coursework 1

by Saba Ansaria

Submission date: 19-Nov-2021 11:59PM (UTC+0000)

Submission ID: 163503376

File name: dmmi_assignment_1.pdf (1.62M)

Word count: 3591

Character count: 19058

ACS61013 Data Modelling and Machine Intelligence Course work 1 Assignment

Saba Firdaus Ansaria
Registration Number: 210110201

November 19, 2021

Contents

1 Abstract

Data analysis is a modern-day necessity which is data-centric learning and decision making. In this era of machine learning, it is very important to prepare and manage data in a proper way. If the data is not prepared in a correct way, it may lay some inherent bias in the underlying model, which can cause significant damage to the model outcome. Therefore, data analysis is a collection of some structured processes such as data collection, data processing, bias removal, feature engineering, model training, regularisation, filtering etc. Understanding the pipeline of these machine learning algorithms is very crucial for learning data, its attributes, and domain. The goal of this assignment is to explore these processes and learn each of these steps with in-depth analysis.

2 Introduction

A machine learning pipeline is a structured collection of processes, each consisting of some particular machine learning algorithm. The algorithms are chosen based on their nuances and the data they are going to process. Different type of data has different nuances and complexities. They have different biases and risks associated with them. For example, handling health oriented data that contains personal information about different people will be much more sensitive than handling housing data. Therefore, an effective machine learning pipeline removes as much bias and sensitive information so that the sensitive information becomes blurred and the data becomes more neutral.

In this following report I explored the housing dataset and understand its domain, characteristics and nuances (Section 3, 4. Based on the understanding, I pre-processed the data, filtered biased information and the irregularities (Section 5. After preprocessing, I examined the correlation among the features to understand the data better and separate the outliers (Section 6. The feature engineering leads to the modelling part where I train a machine learning model with the data (Section 7. The machine learning model is evaluated and necessary measures are taken to restrict overfitting and underfitting condition (Section 8, 9). Furthermore, I compared the model with other two machine learning models and presented their similarities, differences, advantages and disadvantages (Section 10). Finally, I drew conclusion based on the experimental evidence I had in the previous sections.

3 Dataset

The dataset is a house pricing dataset that has house prices with their corresponding categorical and continuous feature attributes. The dataset contains 1460 observations and 26 features. The CW_ dataset.csv is the initial dataset without any modification or feature engineering.

4 Domain Analysis (Level 1)

Conduct a domain analysis and present your findings as related to the domain of the coursework. Discuss how what you have found from your domain analysis will support and be carried over to other parts of your coursework.

Orange framework has been used here for importing the data and visualising the distribution and the correlations. The *CSV File Import* widget and the *Distributions* widget have been used. The data set describes the quantities and qualitative attributes of a property. Generally, a typical buyer had concerns with the attributes of a property such as

- What type of property it is?
- What is the quality of the location of the property?
- How are the amenities (electrical, heating, water supply, pool, fireplace etc)?
- What is the condition of the kitchen, bathroom?
- Is there a garage or parking space, and what is the condition?
- What is the total area and the overall condition of the property?

These questions have been answered in terms of attributes in the data set. The data set contains 26 variable attributes such as Id, MSSubClass, MSZoning, LotFrontage, LotArea, LotShape, BldgType, HouseStyle, OverallQual, OverallCond, Foundation, BsmtQual, BsmtCond, Heating, HeatingQC, CentralAir, Electrical, KitchenQual, Fireplaces, FireplaceQu, GarageQual, GarageCond, PoolQC, Fence, SaleCondition, SalePrice. The target variable is 'SalePrice'.Clearly some of the variables are categorical and some of the variables are continuous numerical values. Also, there are variables related to the heating, electrical systems and other commodities of the house. It is evident that these variables are dependant on each other. Clearly the overall quality and the location of the property should have high influence on the pricing of the property. From the distributions, it is evident that most of the houses are 1-story single family detached and situated at residential low-density areas. 'HouseStyle' and 'MSSubClass' both variable share similar attributes of the property. Most of these properties have gas heating systems with ranging from excellent to average quality. A huge portion of these properties has central air condition. The 'GarageCond' and 'GarageQual' convey same information. However, in some places they contradict each other (possible outliers). In some cases, the 'OverallCond' and 'OverallQual' contradict each other. However, this contradiction is natural because the overall quality consists of material quality, amenities and other utility stuff but the overall condition also includes the age and the location of the property. The presence of fireplace and good kitchen also influence the price of the property. The bottom line is that the price of a property varies on multiple variables and some of them has been discussed earlier.

Based on this domain knowledge, it will be easier to understand the inter-dependency and correlation of the variables in this data set. The understanding of the domain is useful to construct an efficient pre-processing process in the machine learning pipeline, which is discussed in the next section (Section 5).

5 Data Pre-processing (Level 2)

Achieve level 1 as well as conduct data cleaning and pre-processing. Discuss how you used your understanding of the domain from level 1 to support this task.

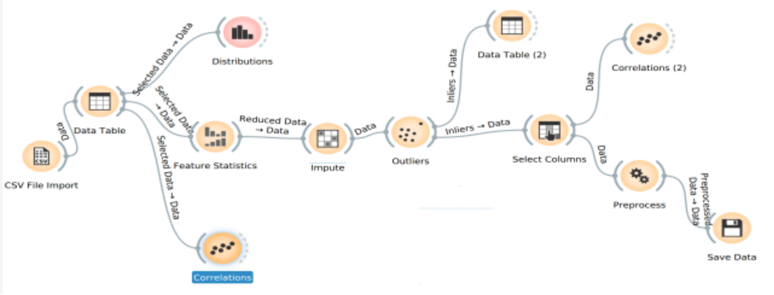
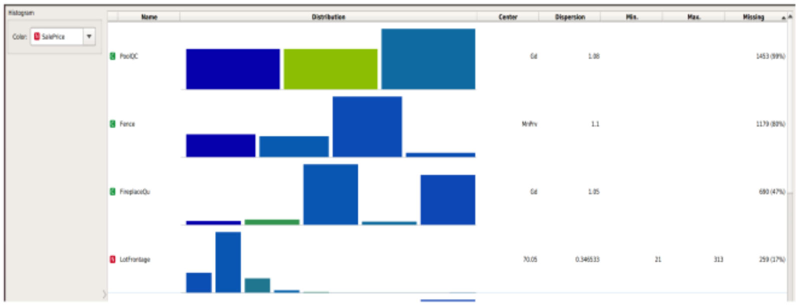


Figure 1: Preprocess Pipeline Orange

The data pre-processing is also done using *Orange* framework.

I have imported the csv file with import widget and used the 'Feature Statistics' widget to analyse the number of missing features. It is found that 99% of *PoolQC* data, 80% of the *Fence*, 47% of *FireplaceQu*, 17% of the *LotFrontage*, 5% of the *GarageCond*, 5% of the *GarageQual*, 2% of the *BsmtCond* and 2% of the *BsmtQual* data is missing. If the rows are discarded due to data insufficiency, we will lose 1455 samples, which should not be the case. Therefore, the variables with more than 45% missing data has been discarded because they consists of minimum of maximum 1455 number of data samples. So we lose few columns but not the whole data. Also the distributions of those variables w.r.t main target variable *SalePrice* are not normal distribution. Hence, I discarded *PoolQC*, *Fence*, *FireplaceQu*. Furthermore, the rest of the varibales where data is missing about less than equal to 5% data is imputed using the *Impute* widget to replace the missing items with avarage/most frequent occurances. After this, there is no significant change on the data distribution on those variables. The data distribution for *LotFrontage* remains the same with 70.05 centre and 0.3 dispersion, which indicates that the distribution on that variable was already biased.

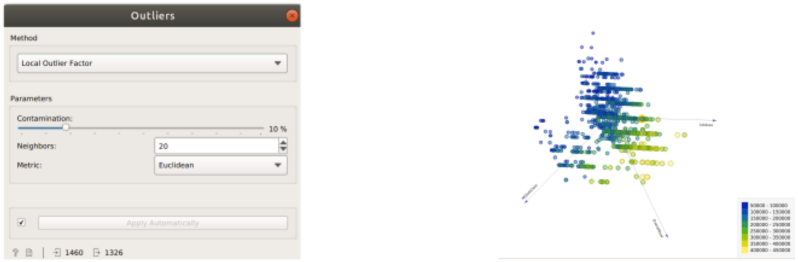

Histogram

Color: SalePrice

Name	Distribution	Center	Dispersion	Min.	Max.	Missing
PoolQC		Gd	1.08			1453 (99%)
Fence		MnPrv	1.1			1179 (80%)
FireplaceQu		Gd	1.05			690 (47%)
LotFrontage		70.05	0.346533	21	313	259 (17%)

Figure 2: Histogram of features with missing data

After this the contradicting and biased samples need to be removed upto a threshold. Euclidean distance has been considered along with 10% contamination threshold while considering 20 neighbour points at the same time. The *Outliers* widget has been used to discard some negative samples.



Outliers

Method

Local Outlier Factor

Parameters

Contamination: 10 %

Neighbors: 20

Metric: Euclidean

☑ Apply Automatically

1460 1326

(a) Outlier detection and removal in Orange. (b) Linear projection after outlier removal.

Figure 3: Outlier removal and linear projection of features.

6 Feature Engineering (Level 3)

Achieve the previous levels plus discuss the steps taken in feature engineering (e.g correlation analysis) and preventing bias in the dataset to be used to training the machine learning algorithms. Answer the following questions:

- **Which data features are more correlated to each other and explain why you think they are.**
- **Which 5 variables closely correlate with the target price column and using your knowledge of the domain (Hint: Use your domain analysis), explain why**?

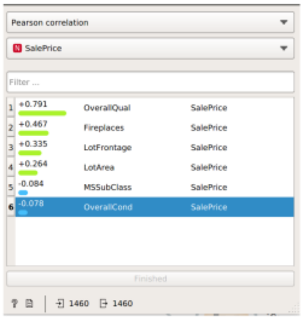
The correlation among the features in the data is shown in Table 6. As discussed in Chapter 4, the feature variables are closely related to each other. The quality of house amenities influences the property values and its quality hugely. The values of *OverallQual* and *Fireplaces* have a high positive correlation of 0.397. The area of the property is correlated with the area adjacent road area with a coorelation value of 0.307. These correlations are quite understandable from the domain analysis. Also the area and fireplace are a deciding factor for the final target price. All these area and amenity related attributes influence the overall quality of the property as seen from Table 6. However, in some places they bear negative correlation. For example, fireplaces are used to be in the old buildings, therefore the properties with fireplaces are old and as a result the overall quality is degraded over time.

The correlation analysis w.r.t target price is done in two stages. Because it was interesting to see the correlation between the dependent variables before and after pre-processing & feature engineering. The first correlation analysis is conducted at the beginning of the pipeline and the second correlation analysis is conducted after pre-processing and feature engineering. The discreet variable has been converted to a continuous variable using the 'Preprocess' widget where two functions have been used, i.e, continuize discreet values and normalising the values. I assigned the most frequent values of the discreet feature value as a new continuous variable. The correlation at the beginning and after preprocessing is shown in Figure 4.

The five features such as OverallQual, Fireplaces, LotFrontage, LotArea, MSSubClass hold highest correlation with the target sales price. While observing the distribution, it is found that some of the varibales are not well distributed and it may provide inherent bias to the data. So all these variables are made continuous with zero mean and one

Correlation	Feature 1	Feature 2
0.791	OverallQual	SalePrice
0.467	Fireplaces	SalePrice
0.397	Fireplaces	OverallQual
-0.357	LotFrontage	MSSubClass
0.335	LotFrontage	SalePrice
0.307	LotArea	LotFrontage
0.271	Fireplaces	LotArea
0.264	LotArea	SalePrice
0.236	Fireplaces	LotFrontage
0.234	LotFrontage	OverallQual
-0.14	LotArea	MSSubClass
0.106	LotArea	OverallQual
-0.092	OverallCond	OverallQual
-0.084	MSSubClass	SalePrice
-0.078	OverallCond	SalePrice
-0.059	MSSubClass	OverallCond
-0.053	LotFrontage	OverallCond
-0.046	Fireplaces	MSSubClass
0.033	MSSubClass	OverallQual
-0.024	Fireplaces	OverallCond
-0.006	LotArea	OverallCond

Table 1: Overall correlation between the feature variables.



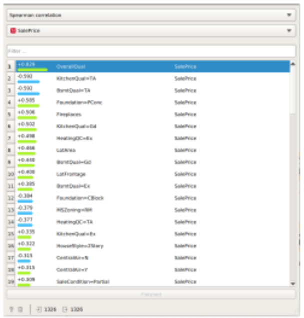
Pearson correlation

SalePrice

Filter ...

1	+0.791	OverallQual	SalePrice
2	+0.467	Fireplaces	SalePrice
3	+0.335	LotFrontage	SalePrice
4	+0.264	LotArea	SalePrice
5	-0.084	MSSubClass	SalePrice
6	-0.078	OverallCond	SalePrice

(a) Feature correlation before pre-processing.



Spearman correlation

SalePrice

Filter ...

1	+0.829	OverallQual	SalePrice
2	-0.592	KitchenQual=TA	SalePrice
3	-0.592	BsmtQual=TA	SalePrice
4	+0.585	Foundation=PConc	SalePrice
5	+0.506	Fireplaces	SalePrice
6	+0.502	KitchenQual=Gd	SalePrice
7	+0.498	HeatingQC=Ex	SalePrice
8	+0.466	LotArea	SalePrice
9	+0.440	BsmtQual=Gd	SalePrice
10	+0.400	LotFrontage	SalePrice
11	+0.395	BsmtQual=Ex	SalePrice
12	-0.384	Foundation=CBlock	SalePrice
13	-0.379	MSZoning=RM	SalePrice
14	-0.371	HeatingQC=TA	SalePrice
15	+0.335	KitchenQual=Ex	SalePrice
16	+0.322	HouseStyle=2Story	SalePrice
17	-0.315	CentralAir=N	SalePrice
18	+0.315	CentralAir=Y	SalePrice
19	+0.309	SaleCondition=Partial	SalePrice

(b) Feature correlation after pre-processing.

Figure 4: Feature variable correlation.

standard deviation. After the transformation, the correlation with target price makes lot more sense. This time *OverallQual, Fireplaces, Kitchen=Gd, HeatingQc=Ex, Foundation=Pconc* and along with other continuous highly correlated with the housing target price. This seems more logical. Now the data dependency is well distributed and correlated with more that five variables (Figure 4). Also from the other correlated features it is evident that the housing price is well adjusted with independent housing amenities.

7 Machine Learning Model (Level 4)

Achieve all the previous levels as well as explain how:

- **You decided on the choice of the best machine learning algorithm to apply to the problem.**
- **You used Orange, MATLAB, Python or a combination of tools to develop an effective machine learning pipeline from data cleaning up to the point of evaluation**

The target house price is a continuous variable. Therefore, to predict the target house price from another variable would be a regression task. The number of sample population is less than 1500. Hence, neural nets would not be a good option because neural network are data hungry. Also, tree or random forest based regressors would not be a great choice either (Shown as a comparison in Chapter 10). Linear regression has been chosen as the preferred model for this regression task here due to its simplicity and quick execution.

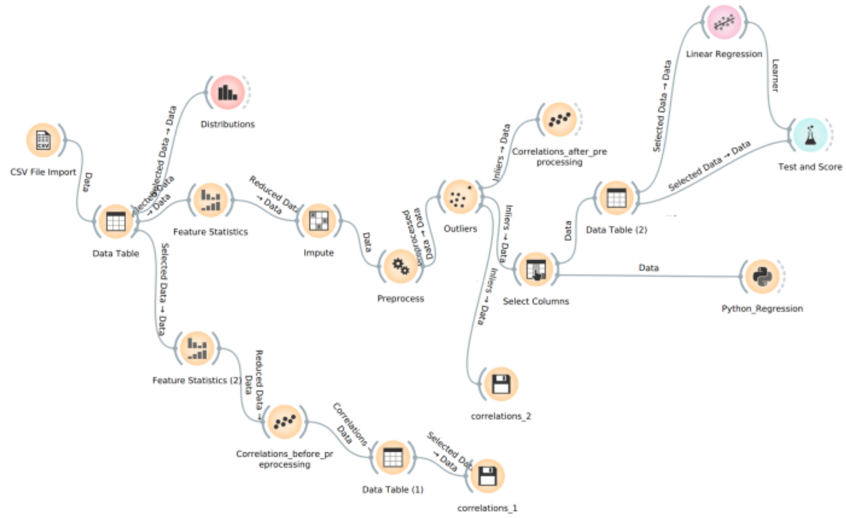


Figure 5: Regression Pipeline Orange

I have used Orange framework to build the data pipeline and its widgets to model and train the data as well. I separately used the python widget for running my python program for training another regression model as well. The pipeline is shown in Figure 5. The step by step process of building the pipeline is given below

Step 1: Add the *CSV File Import* widget in the Orange work space and import the csv file 'CWdataset.csv'.

Step 2: View the data on the *Data Table* by adding the widget and connecting the data-flow from the *CSVFile Import* widget.

Step 3: Add the *Feature Statistics* widget and connect the data pipeline from the previous *Data Table* module. The feature distribution can be seen by double clicking on the *Feature Statistics* module.

Step 4: Open *Feature Statistics* and select the features where more than 80% data is present for overall samples. Send them to the next module by clicking on 'Send selection'. By default, it sends automatically.

Step 5: Add the *Impute* widget and connect the data pipeline from the previous *Feature Statistics* module. For this assignment I impute the missing data with the most frequent feature value present within that feature scope. So, double click on the 'Impute' module and select the method as 'Average/Most frequent'.

Step 6: Next, we need to add the Preprocess widget to convert the discreet features into continuous features. After adding the data line from *Impute* into Preprocess, double click on the Preprocess module and double click on the option 'Continuize Discrete Variables' and select 'Most Frequent is base'. Now apply the preprocessing (I have tried normalising the data to avoid bias in the distribution but it did not make much difference to the model).

Step 7: After the preprocess is done, we are left with the same number of sample size. We have removed some features in the *Feature Statistics* module but the total number of samples is the same as the beginning.

Step 8: Now add the *Outliers* widget to remove some samples with with contamination factor of 10% and considering 20 neighbours for each sample. Add the data flow from the Preprocess module to the *Outliers* module and apply the mathod. This method removes 134 samples from the population. I have added *Correlation* widget at different steps and analysed the correlations among the features (discussed in the previous section).

Step 9: Add the *Select Columns* widget and select 'SalePrice' as the target variable. This makes the data ready for the model learning part. Add all the data flow connections shown in Figure 5.

Step 10: Add the Linear Regression model widget from the menu and Test and Score module as well. Connect the data lines from *Select Columns* to Linear Regression and *Select Columns* to Test and Score.

Step 11: Open Test and Score and select Cross Validation option with 5 folds.

Step 12: Add a Python Script widget and connect the dataflow from *Select Columns*. Now add the python script to run a separate regression model to visualise the learning curve with different regularization.

Step 13: The model is evaluated using mean square error. 5-fold Cross validation is used both in Orange Regression model and my python script based regression model. The code is supplied and the learning curve is shown in Figure 7.

8 Cross validation (Level 5)

Achieve all the previous levels plus discuss how you applied cross validation techniques in the machine learning pipeline.

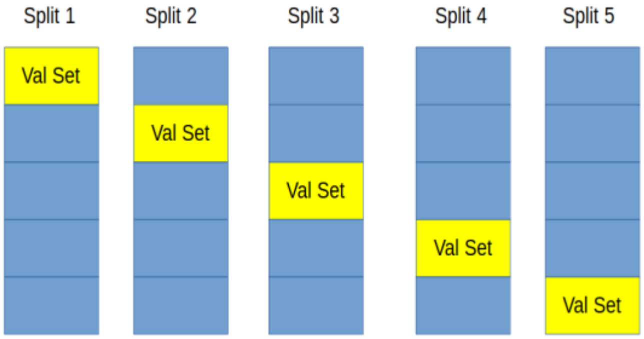


Figure 6: 5-fold cross validation

In this assignment, the K-fold cross validation regime has been used. In this case k=5 (Figure 6), and this 5 fold cross validation technique chunks the whole data sample population into 5 chunks. Each chunk is left out as a test set where the rest of the four chunks are used as train sets. The train and test sets are divided with 80%/20% ratio. Eventually, in five different iterations, each of the 5 different chunks are used as test sets for once. By this regime, we can use the whole data set and it reduces any biases in the data set. As a result, the evaluation gives the whole picture of the data set. In the Orange framework Test and Score module has been used and the 5- fold cross validation is applied on that. In the python regression model, the scikit learn cross_validate module has been used. This evaluation technique gave an average representation of the whole data set evaluation by averaging over the 5 different evaluation split model training and testing. No training data is mixed with test data in any evaluation.

9 Learning (Level 6)

Achieve all the previous levels as well as discuss how effective your pipeline is at preventing overfitting and underfitting through the application of learning curves.

The pipeline deploys a series of precise data preprocessing steps that handle the missing data, bias and data distribution. In Chapter 7, at step 4 the pipeline removes some features with a huge missing/skewed value. At step 5,6 the pipeline imputes missing data as well as make the feature continuous. Continuous feature space is very crucial for training regression model. Otherwise the learning algorithm makes discrete labels continuous at training time. The proposed data pipeline path handles that well. As a result the resulting model is less prune to overfitting.

In the python script, three different linear regression models have been run. One with no regularisation and the other two with ridge and lasso regularisation. The learning curve is shown in Figure 7. The script is provided with this assignment. Certainly, the number of samples clearly affects the regression model. I have tried with sample sizes 100, 200, 300, 400, 500, 600, 800, 1000 and the learning graph at Figure 7 clearly shows that the higher number helps the training and prevents underfitting. Furthermore, the regularisation methods helped the normal regression model slightly to overcome the overfitting problem.

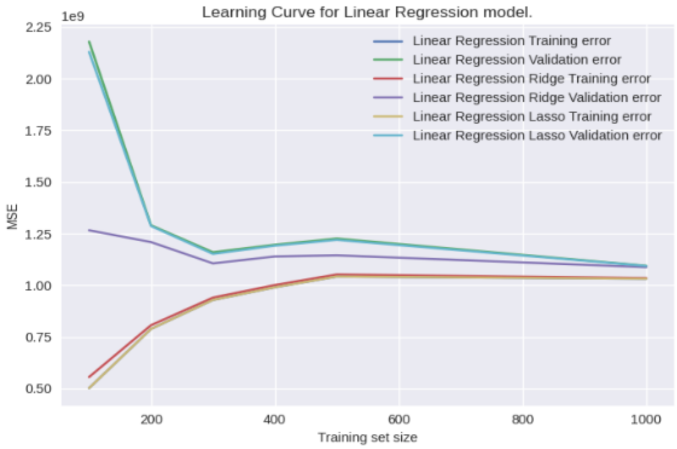


Figure 7: Linear Regression Learning Curve Train vs Validation

The ridge regression penalises the coefficients if they are big on size and as a result coefficients minimize a penalized residual sum of squares. Lasso regression also adds a L1 regularisation term with the least squares cost function for regression. However, it tries to estimate the sparsity by finding nonzero coefficient solutions with less number of coefficients. As a comparison, the random forest regression learning curve also shown in this report in Figure 8. The significant difference between these two models are noticable.

10 Comparison (Level 7)

Achieve all the previous levels and the below:

- You can compare your choice of machine learning algorithm with at least two other algorithms that we have not covered in class.
- Discuss the mathematical peculiarities of the algorithms you have chosen (strengths and weaknesses) and how they impact the results you obtained.
- Apply the appropriate metrics to compare the algorithms you have chosen with the algorithms we have discussed in class.

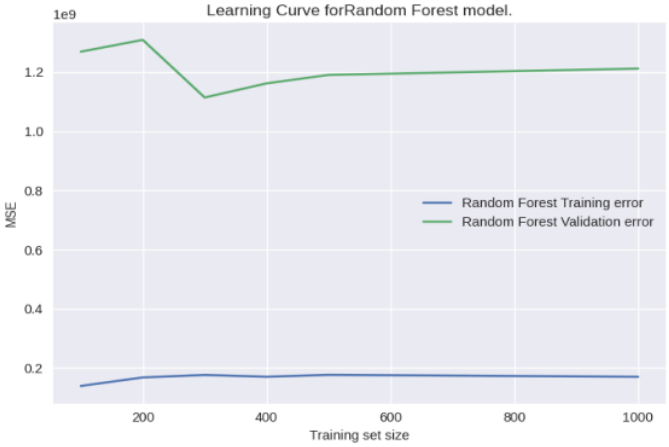


Figure 8: Random Forest Regression Learning Curve Train vs Validation

- Discuss the effects of model complexity of the chosen algorithms on the learning curves generated.

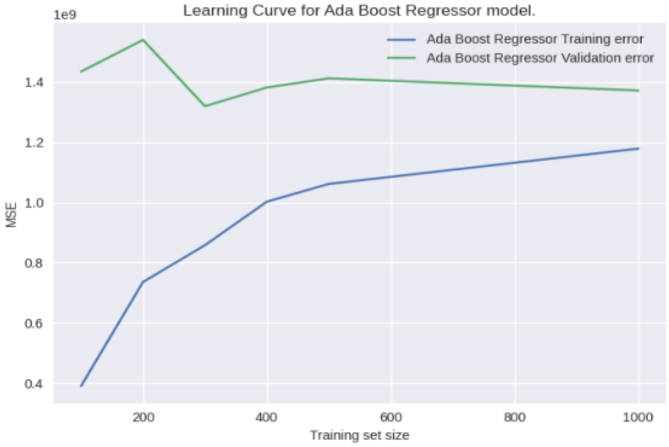


Figure 9: Adaboost Regression Learning Curve Train vs Validation

The Linear regression model is compared with two regression models, which are Adaboost and Gradient Boost regression. Both ada boosting and gradient boosting came from a family of algorithms which is called boosting. Boosting consists of a set of weak learners, contrasting with the linear regression where a set of linear coefficients fit the data. In boosting the set of weak learners are trained on the data and then the outcome of those weak learners are averaged in a sequential mixture of experts technique. In adaboost the model weights are re-weighted in iteration and the final weighted output of

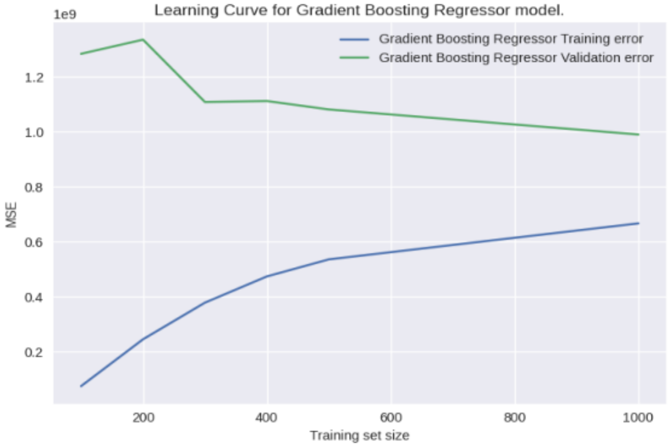


Figure 10: Gradient Boost Regression Learning Curve Train vs Validation

all sub learners decide the final output. Gradient boosting uses gradients to train the sublearners where the loss is back-propagated.

Adaboost is sensitive to outiers where Gradient boost is more precise and robust w.r.t outliers from an optimisation point of view. In both Figure 9 and 10 it is clearly evident that the model needs more data samples for training. The collection of weak learners and the optimisation strategy makes the boosting algorithms data hungry and they are mostly used with other weak learners. Between 9 and 10 it is evident that gradient boosting needs more data than ada boosting. It is because of the gradient based optimisation function. However both of these models show high variance problem compare to the linear regression model.

11 Conclusion

In this report, I have discussed a machine learning data pipeline with regression modelling. The nuances of data preprocessing and feature engineering have been discussed with Orange framework and Python programming language. Linear regression learning has been discussed with learning curves along with two different regularisation techniques. For model comparison, boosting regression algorithms have been shown with the learning curves. The graphs clearly show the efficiency of linear regression in low population data set.

coursework 1

ORIGINALITY REPORT

11%	1%	1%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Sheffield Student Paper	11%
2	tools.ietf.org Internet Source	<1%
3	www.forbes.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

coursework 1

GRADEMARK REPORT

FINAL GRADE

68/100

GENERAL COMMENTS

Instructor

Yes, there are many features that affect house prices but what are the key ones to bear in mind as you develop your pipeline? Discarding the columns is not really a good idea. This is because NA might not mean the data is missing. It might mean that it is not applicable since not all houses have pools. Sorry, it is not really clear (table 6. Where is table 6??) which features correlate to each other in your explanation. Good thinking about not using Neural Networks. Though, I am not sure about the use of regression because the generated model could vary widely. Perhaps that is why you got very large MSE. The way you have used the k-fold/hold-out validation is not completely right. Please check the lecture notes.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12