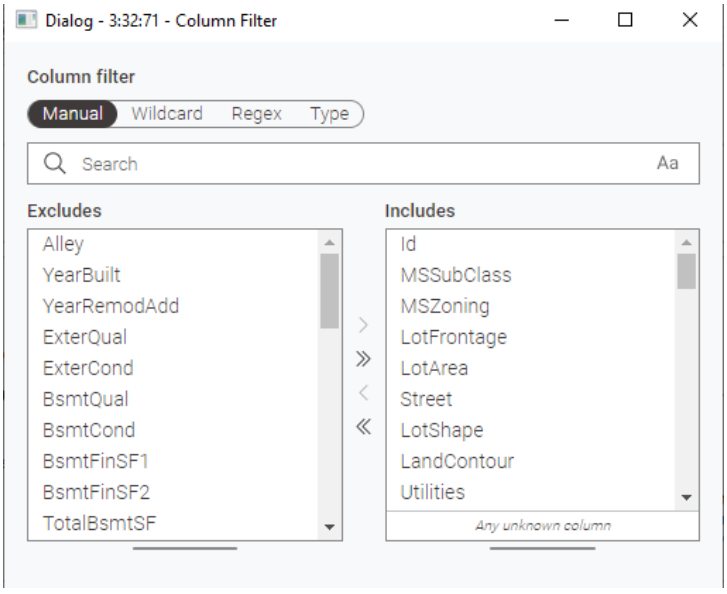


Fig.10 Column filter before partitioning

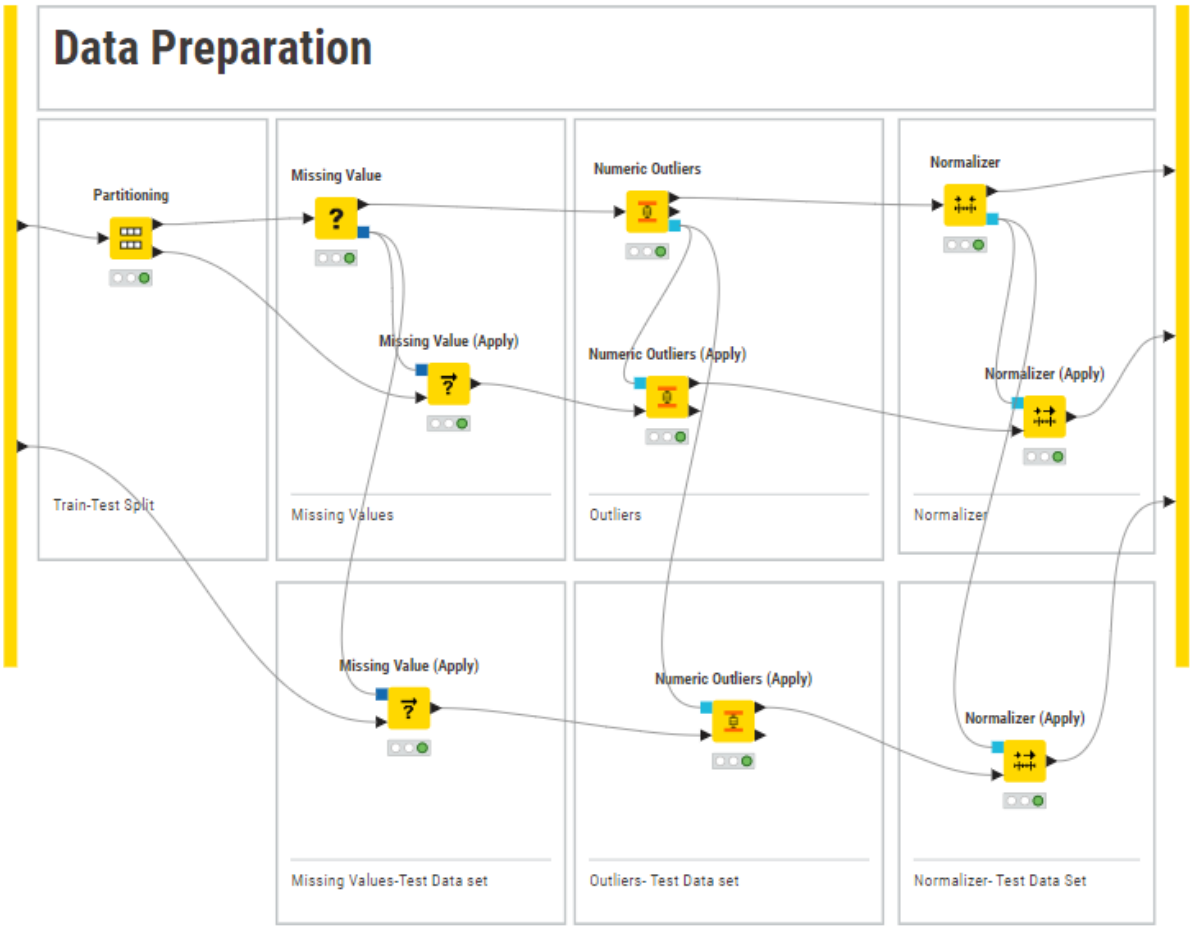


Fig 11. Data Preparation section

Table View

Rows: 7 | Columns: 3

RowID	Prediction Linear Regression *Number (double)*	Prediction Random Forest *Number (double)*	Prediction XGBoost *Number (double)*
R^2	0.784	0.773	0.874
mean absolute error	25,499.317	22,972.092	18,568.852
mean squared error	1,376,032,205.534	1,446,891,360.86	800,572,735.479
root mean squared error	37,094.908	38,038.025	28,294.394
mean signed difference	-2,679.115	-1,951.573	-367.148
mean absolute percentage error	0.148	0.133	0.107
adjusted R^2	0.784	0.773	0.874

Fig 12. Performance Comparison of Regression Models on House Price Prediction

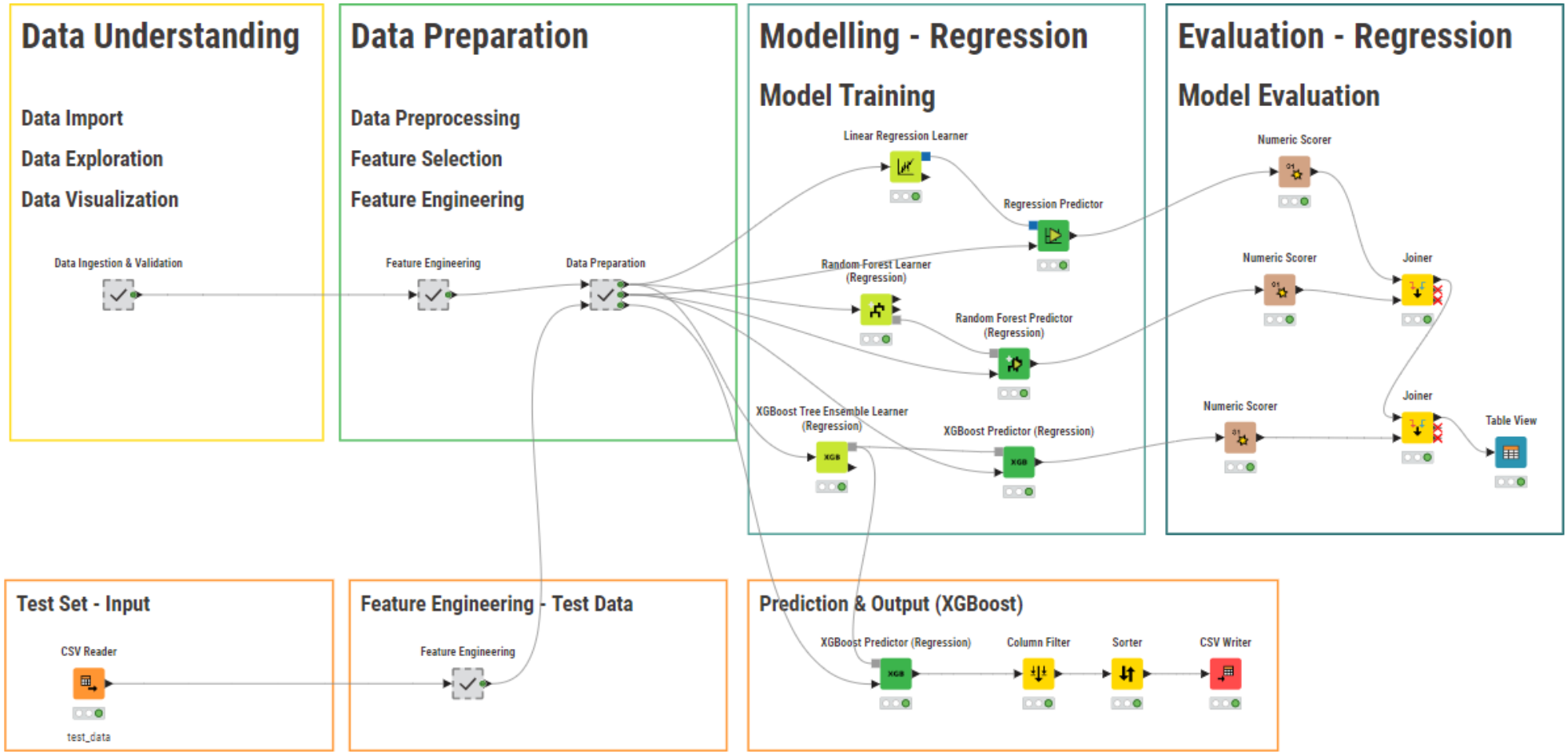


Fig 13. Overview of the KNIME Workflow for House Price Prediction