Proceedings of the 2nd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/19/20230562

Prediction of intel CPUs' price with regression analysis

Dongrong Joe Fu

Eleanor Roosevelt College, Department of Math, University of California San Diego, La Jolla, 92093, San Diego, California, United States

dofu@ucsd.edu

Abstract. CPU stands for the central processing unit. It is a unit that executes the instruction and allows the computer to run programs, making it an essential computer component. When people are looking for a new computer, either buying a built one or building one, CPU takes up a significant portion of the computer budget. The existence of Moore's Law indicates that the CPUs' price is predictable. This essay constructed the two models, multinomial linear regression and multivariable polynomial regression models, based on parts of Intel CPUs' parameter value to predict their recommended sale price. According to the data set used in this research and the built model, the multinomial linear regression model is more accurate in predicting.

Keywords: python, regression analysis, prediction, intel CPU.

1. Introduction

CPU stands for the central processing unit. As it provides the instruction and processing power for a computer to work, it is an essential component of a computer. The more powerful the processor, the shorter the time the computer needs to finish a job. Everyone wants to get the best CPU for their computer, but the budget is a concern. Intel is currently one of the largest CPU manufacturers. This research was conducted to help Intel fans who want to buy a new computer estimate the budget, either building their own or buying a prebuilt one. This paper will discuss how to build a multinomial linear regression model and multivariable polynomial regression model based on numerous parameters of the CPUs and compare which model is more suitable for predicting recommended CPU sale prices.

2. Literature Review

Moore's Law is a prediction that Gordon Moore, the co-founder of Intel, made in a magazine article, "Cramming more components onto integrated circuits." He predicted that the number of transistors on a microchip would double approximately every two years, leading to exponential increases in computing power and decreases in the cost of electronic components, such as CPUs. Due to the rapid development and evolution of the computing industry, this prediction has held true for more than 50 years [1]. As a stated in Moore's Law, computer components, such as CPUs and GPUs, have become faster, smaller, and more powerful over the decades. This change has made our life and works much more convenient. Although some argue that Moore's Law would eventually end, it has continued to

DOI: 10.54254/2753-8818/19/20230562

hold true due to ongoing technological and manufacturing innovation. Because of Moore's Law indicates that CPU prices are predictable.

Regression analysis is commonly used for predicting housing prices based on different features of houses, such as living area, material or the roof, etc. [2]. CPUs also have different parameters that influence their price, so regression analysis should also be applied to predict CPUs' prices.

3. Methodology

This research will focus on two regression models. The first was the multinomial linear regression model, and the second was the polynomial regression model.

3.1. Multinomial Linear Regression

To model a linear relationship between a random response, Y, and explanatory independent variables, xs, the following formula can be used:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

to find the line of best fit, where each x is known; they are the parameters of each CPU. βs are unknown; they are the coefficient needed to be found ε, epsilon, is a random variable of error. It is the shortest distance between the predicted Y and the actual Y. Since it is assumed that the expected epsilon is zero, which means all the predicted Y and actual Y are matched, the expected Y would be

$$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

To find the best values for βs, a line with the smallest error need to be found. In order words, βs that minimize the sum of the absolute value of each epsilon is the best coefficient for the model.

3.2. Multivariable Polynomial Regression

The way to build a multivariable polynomial regression is about the same as to build a multinomial linear regression model. However, the line of best-fit function has more known variables. The exponent of each explanatory independent variable, xs, is also considered an explanatory variable. In addition, interaction terms, such as $x_k x_{k+1}$, are included in the formula. By adding interaction terms, the effect fluctuation of explanatory independent variables on Y caused by the difference of other independent variable values would be taken into account. For a Y with two variables, the formula of the line of best fit for its multinomial linear regression model would be $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$. For multivariable polynomial regression, the formula of the line of best fit would be:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \varepsilon$$

Similar to multinomial linear regression, the best βs is the ones that minimize the sum of the absolute value of each epsilon.

4. Data Acquisition and Processing

The data set is from Kaggle, a platform that shares free resources for data scientists and machine learning practitioners. Prediction of CPU price seems to be a non-popular topic; there are few relevant data sets. The data set, Computer parts(CPUs and GPUs), was the most relevant one that can be possibly found on Kaggle. The data set is twenty-two hundred and eighty-three by forty-five, which are a lot, but there are a high proportion of missing value and too many parameters to use.

According to some online research done, this paper would keep seventeen parameters and build regression models based on them. They are the key parameters that determine the power of CPUs. Also, using all parameters might cause over-fitting, which means that the model is exactly for the training set and not applicable to other unseen data sets. Still, this can be prevented by limiting it to seventeen variables. The data type in the data set is initially an object. They were changed to float for calculation purposes and added units to numerical data columns. Also, the CPU industry had tremendous technological innovation during the previous two decades. Old CPUs might make the

DOI: 10.54254/2753-8818/19/20230562

prediction less accurate due to the performance difference. Therefore, all the end-of-life and end-of-interactive support CPUs in the status columns were removed. *Figure 1* shows a summary of the filtered data set. A brief definition of each chosen CPU' parameter can be found in Appendix A.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1058 entries, 0 to 1057
Data columns (total 17 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Product_Collection                    1058 non-null   object
 1   Vertical_Segment                      1058 non-null   object
 2   Processor_Number                      1047 non-null   object
 3   Status                                1058 non-null   object
 4   Launch_Year                           1044 non-null   float64
 5   Lithography(nm)                       1044 non-null   float64
 6   Recommended_Customer_Price(USD)       879 non-null    float64
 7   nb_of_Cores                           1058 non-null   float64
 8   nb_of_Threads                         1018 non-null   float64
 9   Processor_Base_Frequency(GHz)         1045 non-null   float64
 10  Cache(MB)                             1056 non-null   float64
 11  TDP(W)                                1021 non-null   float64
 12  Max_Memory_Bandwidth(GB/s)            778 non-null    float64
 13  Graphics_Base_Frequency(MHz)          599 non-null    float64
 14  Intel_Hyper_Threading_Technology_     1006 non-null   object
 15  Intel_Virtualization_Technology_VTx_  1044 non-null   object
 16  Instruction_Set(bit)                  980 non-null    float64
dtypes: float64(11), object(6)
memory usage: 140.6+ KB
```

Figure 1. Data set summary after filtering.

The data set is incomplete. There is a proportion of NaN values in each column, which need to be filled before modeling. Columns 4 to 16 are the ones that need to be filled as they are the number that needs to be fit into the model. There are four common ways to fill in the missing value: fill in all with 0s, fill in all with the average of the column, fill in by KNN, and fill in by random forest. Generally, people choose random forest because it yields models with better performance and more accurate prediction, so this paper is going to use it.

4.1. Random Forest for Missing Numerical Values

When a decisions tree was used for filling in the missing values, each decision tree will randomly choose a non-empty value from the column to fill in the missing values, and it will predict the possible values that can be filled in the next entry and so on until it filling all the values. However, this might cause overfitting. While random forest builds multiple decision trees at the same time. It allows each of them to bootstrap from the original data set and replace the data, which prevent them from making errors and wrong prediction [3]. By doing this, each decision tree in the random forest will use part of the data instead of all, preventing overfitting. After that, the algorithm will predict the most suitable value for each missing entry based on the data and decisions made by each decision tree in the random forest. Random forest filling can be only used for numerical data, but the 14^{th} and 15^{th} columns are Boolean. They label whether the CPU has those two technologies or not. For the best result, the column was filled from the least number of missing values to the most. So the order is the number of cores, cache, processor base frequency, launch year, lithography, thermal design power, number of threads, instruction set, recommended sale price, and, last, max memory bandwidth.

4.2. KNN Imputation for Missing Values

After filling in all the numerical data, Boolean data, hyperthreading technology and virtualization technology, are next. This time, instead of random forest, KNN was used for a classification task. KNN means k-nearest neighbors algorithm. It classifies a data point by choosing the most frequent category between its k nearest neighbors. If k is small, the prediction is likely to be sensitive to noise and overfitting the training data; if k is large, the prediction is expected to be more biased and less accurate. To choose the most suitable k, the error rate of the classifier with random testing sets was calculated with k from 1 to 10. Since the algorithm does not output the same result due to randomness, the k with the highest accuracy in 5 cross-validations was chosen. In this paper, the model was trained to learn how to classify whether a CPU has these two technologies, then create labels for them, where

DOI: 10.54254/2753-8818/19/20230562

zero means the CPU has no corresponding technology, and one means it does. If the corresponding entry is null, it would be filled with the label. Since the virtualization technology columns have less proportion of NaN values, it was filled first. The result shows that the best k for the Intel virtualization technology column is 9 and has 95% accuracy; and the best k for the Intel hyperthreading technology column is 7 and has 79% accuracy.

4.3. Outliers Removal

After filling in all the empty values, a scatter plot was plotted between the recommended sale price and each CPU parameter to spot extreme data points.

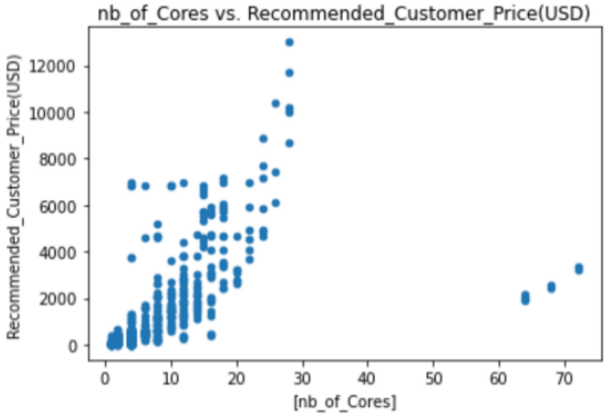

Figure 2. Scatter plot that indicates outliers in number of cores column.

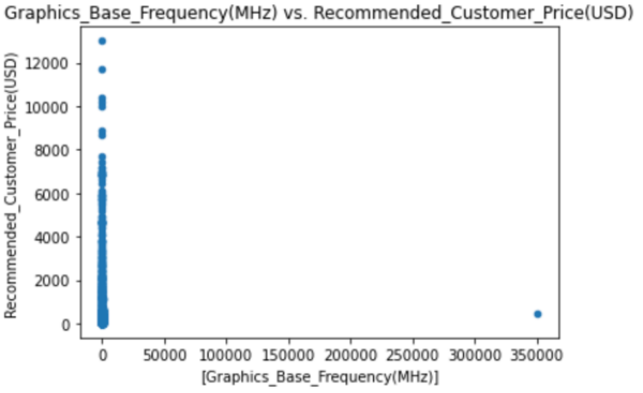

Figure 3. Scatter plot that indicates outliers in graphics base frequency column.

Figure 2 shows that there are CPUs with more than 40 cores, and Figure 3 indicates that one CPU has 350 gigahertz base frequency, which to research, such CPU do not exist. These extreme data points indicated that there are outliers in the data set. This could be caused by human input error, like the 350 gigahertz base frequency case, which is a typo in the unit of the original data set. This could also be due to the error made by the random forest and KNN algorithm on the filled-in values. The typical way to remove outliers is to remove any value that is 1.5 IQR higher than the median and 1.5 IQR lower than the median. Statistically, the probability of having extreme values like these is considerably small, meaning those values are likely outliers. Therefore, they need to be removed, and about two hundred and eighty rows were removed.

DOI: 10.54254/2753-8818/19/20230562

5. Data Visualization

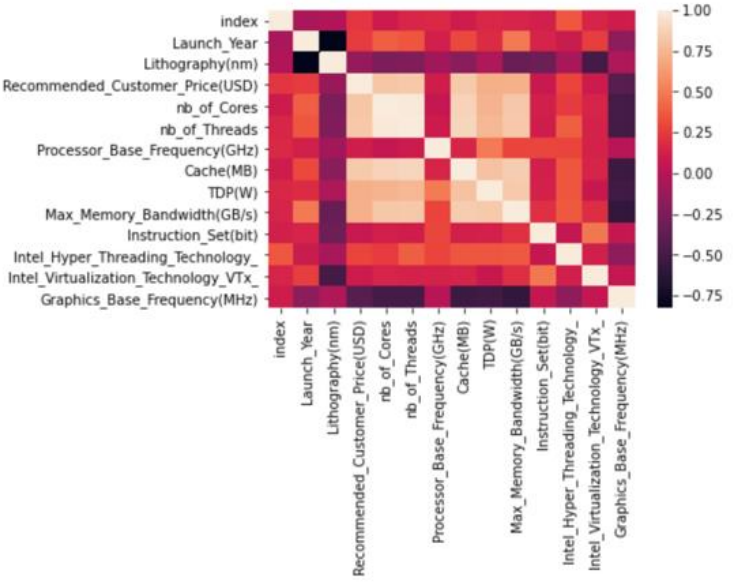

Figure 4. Correlation heat map between every two parameters.

```
index                                0.230893
Launch_Year                          0.255733
Lithography(nm)                     -0.140388
Recommended_Customer_Price(USD)      1.000000
nb_of_Cores                          0.816206
nb_of_Threads                        0.833736
Processor_Base_Frequency(GHz)        0.117450
Cache(MB)                            0.842073
TDP(W)                               0.720611
Max_Memory_Bandwidth(GB/s)           0.722580
Instruction_Set(bit)                 0.076189
Intel_Hyper_Threading_Technology_    0.291808
Intel_Virtualization_Technology_VTx_ 0.095086
Graphics_Base_Frequency(MHz)        -0.402564
Name: Recommended_Customer_Price(USD), dtype: float64
```

Figure 5. Correlation heat map between each parameter and price.

Figure 4 is a correlation heat map plotted by the seaborn module in Python. It displays the correlation between every two variables. The lighter the color, the more positive the correlation. Figure 5 shows the correlation coefficient between the recommended sale price and each numerical data. The correlation is pretty decent, which shows the parameters chosen to include in the model are the parameters that significantly determine CPUs' price. The correlation between the recommended sale price and lithography is negative because for two CPUs with identical performance, the smaller the lithography, the smaller the power consumption, which means it requires more advanced manufacturing technology, so its sale price will be higher. You can also tell there is no correlation between the sale price, instruction set, and visualization technology. After filtering, all the remaining CPUs have a 64-bit instruction set and visualization technology.

6. Model Building

Using the entire data set for modeling might cause overfitting. Therefore, before modeling, the data set needs to be separated into a training set to train the model and into a testing set to evaluate the model. The training set was set to be randomly 80% of the whole data set. In this research, columns 4 to 16

DOI: 10.54254/2753-8818/19/20230562

are xs that are used to predict the Y, the recommended price of CPUs. Epsilons are the differences between predicted and actual prices with different βs.

The multinomial linear regression model was built by the Linear Regression class in sklearn.linear_model module. Based on the fitted training set, the function will output the most suitable β for each x that has the smallest sum of the absolute difference between each predicted and actual Intel CPU price. *Figure 6* shows the line plot with the recommended sale price predicted by the multinomial linear regression model and the actual sale price in the testing set.

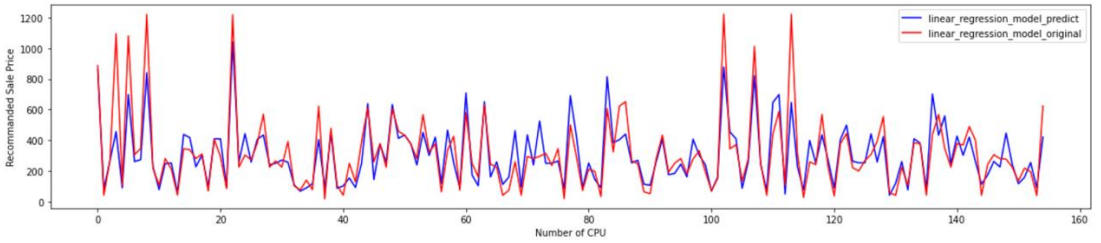

Figure 6. Multinomial linear regression model prediction.

A scatter plots was then plotted to show the relationship between the recommended sale price and columns 4 to 16, which can be found in Appendix B. Some of them seem to have a non-linear relationship. Therefore, a polynomial regression model should be built for model performance comparison. Since the way to find βs for the multivariable polynomial regression line of best fit is about the same as the multinomial linear regression line of best fit, the Linear Regression class in sklearn.linear_model module can be still used, but with more steps. Firstly, Polynomial Features from sklearn.preprocessing was used to set the degree of polynomial feature, that is, the highest exponent term in the model. Several randomly assigned 80% of the whole data set was used as training sets to build the model with polynomial feature degrees 2, 3, and 4. A huge gap between some predicted and actual prices was always found if the degree was set to be above 2. This indicates that overfitting happens if the degree is more than 2. Therefore, the polynomial feature degree was set to 2 for this model. Secondly, poly.transform function was used to create exponent and interaction terms for degree 2 polynomial feature. By adding interacted terms, the effect fluctuation of a CPU parameter on the recommended sale price caused by the changes in other parameters' values can be considered. Lastly, the training set data can be fit into the model, and Linear Regression function will calculate the best βs. *Figure 7* shows the line plot with the recommended sale price predicted by the multivariable polynomial regression model and the actual sale price in the testing set.

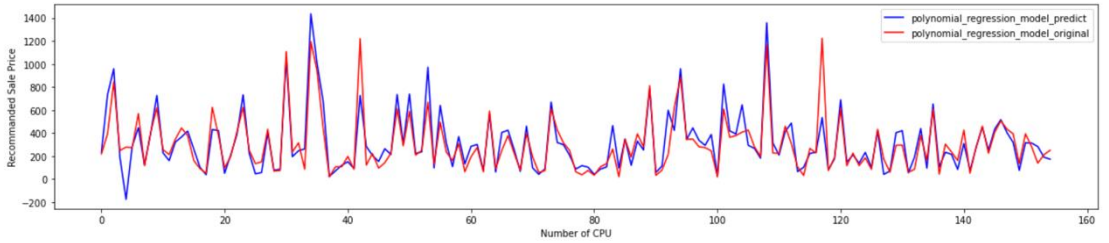

Figure 7. Multivariable polynomial regression model prediction.

7. Result

Comparing models' mean square error, MSE, is a common method to evaluate the performance of models. Its formula is:

DOI: 10.54254/2753-8818/19/20230562

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

Where n is the total number of data points, MSE measures the average squared difference between each predicted data point and the actual data point. The smaller the MSE means, the better the line of best fit is. Since the training and testing sets are randomly assigned each time, the models' performance would be heavily influenced by their assignment. To prevent this error to the greatest extent, another technique called cross-validation would also need to be used.

Also, since the training set and testing set are assigned each time randomly, the performance of each model evaluated by the corresponding testing set fluctuates. Therefore, k-fold cross-validation need to be used. It divided the original data randomly into k training sets without replacement and used $k-1$ part use training set for modeling and 1 part for testing. Repeating these steps k time, each subset of the original data set can be a test set, and the rest can be a training set, lowering the model performance's sensitivity to data partitioning. The average MSE value of k group of test results can then measure model performance.

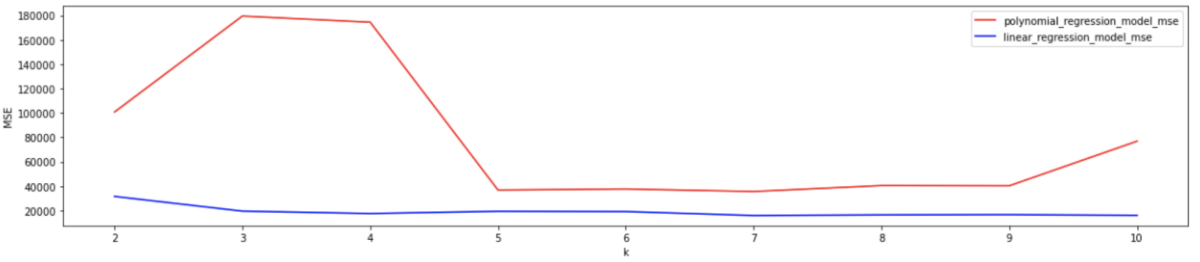

Figure 8. Changes in MSE of two models with difference k value.

Figure 8 shows how MSE of two models varies by changing k value. The MSE of multinomial linear regression is relatively stable, whereas the MSE of multivariable polynomial regression fluctuates considerably. This indicates that the polynomial regression model would make significant errors with particular training and testing sets. After plotting the actual price versus the predicted price made by the polynomial regression model several times, the model was found to make ten times higher predictions than the actual price sometimes. Besides, MSE of multinomial linear regression is always lower. From this, it can be concluded that the multinomial linear regression model is more suitable for this data set. Its MSE for 10-fold cross-validation is 16071.

8. Conclusion

After filtering, all the remaining CPUs have a 64-bit instruction set and visualization technology, which means that the model would be less accurate in predicting the recommended sale price of a CPU without a 64-bit instruction set and visualization technology. However, due to the incomplete data set, this can only be solved by manually collecting data from Intel's official website. Besides, a considerable proportion of missing data were filled by random forest. However, they can't be explained, so it's hard to understand how or why they were filled with the current value they had. This impenetrability means that the model must be trusted as is and the results accepted as is. In addition, the parameters selected for the model are based on research and personal experience. This research cannot prove that the chosen parameters have the most significant impact on CPU prices. Statistically, they might not be the top influential variables to the recommended sale price. To evaluate the importance of each feature, we can use random forest classification to rank each feature in order. This is something the research will focus on next. The next research will compare the top 12 most influential features chosen by random forest classification to see if there's a big difference. After that, the process done in this research would be repeated with additional updated CPUs data from Intel's

DOI: 10.54254/2753-8818/19/20230562

official website and see if the conclusion, that is, multinomial linear regression is more suitable, is still valid.

Reference

[1] R. R. Schaller, "Moore's law: past, present and future," in *IEEE Spectrum*, vol. 34, no. 6, pp. 52-59, June 1997, doi: 10.1109/6.591665.
[2] Yu, H., & Wu, J. (n.d.). (rep.). *Real Estate Price Prediction with Regression and Classification*.
[3] Tang, F, Ishwaran, H. Random Forest Missing Data Algorithms. *Stat Anal Data Min: The ASA Data Sci Journal.* 2017; 10: 363– 377. https://doi.org/10.1002/sam.11348.

Appendix A

Table 1. The definition of each chosen CPU parameter.

Columns Index	Parameter Name	Definition
0	Product_Collection	The full name of Intel CPUs includes their brand, brand modifier, generation indicator, SKU numeric digits, and product line suffixes.
1	Vertical_Segment	Indicates the type of Intel CPUs: mobile, desktop, embedded, or server.
2	Processor_Number	Shorter name of Intel CPUs.
3	Status	Status of Intel CPUs.
4	Launch_Year	Official launch year of Intel CPUs.
5	Lithography(nm)	A measurement of how small the manufacturer can make the transistors.
6	Recommended_Customer_Price(USD)	Official recommended customer price made by Intel.
7	nb_of_Cores	The number of processing units, cores, that are integrated into the processor circuit.
8	nb_of_Threads	The number of execution units, threads, on concurrent programming.
9	Processor_Base_Frequency(GHz)	The number of cycles per second a CPU can execute when the CPU is operating regularly. The higher frequency, the faster CPU.
10	Cache(MB)	Cache stores copies of the data from frequently used main memory locations, which reduces the average cost to access data from the main memory. The CPU cache measures how much frequent-used data that CPU can store in a computer.
11	TDP(W)	Thermal Design Power. It refers to the power consumption under the maximum theoretical load. Generally, more watts means better performance.
12	Max_Menmory_Bandwidth(GB/s)	The maximum rate CPUs can read data from or store data in a semiconductor memory.
13	Graphics_Base_Frequency(MHz)	Maximum rendering graphics speed of the integrated graphics processing unit in the CPU. If the CPU does not have an integrated graphics processing unit, the value would be 0.

DOI: 10.54254/2753-8818/19/20230562

Table 1. (continued).

14	Inter_Hyper_Threading_Technology_	Whether the CPU has the technology that allows more than one thread to run on each core.
15	Intel_Virtualization_Technology_VTx	Whether the CPU can help a computer to run multiple operating systems simultaneously.
16	Instruction_Set(bit)	The instruction set is a set of instructions in a CPU used to calculate and control a computer system. Instruction set with higher bit number allows users to store more RAM.

Appendix B

Figure 9 shows the relationship between the recommended sale price and columns 4 to 16.

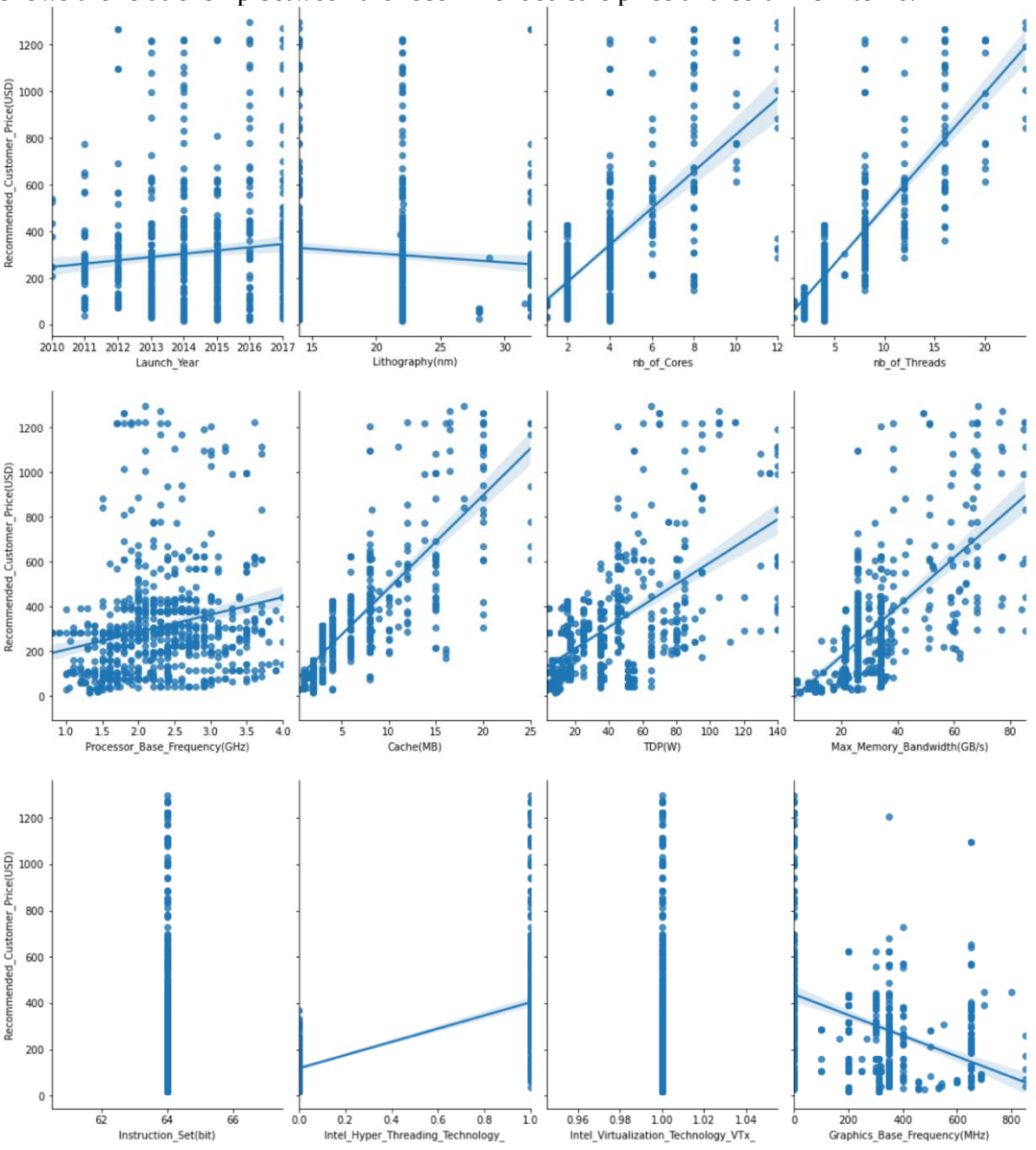

Figure 9. Relationship between the recommended sale price and columns 4 to 16.