**ACS61013 Data Modelling and Machine Intelligence**

**COURSEWORK 1**

**Table of Content**

[**Introduction** 2](#_Toc87980710)

[**Section 1** 3](#_Toc87980711)

[**Domain Analysis** 3](#_Toc87980712)

[**Findings of Domain Analysis** 5](#_Toc87980713)

[**Section 2** 5](#_Toc87980714)

[**Data Cleaning and Pre-processing** 5](#_Toc87980715)

[**How Understanding of Domain support this task** 7](#_Toc87980716)

[**Section 3** 7](#_Toc87980717)

[**Feature Engineering & Preventing Bias** 7](#_Toc87980718)

[**Correlated features in our Data set** 8](#_Toc87980719)

[**Section 4** 10](#_Toc87980720)

[**Choosing a Machine learning Algorithm** 10](#_Toc87980721)

[**Choosing Software for Model Building** 10](#_Toc87980722)

[**Section 5** 11](#_Toc87980723)

[**Discuss how you applied cross-validation technique in the machine learning pipeline.** 11](#_Toc87980724)

[**Section 6** 12](#_Toc87980725)

[**Learning Curves** 12](#_Toc87980726)

[**Section 7** 14](#_Toc87980727)

[**K-Nearest neighbour ( KNN)** 14](#_Toc87980728)

[**Adaboost Algorithm** 14](#_Toc87980729)

[**Conclusion** 15](#_Toc87980730)

# 

# **Introduction**

The objective of this report is to suggest the best machine learning model that can predict the housing price for a given dataset. This report describes various steps involved in complete cycle from model development to model deployment. After careful consideration of all the available alternatives **Linear Regression** has been chosen to build a predictive model using **Orange**. The complete report has been divided into seven sections namely **Domain Analysis** , **Data pre-processing** and **Cleaning** ,**Feature Engineering** , **Model Deployment** , **Validation** , **Error Estimation** and **Comparative Studies** with other algorithms. We have done the Error estimation to determine the effectiveness of our model for which we used **learning curves.** To generate learning curves, we have used **python**. In comparative Studies we have discussed which other algorithms could have been used instead of Linear regression. In short, this report can be used as a reference document to understand the process ML model Development.

# **Section 1**

## **Domain Analysis**

The purpose of Domain analysis is to explore domain of the provided data. To find the lineage of data , identify its origin .To ask questions like , what is the data telling us and what is its purpose?

The dataset provided to us is a housing dataset that has 26 columns and 1460 instances. The 26 columns comprise of description of physical attributes of different properties. These attributes consist of sales price , property location , its age , Area , Dimensions , Shape , Type of house , its quality , condition , foundation type , heating , air conditioning , electrical supply etc. The purpose of given data set is to predict the housing prices based on given parameters. Lets discuss some of the findings that we have made on researching about the data.

* Newer houses built after 1946 tends to sell at higher prices then all other types of properties.
* The houses built before 1945 seems to be selling at least prizes.
* Most of the properties for sale are in low density residential area . however, nobody is willing to sell their property if there’s a park nearby.
* The houses with larger LotFrontage tends to sell at higher prices and have more area in most cases.
* Houses in low density area have larger LotFrontage and Area which means they have more open spaces and are generally preferred by single family
* Most of the 2 story houses were built after 1945.
* Houses with a greater number of Fireplaces tends to sell at higher prices.
* Houses with better overall quality tends to sell at higher prices.
* Houses with better overall quality tends to have better kitchen quality as well and tends to sell at hight prices.

Apart to domain knowledge **Exploratory data analytics** techniques are also helpful in deriving meaningful inferences.

* On preparing a heat map of the available data it is visible that **PoolQC** and **Fence** are the two variables that has very small number of instances.

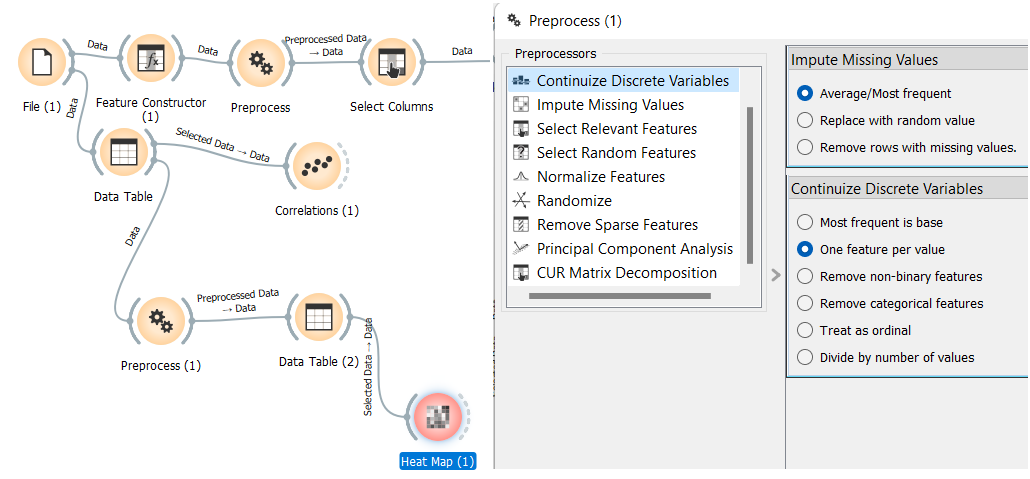
(**Note**: **Heat map** can be obtained by using Heatmap widget available in orange. It can be used to identify the variables that has large number of missing values. **One-Hot Encoding** has been used to visualize each categorical variable as a separate entity)

A picture containing timeline

Description automatically generated

**Figure 1 - Heat map of all the variables in One-Hot-Encoding to check the features with most missing values**

To obtain **One-Hot Encoding** of the dataset **pre-processor** widget is used . The use has been shown below along with orange pipeline. We select one feature per value under the **Continuize Discrete variables** tab to get **One-Hot Encoding**.



**Figure 2 - One-Hot Encoding using Pre-Processor widget**

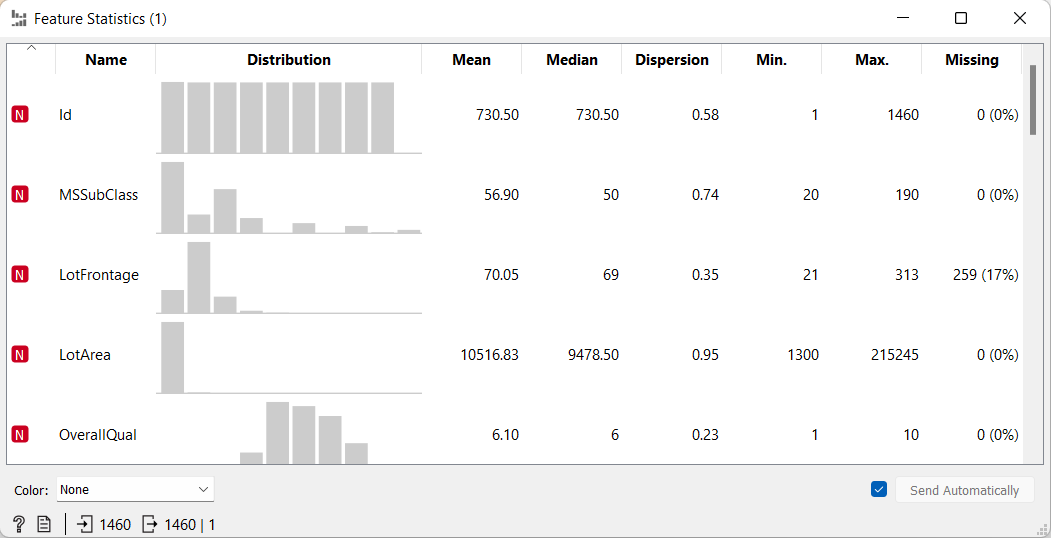
* Here it is observed that variables such as **SalePrice** , **LotArea** & **LotFrontage** are **skewed.**

Chart, histogram

Description automatically generated

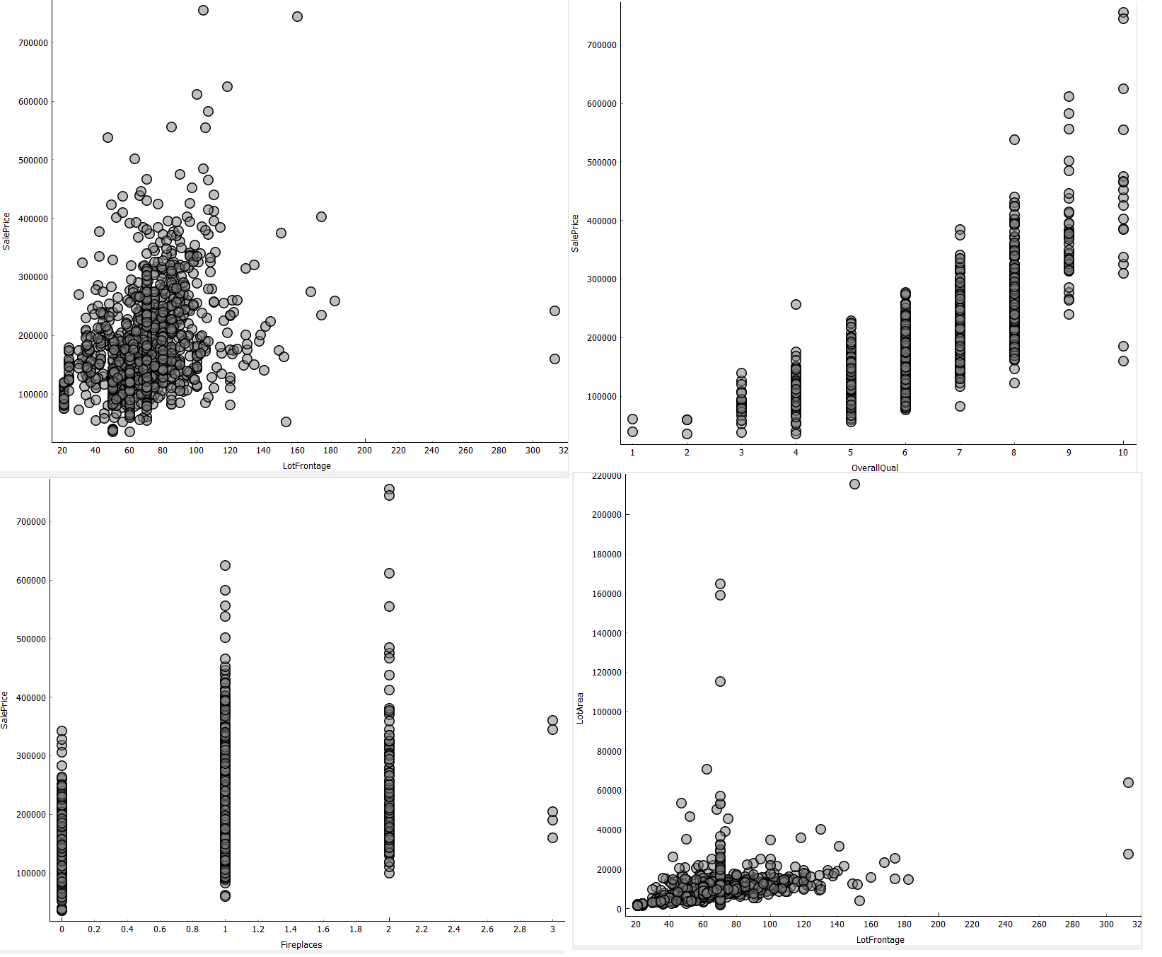
**Figure 3 - Observed Skewed Features ( LotArea, SalePrice, LotFrontage)**

* **LotFrontage** tells us about the length of property connected to street. 17 percent of data is missing under this parameter which is a significant number.
* To understand the **distribution** of different variables, feature statistics widget is very useful . it directly tells us about the distribution of different variables which is very useful in finding outliers.



**Figure 4 - Using Feature statistics widget to understand the distribution of different variables**

**Correlation** helps us in understanding how one or more variables are related to each other .



**Figure 5 - Using Scatter Plot to identify the correlation between different data variables**

## **Findings of Domain Analysis**

During the domain analysis the given dataset was explored using orange pipelines. Various interesting insights about data were found as mentioned in above section. These insights help us in cleaning our data and preparing it to generate a ML pipeline for the prediction of housing prices. For example , Variable named **ID** is irrelevant to our dataset and can be removed. **PoolQC** and **Fence** has large number of missing instances. We also need to remove skewness from our target variable **SalePrice &** independent variables **LotArea** & **LotFrontage** by applying log transform over them. We also used **Heatmap** to analyse the variables obtained after **one-Hot encoding** and we found some variables that can be discarded . Thus, we can conclude that Domain analysis has helped us by providing information regarding data cleaning.

# **Section 2**

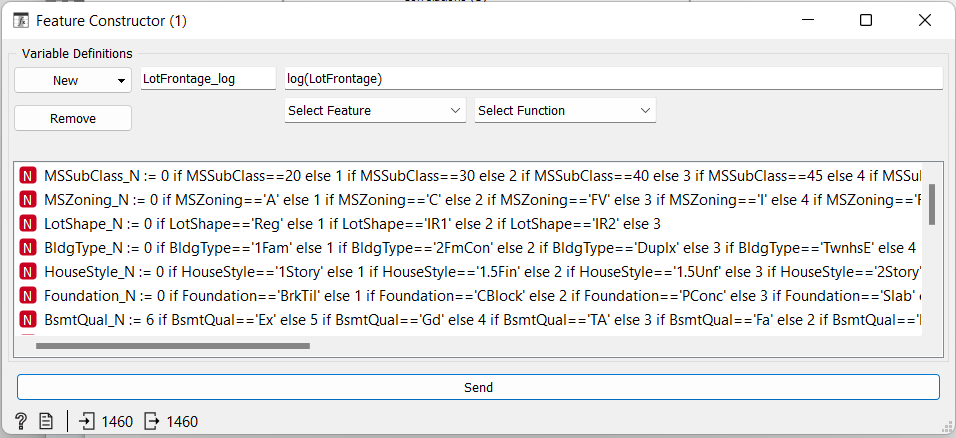
## **Data Cleaning and Pre-processing**

The dataset provides to us has 26 variables and out of those 21 are categorical so we need to convert those categorical variables into numeric values as many machine learning algorithms do not work with categorical variables ( example : linear regression ). These algorithms require the data to be numeric. To convert Categorical variable to be numeric we have two methods:

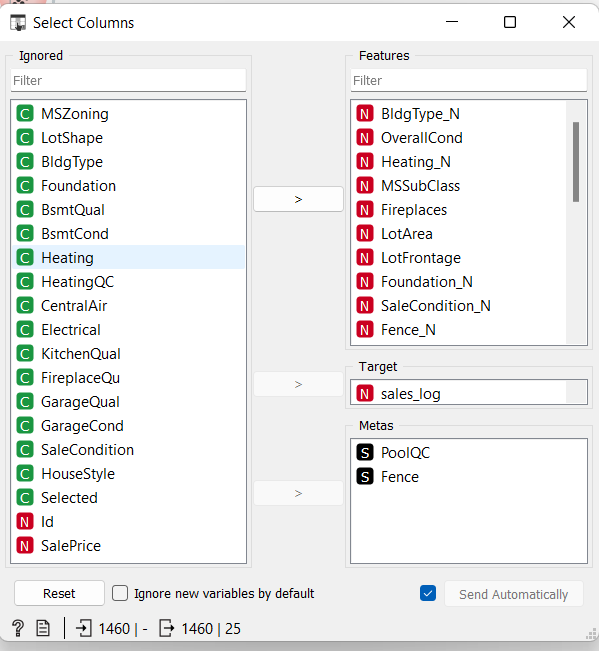
1. One-Hot Encoding
2. Simple Encoding

The use of One-Hot Encoding has already been discussed in **Task 1 (Figure 2 )**

In the figure below we are using **feature constructor** to convert all the categorical variables to numeric values.



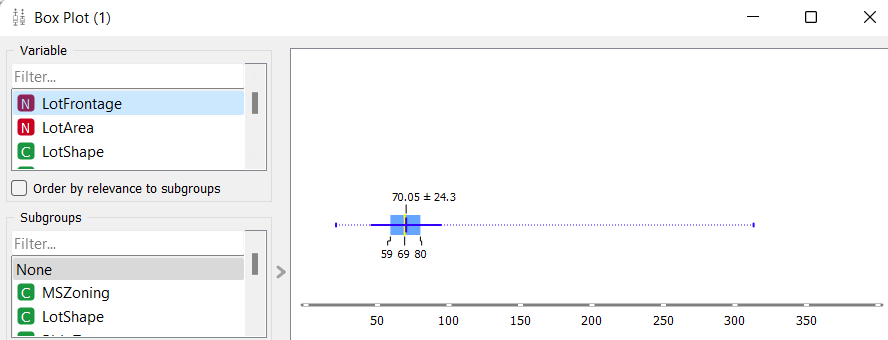
**Figure 6 - Using feature Constructor to convert categorical variables to numerical variables**

****

**Figure 7 Using Select columns widget to remove variables**

**Figure 7** shows Select Column widget that is used for elimination of non-useful variables from our data set. Select column widget takes input from the data table that has pre-processed data and then sends the selected data to a data table that stores data for further processing.

As discussed in Task 1 **LotFrontage** had 17 percent missing value but this is an important parameter and cannot be discarded so we need to impute these values . This is done using Pre-processor widget by selecting impute missing value tab as shown in **figure 2** now an important point here is that we need to impute **median values** in LotFrontage as it has outliers, and these outliers can affect the mean values hence we take median value to impute LotFrontage.



**Figure 8 Box plot used to analyse outliers for LotFrontage**

## **How Understanding of Domain support this task**

Data cleaning is one of the most important aspect in any machine learning algorithm. The rule “Garbage In Garbage out “ applies here . Using an uncleaned dataset may yield models that may break down or sometimes give very optimistic performance. It is therefore very Important to Clean the data before feeding it to a machine learning pipeline. The understanding of Domain is very useful for successfully completely this task as it helps us in recognizing which variables are to be kept for further analysis and which variables to be dropped. Moreover, to give accurate prediction, one needs to know what algorithm to be used and what kind of data is required for that algorithm. This is the reason we converted the categorical variables into numeric values.

# **Section 3**

## **Feature Engineering & Preventing Bias**

Machine learning algorithms requires data input to predict the output . Data is made up of features which are generally in form of columns. Some specific features are vital for algorithms to function properly. Therefore, feature engineering is important. Feature Engineering serves two goals.

* The performance of machine learning models can be improved by feature engineering.
* Feature Engineering prepares the input data to be compatible with machine learning algorithms.

Some of the techniques involved in feature engineering are :

* Handling Outliers
* Log Transform
* Feature Split
* One-Hot Encoding
* Scaling
* Imputation
* Correlation analysis

All the above steps have been taken with our data set as discussed in task 1 . We used **One-Hot Encoding** to convert categorical data to numerical data. We have also **imputed** the missing vales in **LotFrontage .** Special care was taken during this step. We analysed the distribution pattern of **LotFrontage** by using **Boxplot** and filled the missing value with **median** values to avoid any **bias** . This was due to the fact that **LotFrontage** had **outliers** so taking mean value would have produced **Bias** in our result. This has already been discussed in **Task 1**. Moreover, we also performed **Log Transform** on **skewed** data so that our curve doesn’t fit to an extremity and produce **biased** results.

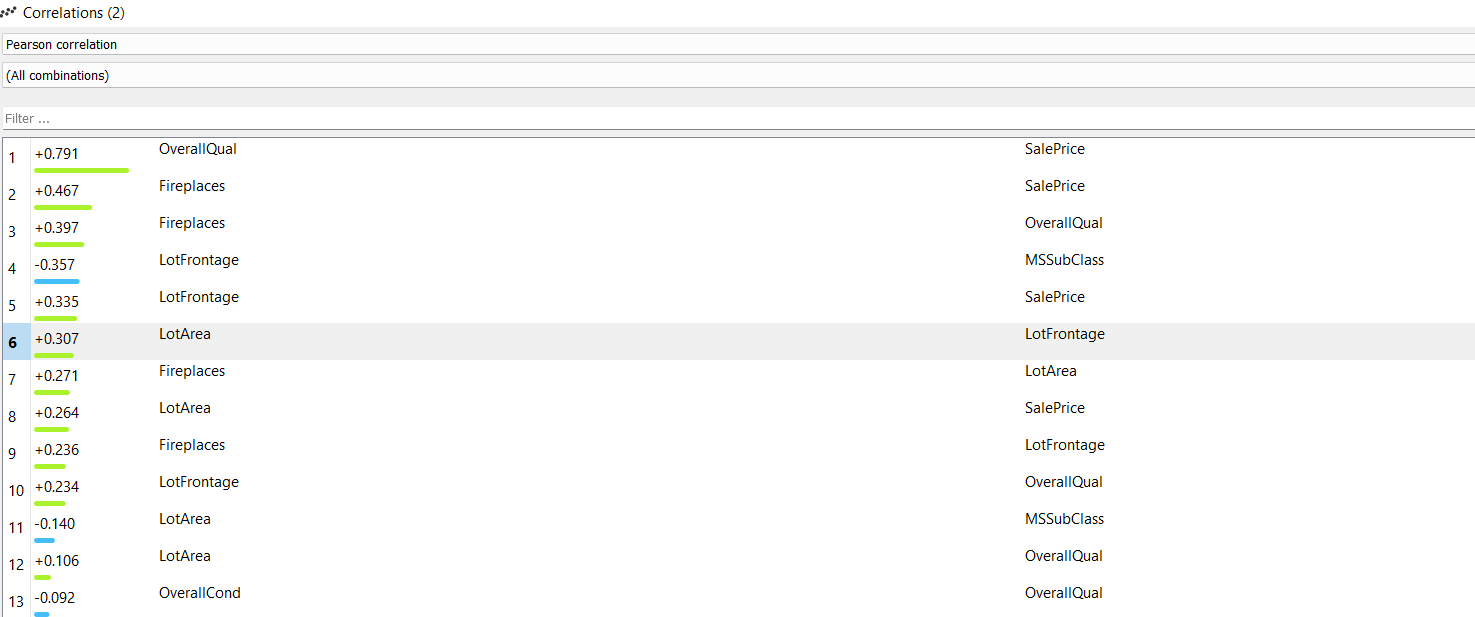
To prevent **bias**, the following steps were taken :

* Variables having low correlation with target variable were not taken for model development. Example **MSSubClass.**
* Log Transformation of skewed variables were done.
* Unknown variables were imputed with values according to their distribution.

**Correlation analysis** is one of the most important data engineering tools. It can help us in finding out the relationship between two quantities, the extent to which movement of two different variables is associated. It tells us about the strength of association between two variables. Generally, we use Pearson’s correlation Coefficient to determine the correlation between two variables. Its value can lie between -1 to +1 . Where 1 means that the two variables are highly correlated , O means that the correlation doesn’t exist and -1 means that the correlation is negative ( inverse proportion ) .

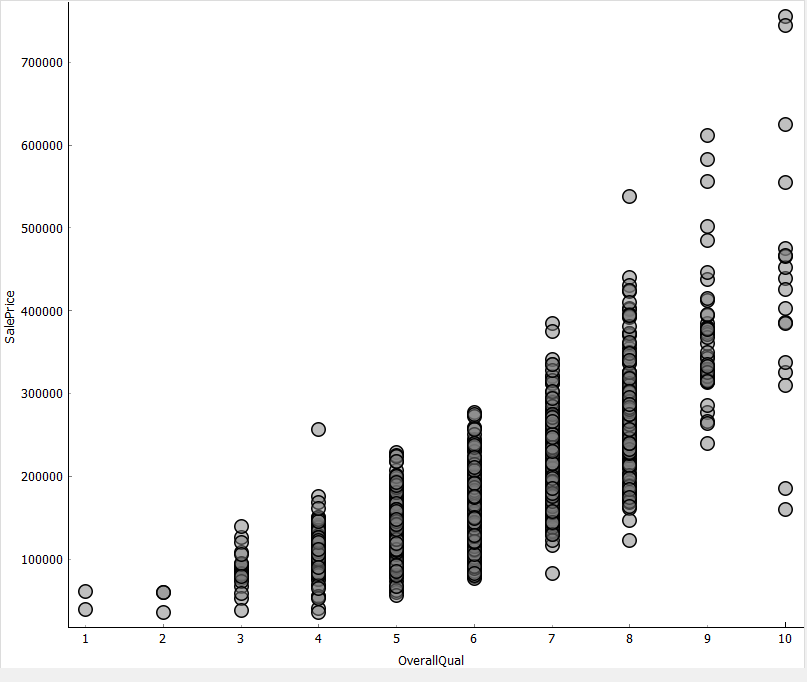
## **Correlated features in our Data set**

To determine the correlation between features in the given dataset Correlation widget is used . It yields the following result.



**Figure 9 Correlation Widget showing Correlation between different variables**

The above table shows the correlation between different variables . It can be verified by plotting a scatter chart between the two-variables showing correlation.



**Figure 10 Correlation between Sales Price and Overall Quality.**

The above figure shows a strong correlation between **SalePrice** & **OverallQual**. If the instances of the two variables plotted on x and y axis are **closely packed** that represents a **strong correlation** , however if the instances are **scattered** throughout then it represents a **weak correlation**. If the scatter plot follows the trend as shown in Figure 14 then it represents **positive correlation** however if it follows an opposite trend ( moving down as we move right side ) then this is called negative correlation.

* If a strong correlation exists between dependent variable and target variable, then we do not need to tinker.
* If two independent variables show a strong correlation ship say > 0.7 then we can either discard one of them or we can sum up the two variables.

**Which 5 variables closely correlate with target price column and using your knowledge of domain explain why ?**

**Answer** : The five variables that are highly correlated to **SalePrice** are **OverallQual, Fireplaces , LotFrontage , LotArea & KitchenQual** . These have been analysed before and after cleaning of the data. This can be very explained from the things that we have learned during Domain Analysis. For example, **OverallQual** of the property is very important parameter it covers everything from its location to the condition of the property , the building materials used . So, a buyer tends to pay more price with a property having better overall Quality. **Fireplaces** also becomes an important parameter ; the reason is normally a house tends to have fireplace in living area so if the number of fireplaces increases that means the house is bigger it has more than one living space which tends to increase its sales price. **LotFrontage** is the length of the property connected to main street so if a lager front of house is connected to the street that can indicate that house may have more area, or we can also say that it has more parking space or garden. Hence having larger value of LotFrontage tends to increase the house price. **LotArea** doesn’t needs an explanation as larger the area means more the price . **KitchenQual** also seems to be directly related to sales price as if the size of kitchen increases that indicates the house is bigger , more over a better quality of kitchen tends to indicate the overall quality of house is also good hence it’s a direct indicative of the fact that the price of the house is high.

**Explain what data features are more correlated to each other and explain why you think they are?**

**Answer:**

**Fireplaces-OverallQual** these two variables are correlated in the sense that generally a house would have a fireplace in its living space so if a house possesses a greater number of fireplaces that means it is a bigger house with a greater number of fireplaces and a bigger house would generally be afforded by rich people only, so they tend to spend more on their house which also improves Overall Quality.

**LotFrontage-LotArea :** LotFrontage is the length of the house connected to main street or road. Houses with More area would tend to have longer length connected to the road.

Discussion of other correlated features has been done in previous question and sections.

# 

# **Section 4**

## **Choosing a Machine learning Algorithm**

Linear Regression is the algorithm of choice here and there are plenty of reasons to go with this .

* It is very versatile, and it uses statical measurements to ascertain the variability in data that our model is explaining. Moreover, it also helps us in pinpointing selective features from a large set of features that holds better predictability towards target variables.
* Linear regression is one of the most transparent and simple algorithms. Unlike techniques such as Random forests which are like a black box, we can easily figure out what’s happening with our data and how our model is working.
* When using Linear Regression its easier to evaluate our model using R2 , MSE and RMSE values.
* Linear regression gives output in form of a continuous value.

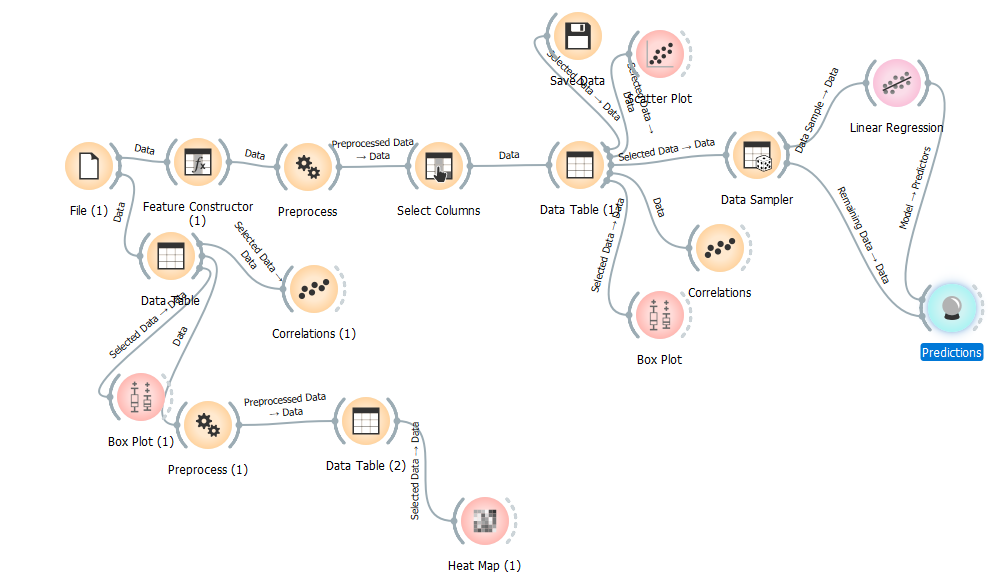
Contrary to these advantages there are some disadvantages as well :

* Linear regression is very sensitive to missing values.
* Linear regression is very sensitive to outliers. Hence normalization is required.

However, after going through advantages and disadvantages its imperative to say that Linear regression appears to be best choice.

## **Choosing Software for Model Building**

After consideration of options in hand , **orange** emerged out to be the most suitable option as orange being a GUI based software tool we can very easily build and change our models and test them very quickly whereas building models on MATLAB and python would have taken a lot more time and effort.



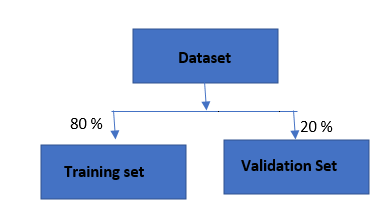
**Figure 11 Orange Pipeline used for model building**

# **Section 5**

## **Discuss how you applied cross-validation technique in the machine learning pipeline.**

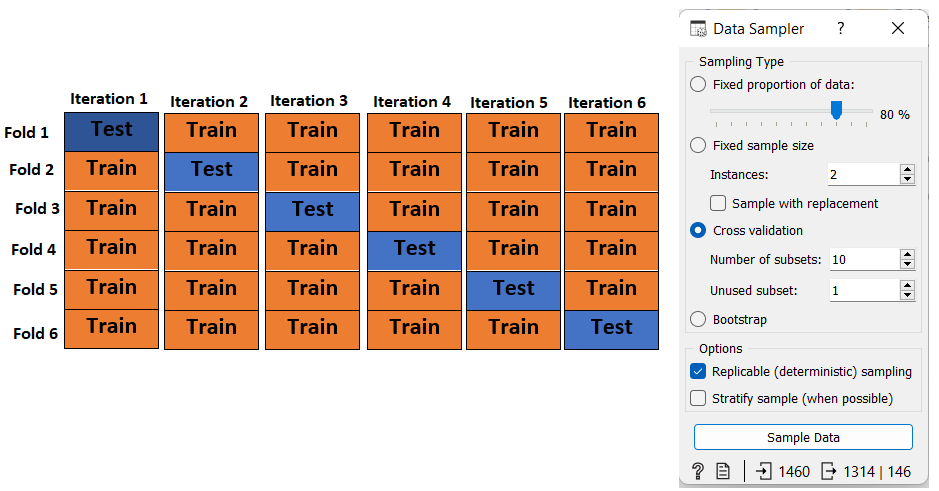
When we build a Machine Learning pipeline, we can’t just fit the model to the available data and then expect it to work fine with the real-world data. We need to test it before applying it in real life situation. We must ensure that our model gives correct output in every situation. For this we use validation techniques. There are various types of Validation techniques. And these have been discussed below: To avoid trends like overfitting or underfitting shown by our model we use parameters such as R2 and RSME to evaluate our model.

**Holdout Method :** In this technique we split our training data into two parts , The larger part is used to train a model and the remaining part is used to get prediction from the trained model. Normally we divide the data in the ratio (70:30 or 80:20) . It is one of the **simplest** types of Cross validation techniques used . The **drawback** with this algorithm is that it **sometimes gives high variance** because there is an uncertainty which instances of our data will be chosen for validation and the results can be very different for different sets. It also **can’t be relied for small datasets**.



**Figure 12 Division of dataset into Training and Validation set**

**K-Fold Cross Validation :** There will never enough data to train our model and use some of it for validation purpose unless it’s a huge dataset. So, when we divide our dataset into training and testing set there is always a risk of losing some important trend which can induce bias in our model. So, we need a model that provide enough data for training and testing, and this is where K-Fold Cross Validation comes in.



**Figure 13 K-Fold Cross Validation Technique Flow chart and Data Sampler widget**

In K-Fold Cross validation technique the given dataset is divided into K subsets , and then we apply Holdout method K times in such a way that each time one of the k sets is chosen as testing dataset and other k-1 sets are used for training purpose. The advantage of this approach is that each datapoint gets to participate in validation dataset once and used in training set k-1 times. This approach results in significant reduction in bias as all the data is used for training purpose and moreover the variance is also less as all the data is being used for validation. This method is also very effective as we interchange the training sets. Generally, the value of k is taken as 5 or 10 .

There are some other Cross Validation techniques . To name a few these are listed below :

* Stratified K-Fold Cross Validation
* Leave-P-out Cross Validation

We are not discussing other cross validation techniques in detail here as we are not using them . **For or model we chose to use K-Fold Cross Validation Technique.**

# **Section 6**

## **Learning Curves**

Our objective is to minimize the error in our machine learning model . The main source of errors is bias and variance. To build accurate model, we need to minimize these two. But both holds an inverse relationship. So, we need to optimise our model to get minimum possible values of these two . The learning curve obtained for our model are shown below :

Diagram

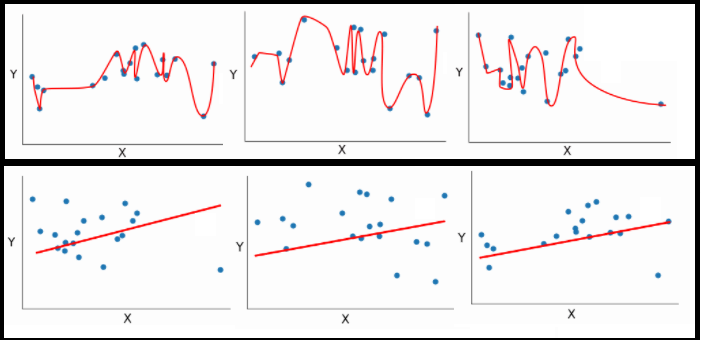
Description automatically generated with medium confidence

**Figure 14 Learning Curves obtained using python**

The interpretations that can be made from the above models are :

* When the training set size is small it is observable that MSE and RSME values for training set are 0 this indicates the modes has no problem fitting this data point. However, the validation dataset shows a very high value of MSE and RSME because its most unlikely that our trained model can fit perfectly on a single data point.
* As we increase the training data size the model’s error increases sharply while the validation error decreases in a similar manner. As we increase the training data set size our model doesn’t predict all the training points perfectly however in case of validation dataset the performance of model improves drastically.
* After 600 data points the model reaches a point where training and validation data set produces mostly the same results. This tells us that adding more data points won’t lead to any better data model.

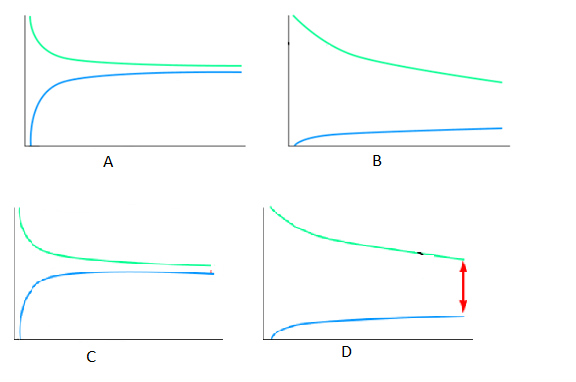
When the model fits well to almost all the data sets without much change that means it has less variance and More bias. We have tried to oversimplify the model . Such a model is called underfitting model. However, if the model has less bias it tends to change more quickly for each data set resulting in more variance . Such a model tends to Overfit. This is where the role of learning curves comes in . We need to maintain a delicate balance between variance and bias so that our models neither overfits nor underfits.



**Figure 15 The top row shows Low bias Data ( Overfit) whereas bottom row shows high bias ( underfit)**

To explain further we need to consider following points :

* If our training Curve stays low, it indicates Low bias whereas a high training error indicates high bias .
* A narrow gap between the training and validation curve indicates less variance.
* If the curves are still converging towards each other this indicates more data is needed
* If the curves converge and then diverge that means on addition of more data the variance increases.



**Figure 16 (A) High Bias (B) Low Bias (C) Less Variance (D) High Variance**

**So, from the above figures we can easily interpret that if the bias is low and variance is high then the model is overfitting on the contrary if the bias is high, but variance is low then the model is underfitting.**

# **Section 7**

The two algorithms that have been chosen to compare with linear regression are :

* K-Nearest Neighbour
* Adaboost Algorithm

**K-Nearest neighbour ( KNN) :** This is one of the most intuitive and simplest algorithms . It can be used for both regression and classification. It uses K Nearest Neighbours ( points ) in order to predict the value of new data point . It is based on “Birds of similar feather flock together “ . The principle behind KNN is nearest neighbours are the points that has minimum distance in feature space from a new data point. To implement this algorithm, we need just two things the value of K ( should always be odd) and a distance metric. The distance metric available are Euclidean distance , Manhattan distance , Hamming distance and Minkowski Distance .

**Value of K :**

* If the value of K is low, then there is overfitting ( High variance )
* At K=1 Error is always zero
* On increasing the value of error is reduced but after a certain value the again starts increasing.

**Advantages of KNN :**

* The learning time of KNN is quick .
* This algorithm is simple to understand.
* It can be used for regression and classification both hence it is very versatile.
* Accuracy is high.
* In this algorithm we don’t need to make any assumption about data

**Disadvantages of KNN :**

* The accuracy of KNN depends upon quality of data.
* This algorithm is computationally very expensive.
* The memory requirement is high as it stores all training data.
* It depends on scale of data.

**Adaboost Algorithm :** To understand Adaboost we need to Understand Boosting. In boosting a model is made from training data and then a creates a second model that tries to remove errors from the first model . In this way models are added till a perfect model is achieved. Adaboost is used to enhance the performance of Decision trees and binary classification problems. Adaboost works best with weak learners hence we use it with decision trees with just one level . Such short trees can only produce just one decision and are called decision stumps.

These stumps are left to make decision and their miscalculation rate is then fed to trained model.

The error is calculated by **error=(correct-N)/N**

Here ,

Error is the misclassification rate

Correct – is the number of successful classifications

N – total number of instances for training

**error=sum(w(i)\*terror(i))/sum(w)**

To calculate the performance of the stump

**Performance of stump = ½ ln[1-TE]/TE**

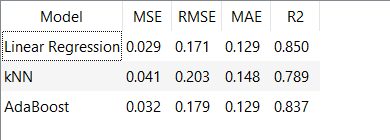
**Advantages of Adaboost:**

* Can be implemented very quickly
* Can be used for tasks like image recognition , text recognition , classification etc.
* It can be combined with any machine learning algorithm.

**Disadvantages :**

* Sensible to noise in data.

**Comparison Metric of kNN and Adaboost with Linear Regression.**

****

**Figure 17 Comparison Metric**

The performance of other two Algorithm is good but not as good is Linear regression . The reason behind this is since the dataset had (outliers ) and both the algorithm are susceptible to noise and outliers.

# **Conclusion**

From the above comparison we can conclude that Different algorithms are suitable for different task and should be selected very carefully after careful consideration of all the parameters such as the type of problem( classification or regression ) . Size of Dataset , number of independent parameters , spread of those parameters and outliers . Moreover, we should also check if the parameters follow any kind of relationship and considering all these factors an algorithm is selected.