# **INFO 5505 Applied Machine Learning for Data Scientists**

## Assignment 1

Name: Ravi Varma Kumar Bevara

### **PART I: Dataset Description**

### Explanation of the meaning of each parameter in the Data, and describe the relationship between training datasets, validation datasets, and testing datasets if have. Describe the data cleaning, transformation, and normalization if have.

The dataset used for this assignment, is provided by the professor in the canvas portal. After looking at the dataset, I came to an assumption that it is related to data on sales of Monet paintings. In total, it has 430 observations, and 6 attributes. From the assignment description, we can see that we are predicting ‘Price’ in this dataset using linear regression machine learning model. So, considering that price is the dependent variables and rest of the attributes are independent variables.  
  
Price – Sale Price where in which the painting is sold. (Unit not mentioned)

Height – Height of the painting that is sold. (As inches is the primary unit, I assumed it as inches)

Width – Width of the painting that is sold. (As inches is the primary unit, I assumed it as inches)

Signed, Picture, House – Based on the uncertainty of these variables, I decided to not include them in the model training.

For the decision regarding the division of dataset into training, validation, and test subsets, as the data is not comparatively big enough, I decided to avoid the validation step. So, following the industry standards, I took 70% of the data into training and rest 30% to testing the predictions. The relation between them is simple as like training data is the set of samples in which the machine is learning or actually training according to the data. Test data is used to measure the goodness of the fit, how much variance the model can explain the dependent variable can fit.

### **PART II: Exploratory Data Analysis**

To start, I decided to use the “Google Colab” as my programming IDE for the tasks. Also, I decided to store my dataset on the google drive. So, the first step is to import various python modules which can be used to better understand the data. Then, I mounted the google drive for accessing the dataset.

A picture containing application

Description automatically generated

To better understand the variables, we need to look how the data is distributed and what’s the shape of the data. I continued with exploring the columns of the data. In this process, I recognized the attributes “HOUSE” has a space associated with it in the column name. The modules NumPy and pandas has various functions such as .head() to check the structure of the data. Also .columns will result all the columns of the dataset.

Table

Description automatically generated

Once, I came to know about the variables, then I wanted to get a better understanding of each variable in the dataset separately.

Table

Description automatically generated with medium confidence

.describe() will be summarize all the descriptive statistic attributes such as mean, count, standard deviation, min and max of numeric variables.

For dealing with the null values, I simply used the .dropna() for dropping all null values, and as the number of observations remain the same, I can conclude that there are no missing values in the dataset.   
Table

Description automatically generated with medium confidence

To reduce the processing time, I decide to remove the redundant columns which will not be used in   
The next step is to observe how each variable is related to the rest of the independent variables. We can use the correlation matrix concept for that, as it describes the relationship between two variables, and I used seaborn heatmap visualization to better understand it as it reflects in matrix view.

Graphical user interface, application, Teams

Description automatically generated

Also, to deal with outliers, I decided to plot the scatter plot for each comparison of dependent variable and independent variables. Also, to view all the comparisons at once, I made use of seaborn pair plot which creates a comprehensible scatterplot between all the variables in the dataset.

Chart, scatter chart

Description automatically generated

To observe the distribution of each variable separately, I made use of the best available plot i.e., histogram from the module matplotlib. From the results, we can conclude that all the independent variables are normally distributed expect the dependent variable ‘Price’.

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

A picture containing window

Description automatically generated

As we can see from the correlation plot, height and width are equally correlated to each other and also, they are positively correlated to price, which is the target variable in training the model. So, I decided to create a new independent variable using height and width. I just multiplied them to obtain a new variable ‘size’. I did the same with this variable by plotting the histogram, I can clearly see that it is positively skewed and to resolve this issue, I decided to use ‘Logarithmic Transformation’ of the variable size using .log functionality from numpy module.

Chart, scatter chart

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Then, I confirmed my assumption with the use of correlation matrix. The results can clearly state that the transformed size is also positively correlated to the dependent variable ‘Price’.

Chart, treemap chart

Description automatically generated

### **PART III: Model Training and Validation**

1. Explanation of the machine learning models that you implemented including the structure of the models, parameters, training process, and cost functions.

After a deep study in the packages available for model training in python, I decide to use sklearn also known as Scikit-Learn to train my simple and multivariate linear regressions.

Graphical user interface, text, application

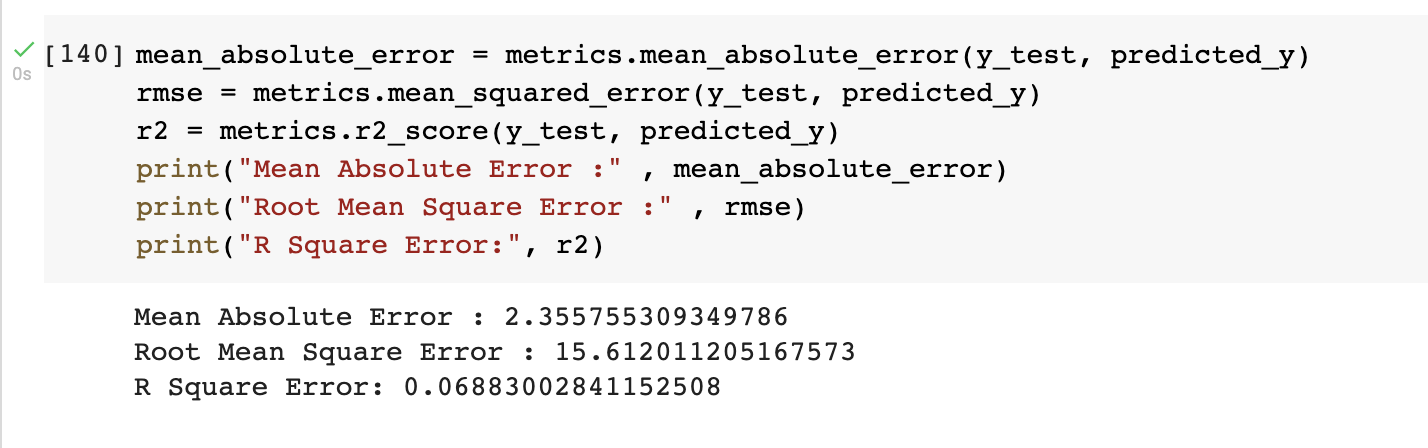
Description automatically generated with medium confidence

Linear Regression is a supervised machine learning algorithm where the prediction will be a continuous variable and used to predict the numerical in a range. As mentioned, simple linear regression uses traditional slope-intercept form i.e., y=ax+b whereas the multivariate linear regression will have multiple independent variables predicting single dependent variable i.e., mathematically represented as y=ax1+bx2+c

Graphical user interface, application

Description automatically generated

For this model, the bias would be -0.6 and the slope is 0.13. According, to this foe every 1 increase in x, there is an increase of 1.13 in y. From the results, we can see that it is having an MAE of 2.35 which is too big for error rate and R2 value is 0.06 which means it is explaining only 6% of variability in dependent variable. Low R2 value doesn’t mean to be bad model, as it is giving how the fit is, but not considering the error rate of prediction.



Graphical user interface, application

Description automatically generated

For this model, the bias would be -1.3and the slope is 0.14. According, to this foe every 1 increase in x, there is an increase of 1.14 in y. From the results, we can see that it is having an MAE of 2.48 which is too big for error rate and R2 value is 0.08 which means it is explaining only 8% of variability in dependent variable. Low R2 value doesn’t mean to be bad model, as it is giving how the fit is, but not considering the

error rate of prediction.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

For this model, the bias would be 1.02 and the slope is 0.002. According, to this foe every 1 increase in x, there is an increase of 1.002 in y. From the results, we can see that it is having an MAE of 2.4 which is too big for error rate and R2 value is 0.078 which means it is explaining only 7.8% of variability in dependent variable. Low R2 value doesn’t mean to be bad model, as it is giving how the fit is, but not considering the error rate of prediction.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

For this model, the bias would be -15.99 and the slope is 2.88. According, to this foe every 1 increase in x, there is an increase of 3.88 in y. From the results, we can see that it is having an MAE of 2.46 which is too big for error rate and R2 value is 0.09 which means it is explaining only 9% of variability in dependent variable. Low R2 value doesn’t mean to be bad model, as it is giving how the fit is, but not considering the error rate of prediction.

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated

For this model, the bias would be -8.99 and the slope is 1.38. According, to this foe every 1 increase in x, there is an increase of 2.38 in y. From the results, we can see that it is having an MAE of 0.8 which is not that too big for error rate and R2 value is 0.25 which means it is explaining only 25% of variability in dependent variable.

Graphical user interface, text, application

Description automatically generated

For both the models, we will be using MSE, which is mean square error method as the cost function. It measures the average squared difference between an actual value and predicted value. The goal of this algorithm is to minimize the MSE to improve the accuracy of the model. We can also think out of the frequentist’s perspective i.e., in the probabilistic perspective, we can employ the Maximum Likelihood Estimation as the cost function.

Text

Description automatically generated with low confidenceGraphical user interface, text, application, email

Description automatically generated

I used MLE and OLS both in my approach, but the later one to justify the best model from my approach. For the error prediction of the data, scikit learn has a module named metrics from which we can calculate the metrics of model such as goodness, fit to the dependent model etc. It is basically the difference for actual value of test data and the predicted one from the trained model. We can also see that the best model from the sklearn module, which is both transformed price and size, is matching this automatic selection.

### **PART IV: Multivariate Linear Regression**

### For the multivariate linear regression, we will be fitting two or more independent variables to one target variable. As price is the target variable here, we will be using the height and transformed size for my pair of independent variables. The reason why I choose them is they have best correlations with the price in the positive direction. But after describing the statistical description of data, I could see that the minimum and maximum values of both attributes are way different, which is the reason I choose to employ the z-score normalization to the data. Proceeding further, I dropped all other variables from the data frame, used the z score normalization as that would bring the data to be normally distributed data.

### Graphical user interface, text, application, email Description automatically generated

For this model, the bias would be -0.6 and the slope is 0.13. According, to this foe every 1 increase in x, there is an increase of 1.13 in y. From the results, we can see that it is having an MAE of 2.35 which is too big for error rate and R2 value is 0.06 which means it is explaining only 6% of variability in dependent variable. Low R2 value doesn’t mean to be bad model, as it is giving how the fit is, but not considering the error rate of prediction.

### Graphical user interface, text, application, email Description automatically generated **PART V: Results**

After training and testing almost 6 models of linear regression the given dataset, it is time to choose the best model, by using the metrics from the scikit learn. Accuracy might be the best one when it comes to classification, but for regression there are some other metrics. As the regression algorithm will predict a numerical value which is continuous variable in a range, the best metric would be to check how far is the predicted value from the actual value. Based on this criteria, R2 value, MSE (Mean Square Error), and MAE (Mean Absolute Error) are the metrics that will suite this type of algorithm.

However, R2 value will determines the variability of dependent variable or target variable of the model. But it does not count the overfitting of the model. So, adjusted R2 value will not consider the additional independent variables of the data while defining the metric of the model. While the former one will define the variability of the dependent variable, MSE will determine the goodness of fit. It will result in the value explaining how much data of all the data points are deviated from the actual data. If the data is too big, the number might be too big to extract some insights, so RMSE i.e., Root Mean Squared Error. It is just the square root of the MSE. It is just for making the interpretation of results easy. The last one is MAE, which will not sum the errors from the data, but it will take the absolute of all of the errors.  
  
Comparatively, R2 value will give the dependent variability and the others will be give me the comparison from all other models.

**PART VI: Summary**

Based on all the results, I would be wondering to try out still the transformations or normalizations of data. I could increase the data split for testing, and I really think the data itself is in such a form. Out the models I built, the last one model with transformed size and price is the best one which has comparatively lowest error of prediction by explaining 25% for the dependent variable.