Housing Price

Project Report

Participations :-

Name	ID
Abdullah Hussien Ibraheem	20221427861
Abdelrahman Ashraf Ragab	20221374041
Ali Mohamed Sayed	20221449583
Fares Mohamed Fathy	20221461330
Raghdan Ramadan Mohamed	20221449509

Report Content :

1-What is our Data ?

2-What is our target ?

3-Deep dive into the project

- Pipelines Done.
- Explanation for every node.
- Final results for every pipeline.
- Pipeline comparison.
- Problems we faced during the project.

1-What is our data ?

-Our data contains *66* columns with *1,151* rows.

-The data is about housing prices , which varies depending on the house features which is explained in the columns such as : GarageQual , Heating and so many features , the better the features are , the higher the house price will be.

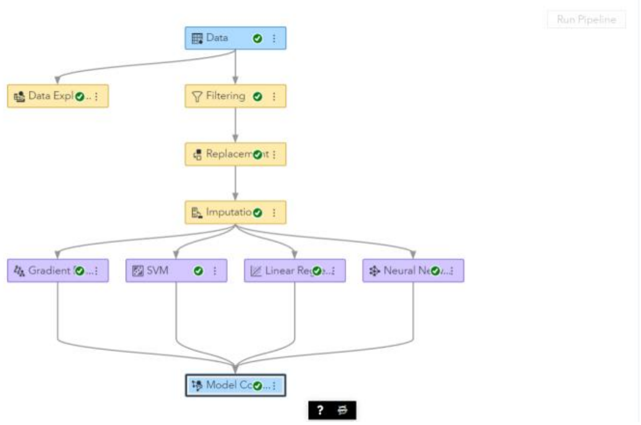
2-What is our target ?

-Our target is to build a model that predicts a house price depending on the input I give to it, To do that We will use different algorithms, such as :

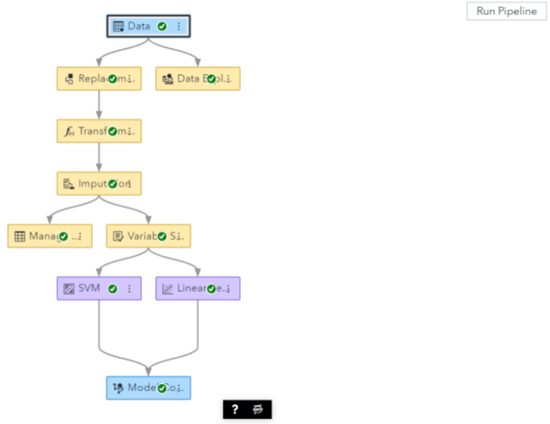
Gradient boosting , Linear regression , neural network and other algorithms.

3-A deep dive into the project

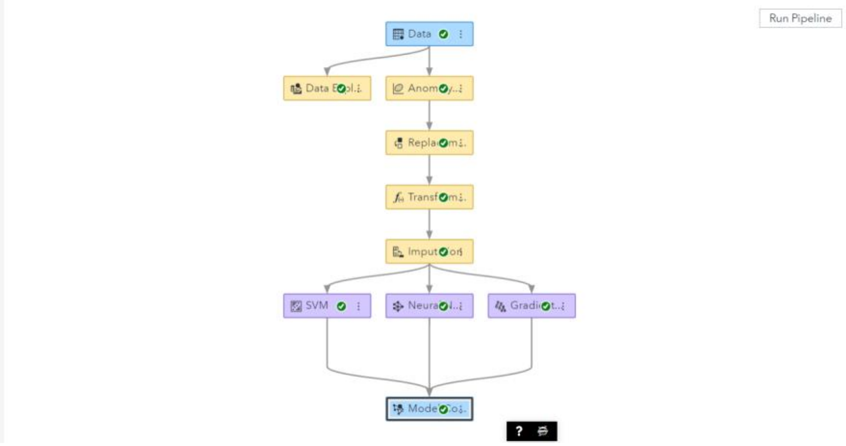
- Pipelines done :-
    1. Pipeline 1



    2. Pipeline 2



    3. Pipeline 3 (Best)



- Explanation for every node

Filtering Node:

- ✓ Class Input Filtering Method: This option allows you to specify the method for filtering categorical/class variables. It determines how to handle classes with low frequencies or rare levels.
- ✓ Percentage Cutoff: It is a threshold specified as a percentage. If the percentage of observations for a class is below this cutoff, it may be filtered out.
- ✓ Max Number of Levels Cutoff: This option limits the number of levels that can be retained for a class variable. If a class has more levels than this cutoff, some levels may be filtered out.
- ✓ Interval Filtering Limits Method: This method is used for filtering continuous/interval variables. It determines how to handle observations that fall outside specific limits.
- ✓ Cutoff for Standard Deviations: This option allows you to specify the number of standard deviations from the mean as a cutoff for filtering out-of-range values.
- ✓ Filter Indicator Usage: This option controls whether to include a binary indicator variable that indicates whether an observation was filtered or not.

Replacement Node:

- ✓ Replacement Value for Unknown Class Levels: This option allows you to specify a replacement value for observations with unknown or missing levels in categorical/class variables.
- ✓ Interval Inputs: These options determine how to handle missing or out-of-range values in continuous/interval variables.
- ✓ You can choose the Default Limits Method, Standard Deviations, or specify a Replacement Value.

Imputation Node:

- ✓ Missing Percentage Cutoff: This option sets a threshold as a percentage. If the percentage of missing values for a variable exceeds this cutoff, it may be flagged or imputed.
- ✓ Reject Original Variables (checkbox): When enabled, this option rejects the original variables with missing values and replaces them with imputed values.
- ✓ Class Inputs: This option specifies the method for imputing missing values in categorical/class variables. The default method will be used.
- ✓ Interval Inputs: These options determine how to impute missing values in continuous/interval variables. You can use the default method, specify data limits for calculating imputed values, or specify a data limit percentage.
- ✓ Distributions Method Random Seed: This option sets the random seed for generating imputed values using the distribution-based method.
- ✓ Indicators: This option controls the creation of binary indicator variables that indicate whether a value was imputed or not.

Sure! Here's an explanation of the nodes you mentioned in the SAS Viya platform:

Transformation Node:

- ✓ Default Interval Inputs Method: This option determines the default method for transforming continuous/interval variables. It specifies the transformation technique to be used.
- ✓ Ranking Criterion for Best Transformation: These criteria are used to evaluate and rank the transformations for interval, binary, and nominal target variables. Different criteria may be applicable depending on the target variable type.
- ✓ Class Inputs: This option specifies the method for transforming categorical/class variables. The default method will be used.
- ✓ Default Class Inputs Method: This option determines the default method for transforming categorical/class variables.

- ✓ Rare Cutoff Value Percentage: It is a threshold specified as a percentage. If the percentage of observations for a category is below this cutoff, it may be considered a rare category and transformed accordingly.
- ✓ WOE (Weight of Evidence) Adjustment Value: This value is used in the calculation of the Weight of Evidence transformation for categorical variables.
- ✓ Missing Values Treatment: This option determines how missing values are handled during the transformation process.
- ✓ Reject Original Variables (checkbox): When enabled, this option rejects the original variables after transformation.

Variable Selection Node

- ✓ Combination Criterion: This criterion determines how variables are combined or selected during the variable selection process.
- ✓ Unsupervised Selection (Checkbox): When enabled, this option performs unsupervised variable selection techniques.
- ✓ Maximum Steps: It sets the maximum number of steps allowed during the variable selection process.
- ✓ Maximum Variables: This option specifies the maximum number of variables to select.
- ✓ Cumulative Variance Cutoff: It is a threshold for the cumulative variance explained by the selected variables.
- ✓ Incremental Variance Cutoff: This threshold specifies the incremental variance required for a variable to be included in the selection.
- ✓ Selection Process: Specifies the type of variable selection process to be performed, such as fast supervised selection, linear regression selection, decision tree selection, etc.
- ✓ Effect-Selection Criterion: This criterion determines the selection of variables based on their effects.
- ✓ Selection-Process Stopping Criterion: It specifies the stopping criterion for the variable selection process.
- ✓ Model-Selection Criterion: This criterion is used to select the best model during the variable selection process.
- ✓ Entry Significance Level: It sets the significance level required for a variable to enter the model.
- ✓ Stay Significance Level: This level determines the significance required for a variable to remain in the model.

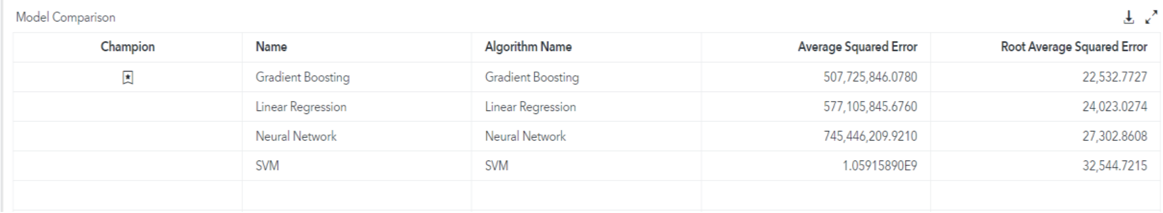
- ✓ Maximum/Minimum Number of Effects/Steps: These options limit the number of effects or steps in the variable selection process.
- ✓ Suppress Intercept: When enabled, this option excludes the intercept term from the model.

Anomaly Detection Node:

- ✓ Standardize Interval Inputs (checkbox): When enabled, this option standardizes the interval inputs before performing anomaly detection.
- ✓ Include Class Inputs (checkbox): When enabled, this option includes class variables in the anomaly detection process.
- ✓ Gaussian Kernel Bandwidth: It sets the bandwidth for the Gaussian kernel used in anomaly detection.

- Final results for every pipeline

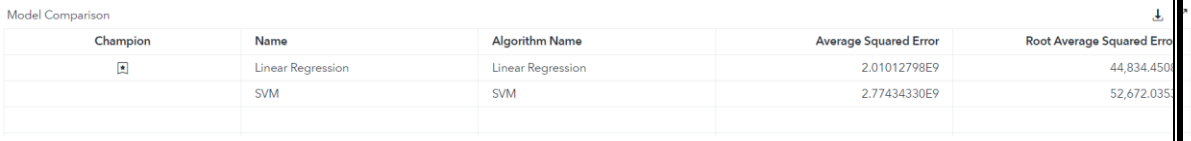
  - Pipeline 1



Model Comparison

Champion	Name	Algorithm Name	Average Squared Error	Root Average Squared Error
★	Gradient Boosting	Gradient Boosting	507,725,846.0780	22,532.7727
	Linear Regression	Linear Regression	577,105,845.6760	24,023.0274
	Neural Network	Neural Network	745,446,209.9210	27,302.8608
	SVM	SVM	1.05915890E9	32,544.7215

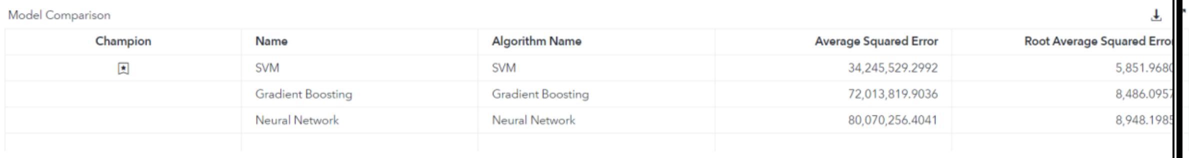
  - Pipeline 2



Model Comparison

Champion	Name	Algorithm Name	Average Squared Error	Root Average Squared Erro
★	Linear Regression	Linear Regression	2.01012798E9	44,834.450
	SVM	SVM	2.77434330E9	52,672.035

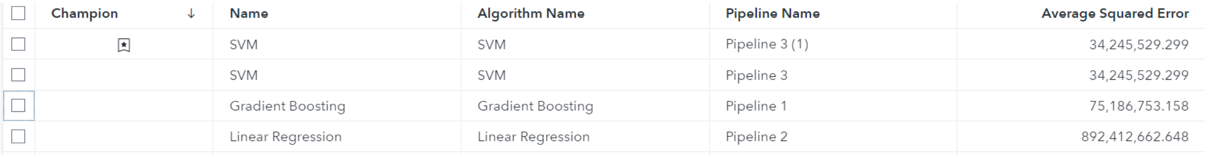
  - Pipeline 3



Model Comparison

Champion	Name	Algorithm Name	Average Squared Error	Root Average Squared Erro
★	SVM	SVM	34,245,529.2992	5,851.968
	Gradient Boosting	Gradient Boosting	72,013,819.9036	8,486.095
	Neural Network	Neural Network	80,070,256.4041	8,948.198

- Pipeline Comparison



□	Champion ↓	Name	Algorithm Name	Pipeline Name	Average Squared Error
□	★	SVM	SVM	Pipeline 3 (1)	34,245,529.299
□		SVM	SVM	Pipeline 3	34,245,529.299
□		Gradient Boosting	Gradient Boosting	Pipeline 1	75,186,753.158
□		Linear Regression	Linear Regression	Pipeline 2	892,412,662.648

- Problems we faced during the project.

  - **At first , we started with an avg squared error that was approximately 5 billion , we tried to work it out with all possible ways that we by using different settings for each node and different settings in each model and we edited in the data in all the various ways and we didn’t reach up to anything , the lowest error we reached and was 34 million.**

  1. **When we opened the pipeline comparison to see the champion model , we found different avg squared error for both pipeline 1 and 2 such that the error in the pipeline comparison is lower than the results in each pipeline individually**
  2. **We tried our best to reduce the error but we failed , We worked very hard to understand all the settings for each node and pipeline we used and we self-studied all the algorithms to increase the chance that we reduce the error but all our attempts failed and we are sorry for that.**

At the end we hope that our project suits the training effort that you Dr.Mohamed did with us and we thank you so much for such a great training.