

Housing Price Prediction Using Rapid Miner Techniques

Alen p shyju, Dr Mithilles Kumar Dubey

Department Of Computer Applications, Lovely Professional University, Punjab

Abstract

A place to stay or house is a basic need of every individual of the world whether it comes to human or animal. However, it has been seen that lots of individual are being homeless. House price prediction plays a crucial role in real estate decision-making, aiding buyers, sellers, and investors in making informed choices. This study employs RapidMiner, a powerful data science platform, to develop an efficient and accurate house price prediction model. The dataset utilized comprises various features such as location, size, amenities, and historical pricing data. The results demonstrate the effectiveness of the RapidMiner-based model in accurately predicting house prices. The developed model provides valuable insights for stakeholders in the real estate market, aiding in decision-making processes. This research contributes to the growing field of data-driven real estate analytics and showcases the utility of RapidMiner as a robust tool for developing predictive models in the domain of house price prediction.

1.Introduction

The real estate market is a cornerstone of economic activity, with the buying and selling of properties being a complex interplay of numerous factors. For both homebuyers and sellers, accurately predicting house prices is of paramount importance. This task is inherently challenging due to the multifaceted nature of the real estate landscape, encompassing diverse variables such as property features, neighborhood characteristics, and economic indicators. In recent years, data mining has emerged as a powerful tool for extracting meaningful patterns and insights from large datasets. RapidMiner, a leading data science platform, provides a comprehensive environment for the entire data science lifecycle, from data preparation to model deployment. This study focuses on leveraging the capabilities of RapidMiner to develop a predictive model for house prices, aiming to enhance the accuracy and reliability of predictions in the real estate domain. The dataset utilized in this research is comprehensive, capturing a wide array of features that influence property values. Through the application of advanced data preprocessing techniques and feature engineering, we aim to distill the most relevant information for accurate prediction. The predictive model is constructed using a combination of regression algorithms within the RapidMiner framework, ensuring a robust and versatile approach to house price forecasting.

As we delve into this exploration of house price prediction using RapidMiner, the objectives are to showcase the platform's efficacy in handling the intricacies of real estate data, to highlight the significance of feature selection in improving model accuracy, and to contribute valuable insights for stakeholders in the real estate market. By the end of this study, we anticipate not only providing a reliable predictive model but also shedding light on the potential of RapidMiner as a transformative tool in the realm of real estate analytics.

2.Methodology

2.1 Data Preprocessing

Data preparation is the third most important step in data mining project. In this step we usually prepare the data for the model. This step involves cleaning all the unusable columns and data which can affect the prediction attribute. All the data preparation steps are described below.

- **Clean the missing values:**

In this step we clean or fill up the missing values in our dataset to make it more reliable for the prediction.

- **Changing the data types:**

If our dataset contains dissimilar data types, then that could be an unavoidable problem in the further process. In this step we change all the categorical values to numerical to make the model simpler to predict

- **Remove all duplicate values:**

if our prediction model contains any duplicate value then it could lead to prediction bias. so, we must remove all the duplicate values from our data set.

- **Data normalization:**

Data normalization is an important step in mining projects to make prediction reliable, but we can see that we do not have any different numerical values, so we don't need data normalization in our model.

	Name	Type
	date	 date_time
	price	# real
	bedrooms	# real
	bathrooms	# real
	sqft_living	# integer
	sqft_lot	# integer
	floors	# real
	waterfront	# integer
	view	# integer
	condition	# integer
	sqft_above	# integer
	sqft_basement	# integer
	yr_built	# integer
	yr_renovated	# integer
	street	 nominal
	city	 nominal
	statezip	 nominal
	country	 nominal

2.2 Data Analysis

Data understanding is an important factor in every data mining project. understand the data better can help to choose right model for the machine learning project. Understand the data can give us a clear picture about price distribution and fluctuation. Which can be helpful for the non-technical business leaders to understand the aspects of the project.

The data contains all information that could be found in 1990 California census. The data has been obtained from Kaggle. It has IO attributes or columns related to housing such floor,view, location and the bedroom preference etc. These attributes are used to predict the price range distribution.

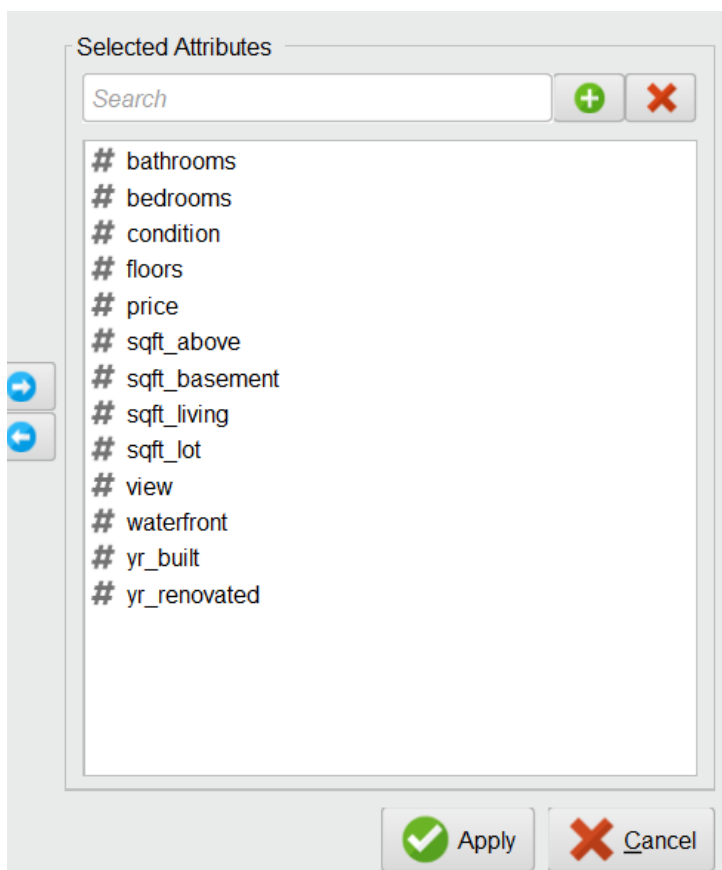
2.3 Model Bulding

- **Select the attributes :**

The Operator provides different filter types to make Attribute selection easy. Possibilities are for example: Direct selection of Attributes. Selection by a regular expression or selecting only Attributes without missing values. See parameter *attribute filter type* for a detailed description of the different filter types.

The *invert selection* parameter reverses the selection. Special Attributes (Attributes with Roles, like id, label, weight) are by default ignored in the selection. They will always remain in the resulting output ExampleSet. The parameter *include special attributes* changes this.

Only the selected Attributes are delivered to the output port. The rest is removed from the ExampleSet.



- **Nominal to Numerical**

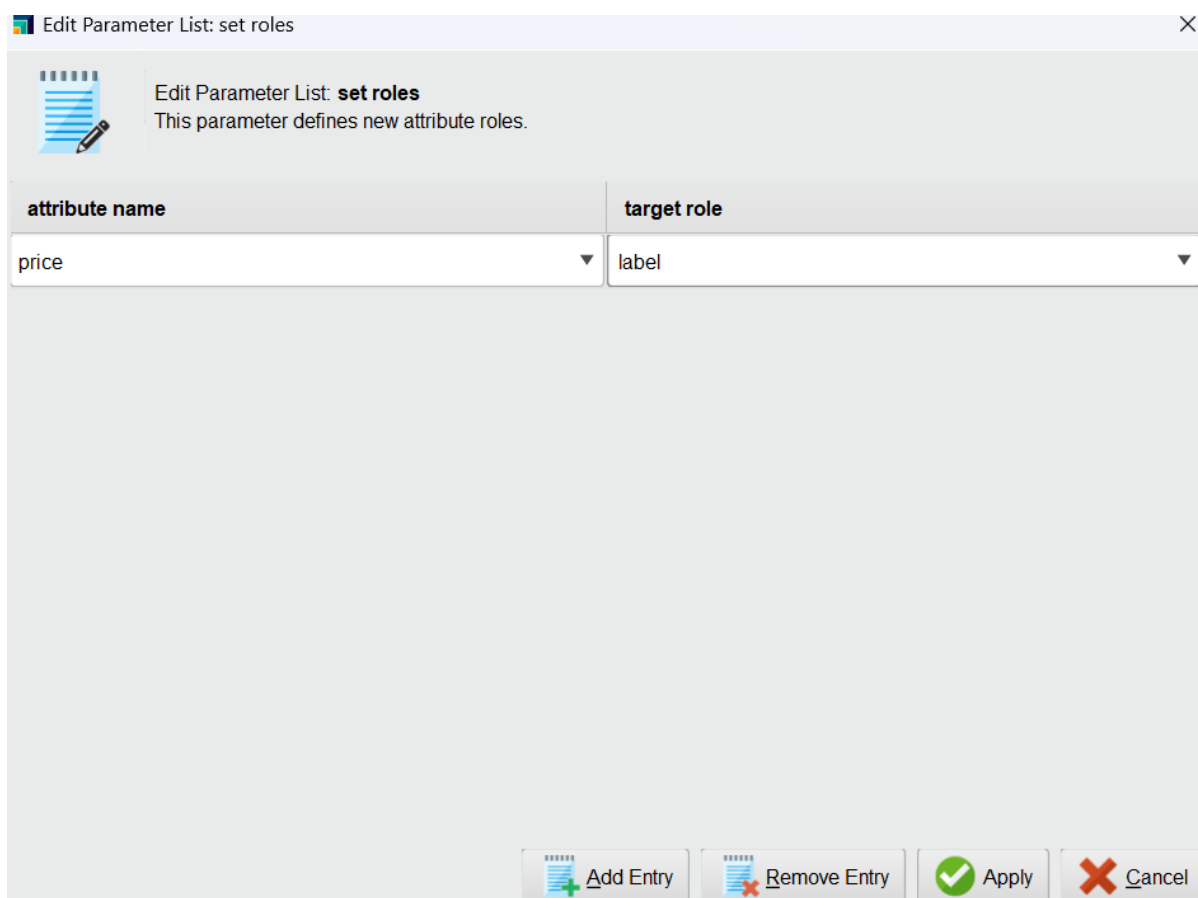
This operator changes the type of selected non-numeric attributes to a numeric type. It also maps all values of these attributes to numeric values.

▪ Set Role

The Operator provides different filter types to make Attribute selection easy. Possibilities are for example: Direct selection of Attributes. Selection by a regular expression or selecting only Attributes without missing values. See parameter *attribute filter type* for a detailed description of the different filter types.

The *invert selection* parameter reverses the selection. Special Attributes (Attributes with Roles, like id, label, weight) are by default ignored in the selection. They will always remain in the resulting output ExampleSet. The parameter *include special attributes* changes this.





Only the selected Attributes are delivered to the output port. The rest is removed from the ExampleSet.



Edit Parameter List: set roles

Edit Parameter List: **set roles**
This parameter defines new attribute roles.

attribute name	target role
price	label

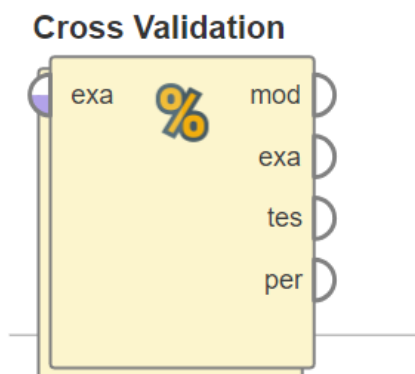
 Add Entry  Remove Entry  Apply  Cancel

▪ Cross validation

The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model. The trained model is then applied in the Testing subprocess. The performance of the model is measured during the Testing phase.

The input ExampleSet is partitioned into k subsets of equal size. Of the k subsets, a single subset is retained as the test data set (i.e. input of the Testing subprocess). The remaining $k - 1$ subsets are used as training data set (i.e. input of the Training subprocess). The cross validation process is then repeated k times, with each of the k subsets used exactly once as the test data. The k results from the k iterations are averaged (or otherwise combined) to produce a single estimation. The value k can be adjusted using the *number of folds* parameter.

The evaluation of the performance of a model on independent test sets yields a good estimation of the performance on unseen data sets. It also shows if 'overfitting' occurs. This means that the model represents the testing data very well, but it does not generalize well for new data. Thus, the performance can be much worse on test data.



▪ Linear regression

Regression is a technique used for numerical prediction. Regression is a statistical measure that attempts to determine the strength of the relationship between one dependent variable (i.e. the label attribute) and a series of other changing variables known as independent variables (regular attributes). Just like Classification is used for predicting categorical labels, Regression is used for predicting a continuous value. For example, we may wish to predict the salary of university

graduates with 5 years of work experience, or the potential sales of a new product given its price. Regression is often used to determine how much specific factors such as the price of a commodity, interest rates, particular industries or sectors influence the price movement of an asset.

Linear regression attempts to model the relationship between a scalar variable and one or more explanatory variables by fitting a linear equation to observed data. For example, one might want to relate the weights of individuals to their heights using a linear regression model.

This operator calculates a linear regression model. It uses the Akaike criterion for model selection. The Akaike information criterion is a measure of the relative goodness of a fit of a statistical model. It is grounded in the concept of information entropy, in effect offering a relative measure of the information lost when a given model is used to describe reality. It can be said to describe the tradeoff between bias and variance in model construction, or loosely speaking between accuracy and complexity of the model.

For this prediction we are using feature selection as none.

Parameters ✕

Linear Regression

feature selection none ▼ ⓘ

☒ *eliminate colinear features* ⓘ

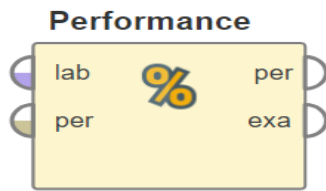
min tolerance ⓘ

☒ *use bias* ⓘ

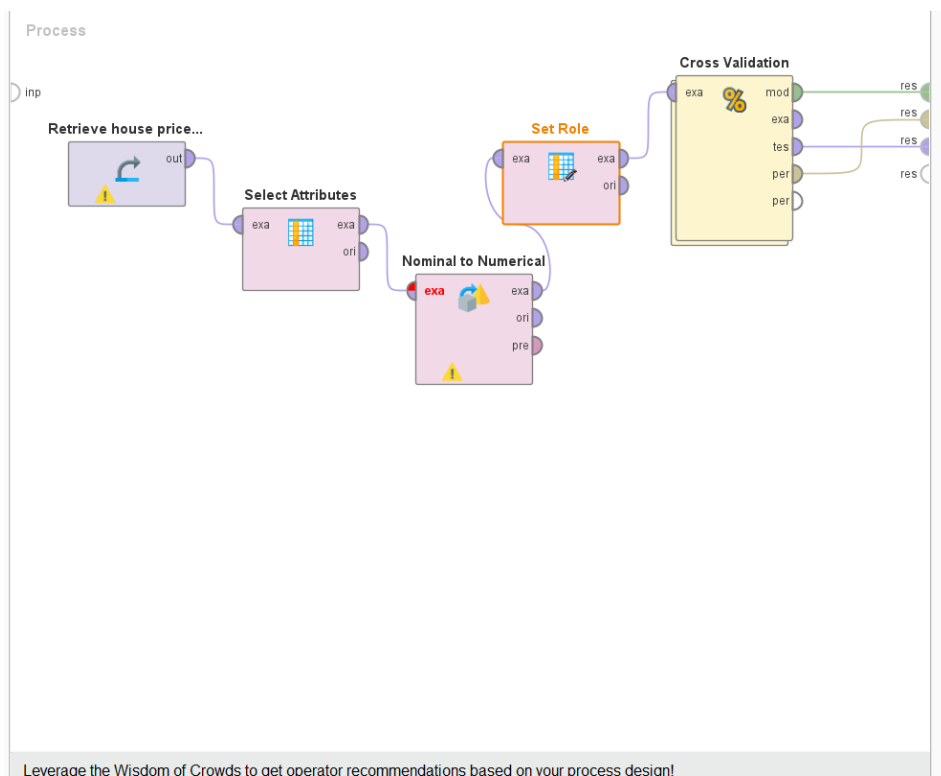
ridge 1.0E-8 ⓘ

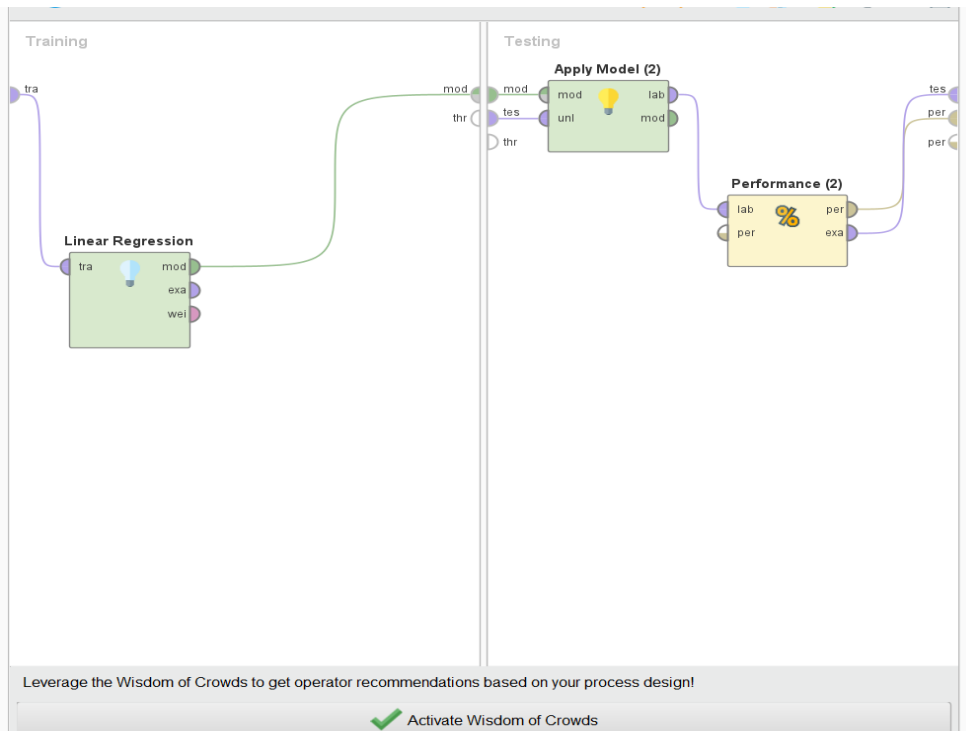
▪ Performance

This operator is used for statistical performance evaluation of regression tasks and delivers a list of performance criteria values of the regression task.



2.4 Design





Retrieve data set house price prediction into process and add operator **select attributes** and select subset select inverse of city, date, statezip, street and connect to **nominal to numerical**. Then retrieve **set role** operator it to dataset. In set role, select attribute name as medium_income and target role set as label. Then connect to **cross validation**, double click on the cross validation add function Linear regression in training side and in parameters select **none** in feature selection and connect it to output and in right of the process, testing side add operator **Apply mode** and connect it to input, retrieve **performance(regression)** operator and connect it with apply model and cross connect performance operator to output. Run the process.

2.5 Statistics Of Data & Visulization

PerformanceVector (Performance (2))

LinearRegression (Linear Regression)

Result History

ExampleSet (Cross Validation)

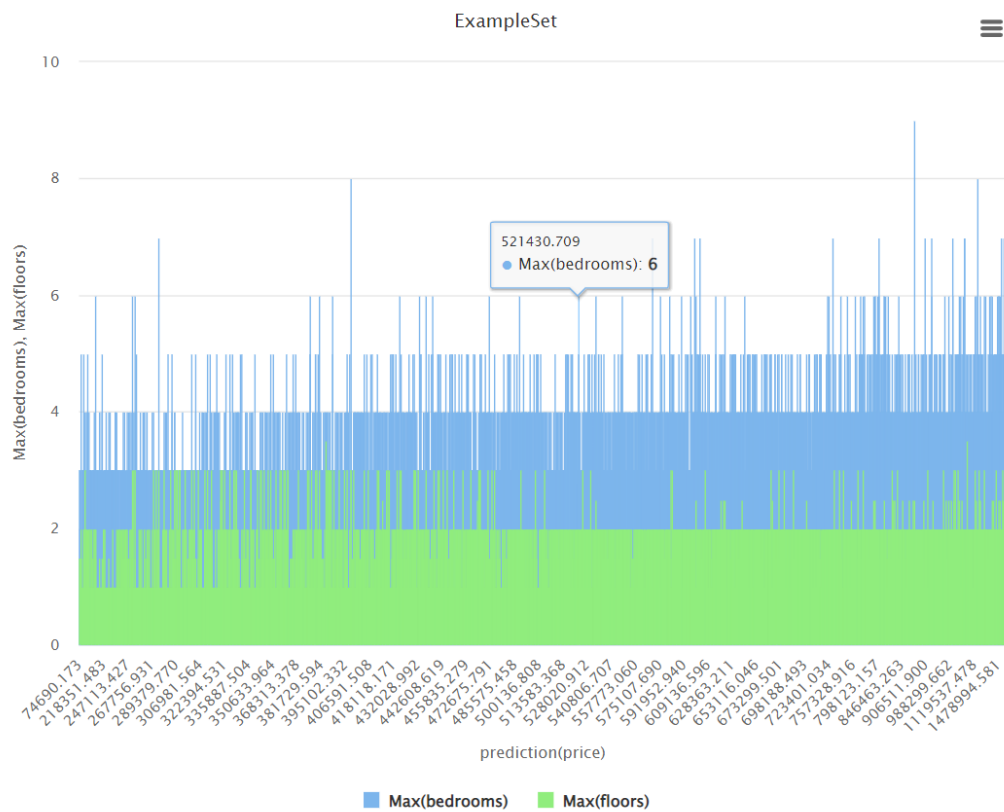
Data

Statistics

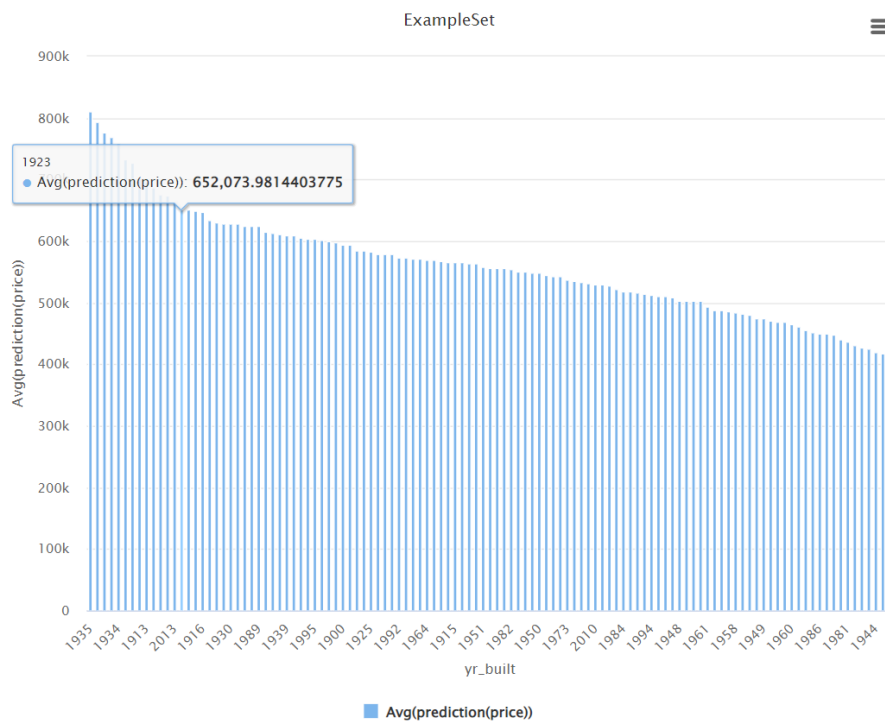
Visualizations

Annotations

Name	Type	Missing	Statistics	Filter (15 / 15 attributes):
<div>Label</div> <div>price</div>	Real	0	<div>Min</div> <div>0</div>	<div>Max</div> <div>26590000</div>
<div>Prediction</div> <div>prediction(price)</div>	Real	0	<div>Min</div> <div>74690.173</div>	<div>Max</div> <div>3532119.848</div>
<div>country = USA</div>	Integer	0	<div>Min</div> <div>1</div>	<div>Max</div> <div>1</div>
<div>bedrooms</div>	Real	0	<div>Min</div> <div>0</div>	<div>Max</div> <div>9</div>
<div>bathrooms</div>	Real	0	<div>Min</div> <div>0</div>	<div>Max</div> <div>8</div>
<div>sqft_living</div>	Integer	0	<div>Min</div> <div>370</div>	<div>Max</div> <div>13540</div>



The above figure shows that if you have \$5,21,430 you will get max 6 bedrooms and 3 floor's house.

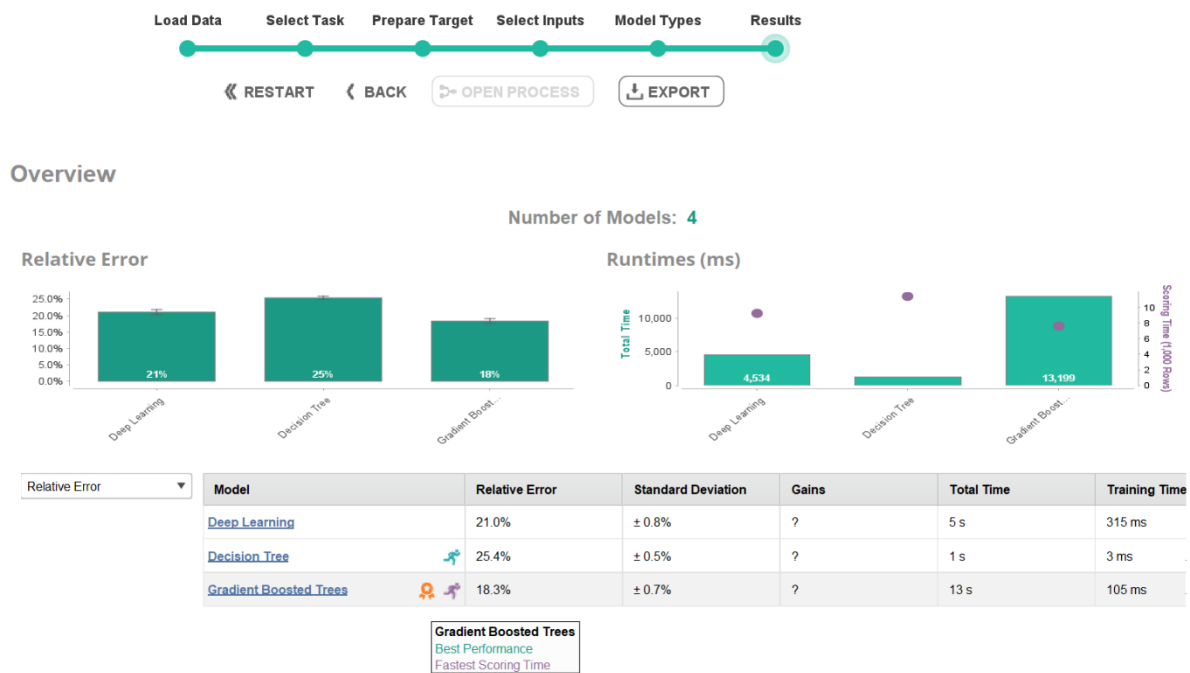


In the above figure shows that the average price of a house that built in 1923 is \$652073

3. Auto Model Selection

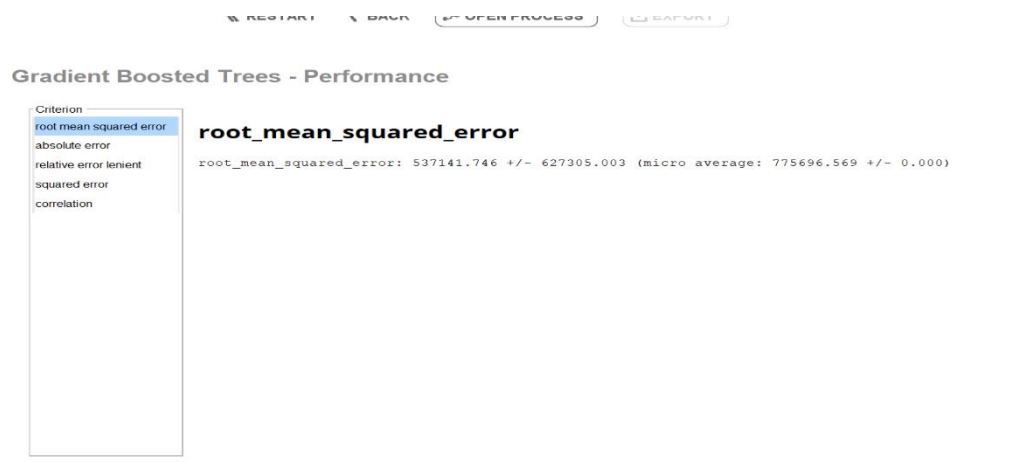
3.1 Random Forest

Random Forest is a kind of ensemble models that combines the prediction of multiple decision trees to create a more accurate final prediction. Random Forest is a verified powerful tool based on previous studies. From this auto model we can understand that **Gradient Boosted Tree** have fastest scoring time and best performance.



In the modelling phase we have implemented linear regression. As I previously mentioned we have a continuous data with median house value. So, in linear regression we have a target variable which predict the value based on other independent variables which we set as x variable. In this dataset we have our target variable y as median house value. on the other hand, we have our independent variable as x. We have propose linear regression in RapidMiner . However, in case of Auto model, we found that gradient boosted tree has the highest accuracy with minimum run time. So finally, we choose gradient boosted tree in rapid miner auto mode. All the process can be shown below.

When we talk about the auto model. Auto model selected various models but out of these models we show that Gradient boosted tree gives us a reliable accuracy and first-time consuming output. So finally, we select the gradient boosted tree.



3.2 Results

In the evaluation phase we have evaluate of rapid miner model. We got a root mean square error in regression model. However, after comparing both the root mean square error in RapidMiner we got to know that we have a higher root mean square error in rapid miner that is 537141.746.

4. Discussion and Conclusion

This paper investigates different models for housing price prediction. In the process of developing a house price prediction model, several key aspects were considered, including the selection of relevant features, data preprocessing, and the application of machine learning algorithms. The accuracy of the house price prediction model largely depends on the quality and quantity of data available for training. Feature selection and engineering play a vital role in enhancing the model's performance, as they help identify the most influential variables affecting house prices. Additionally, the choice of machine learning algorithm and proper model evaluation techniques significantly impact the model's predictive capabilities.

The developed model, based on the current dataset and methodologies, demonstrates a certain level of accuracy in predicting house prices. However, it's essential to acknowledge that the real estate market is dynamic, and external factors such as economic trends, policy changes, and unforeseen events can influence property values. Therefore, continuous monitoring, updating, and refining of the model are necessary to ensure its relevance and effectiveness.

In summary, the future scope for house price prediction involves a continuous evolution of methodologies, incorporation of new data sources, and the integration of advanced techniques to enhance model accuracy and relevance in the dynamic real estate market. Further research about the following topics should be conducted to further investigate these models, especially the combinations of different models:

- The coupling effect of multiple regression models.
- The “re-learn” ability of machine learning models.
- The combination of Machine Learning and Deep Learning methods

- The driven factors for the good performance of tree-based models.
- The faster ways to fit complex models

Reference

1. House Price Index. Kaggle <https://www.kaggle.com/code/emrearslan123/house-price-prediction>
2. Youtube <https://www.youtube.com/watch?v=jlq3rTbOaZQ&t=842s>
3. Rapid Miner <https://docs.rapidminer.com/latest/studio/operators/>
4. Tutorialspoint <https://www.tutorialspoint.com/house-price-prediction-using-machine-learning-in-python>