**ISM 5/6356 - Data Mining Analytics & Visualization**

**Seattle Pacific University**

**Diamond Price Estimation using Microsoft SQL Server Analysis Services 2015**

**Feb 13th, 2018**

**by Kelly Pham , Suruchi Chaudhary**

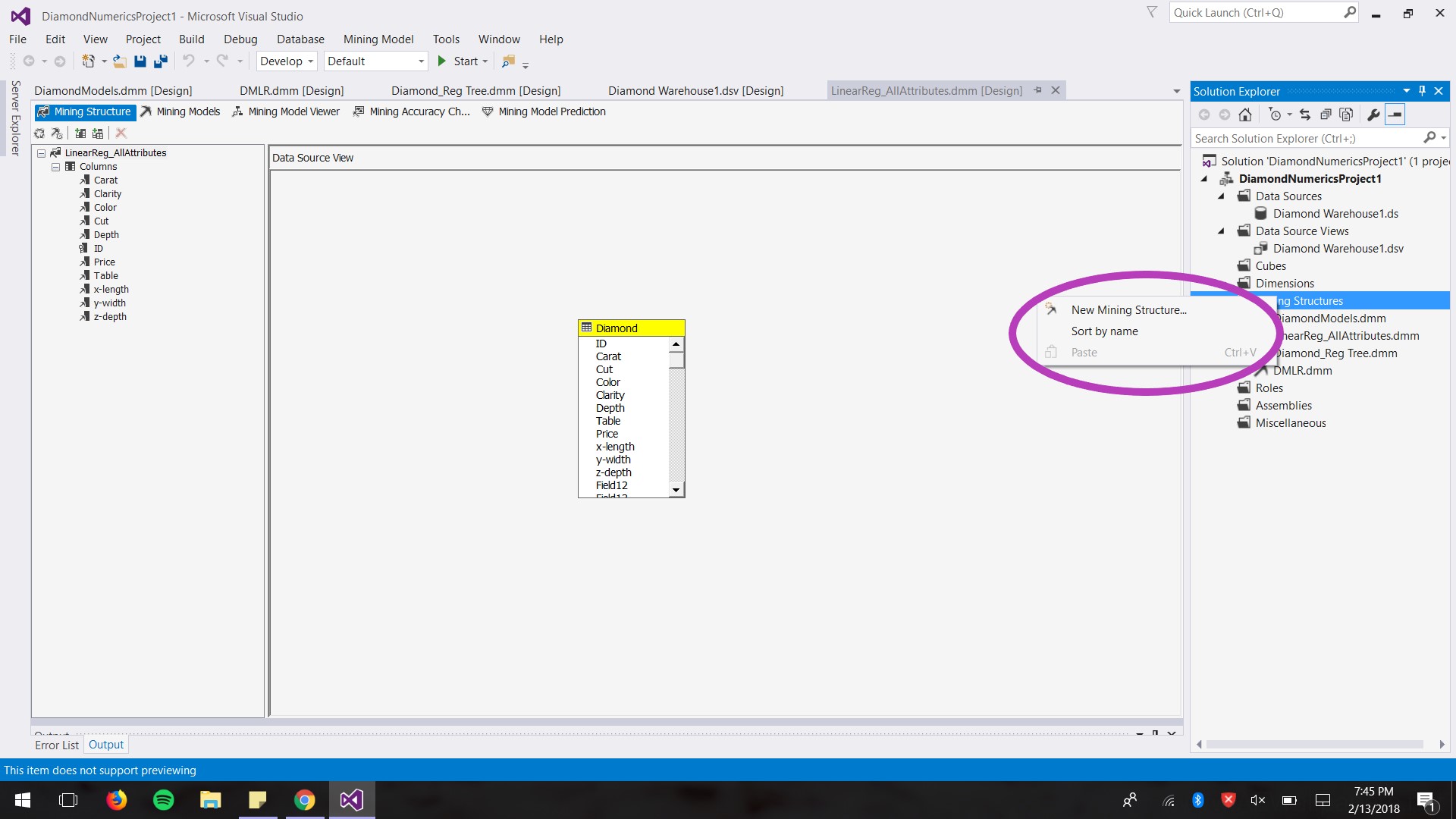
1. **Business Case**

The goal of this data mining project is to generate an algorithm model to predict the price of diamond. This model will predict the price of model based on the most significant attributes, being carat, clarity, color, cut, x-length. With this prediction, a seller can estimate the market price of diamond. On the other hand, this predictive analysis is also helpful for buyers to evaluate the price and quality of the diamond that they are purchasing.

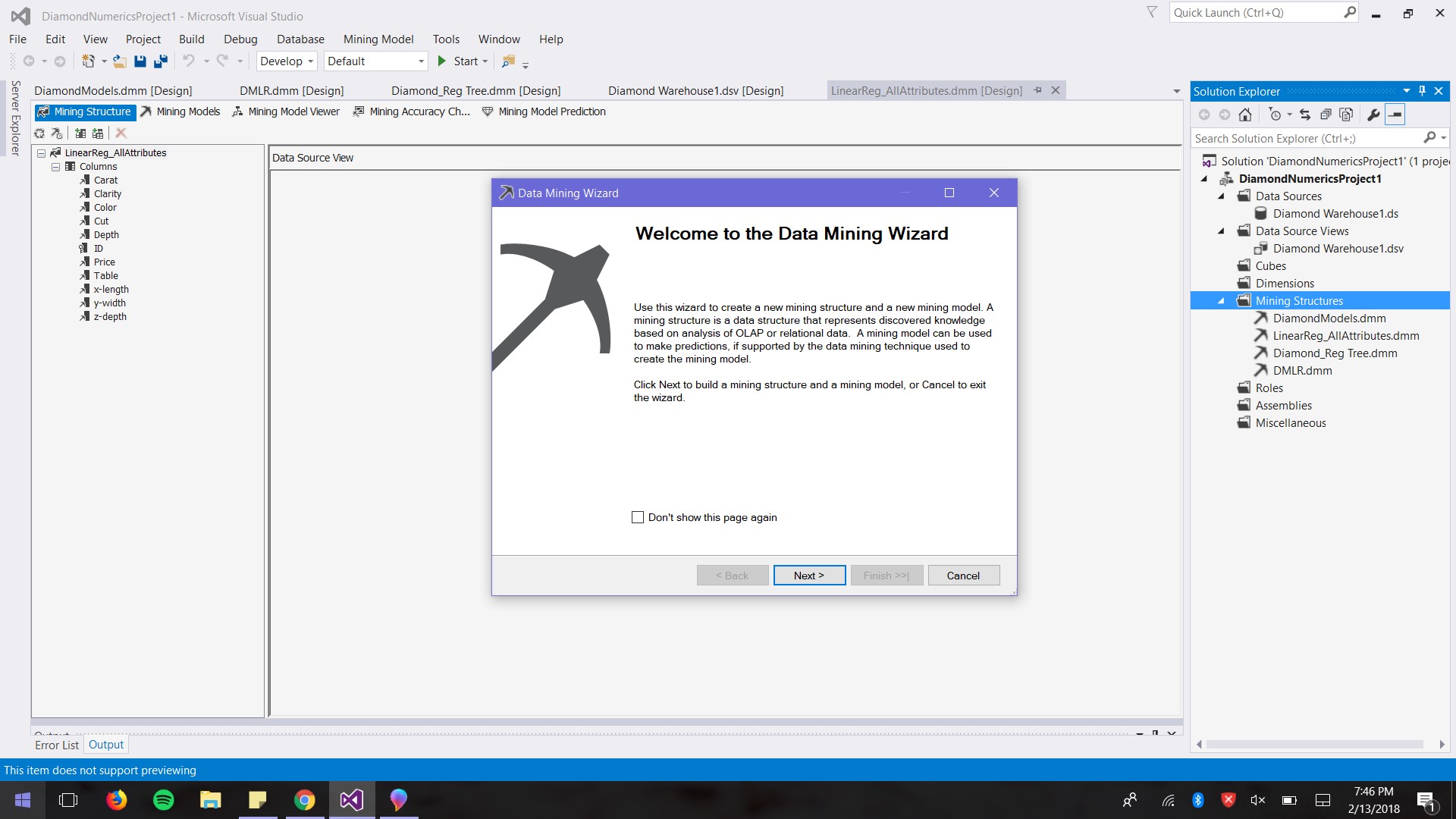
After trying Orange and Microsoft SQL Server, we decided to use the SQL Server. Both of these tools have all functions that we needed for our projects. However, we experienced data importing issue with Orange. The size of our data is almost 54,000 rows and Orange could not load this dataset and crashed. Although Orange has a friendly-user interface, we found it challenging to learn the tool, especially when the learning materials are limited on the Internet. For these reasons, we decided to go with Microsoft SQL Server.

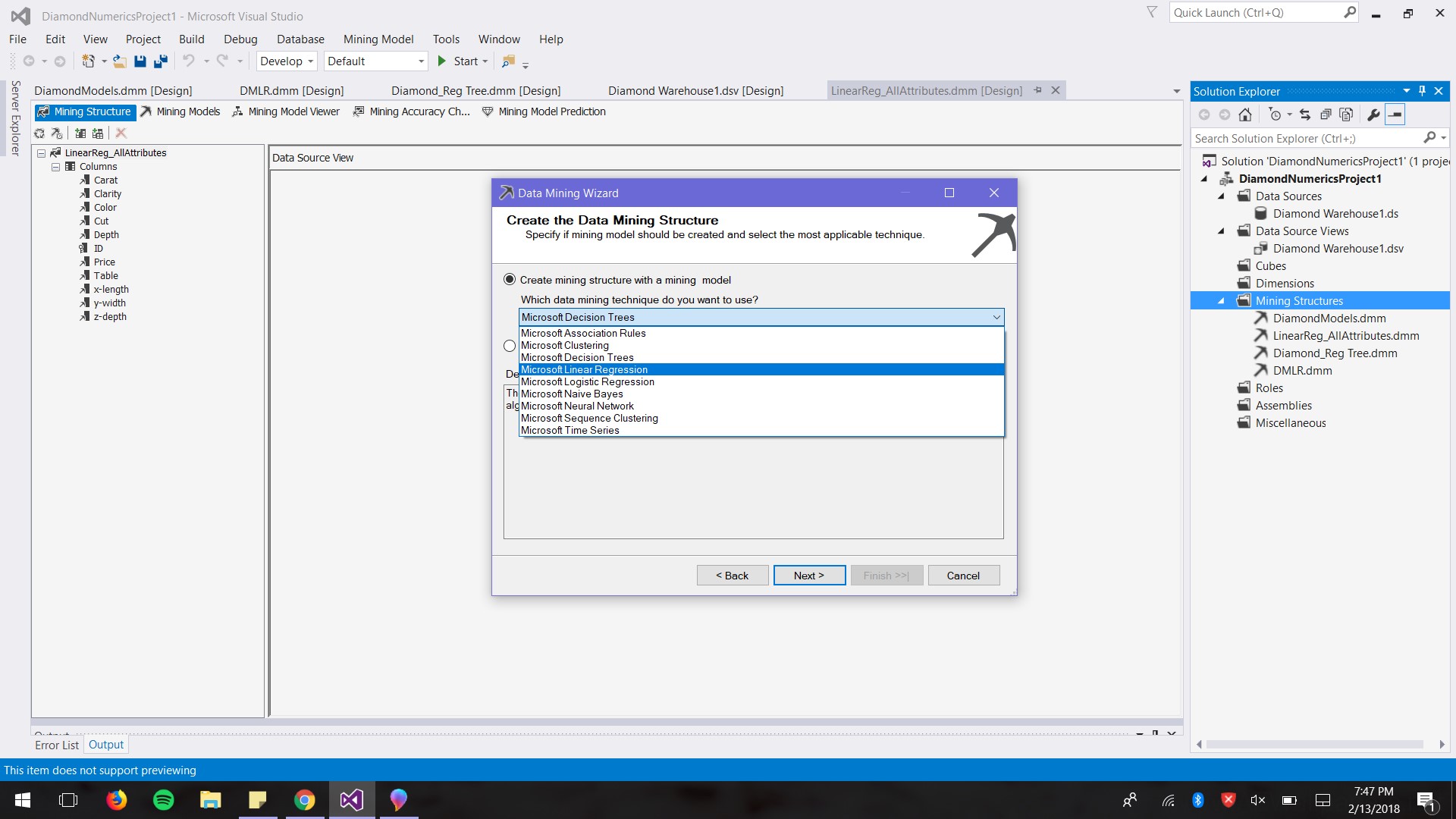
**II. Document how you used the tool.  Many tools are super rich in features and you probably won’t be exploring them all, explain the parts you did use.**

We imported our dataset into SQL Server using the SQL Server Import Wizard. We created the data source and the data source view. The following screenshots will show how we create a mining model in SASS

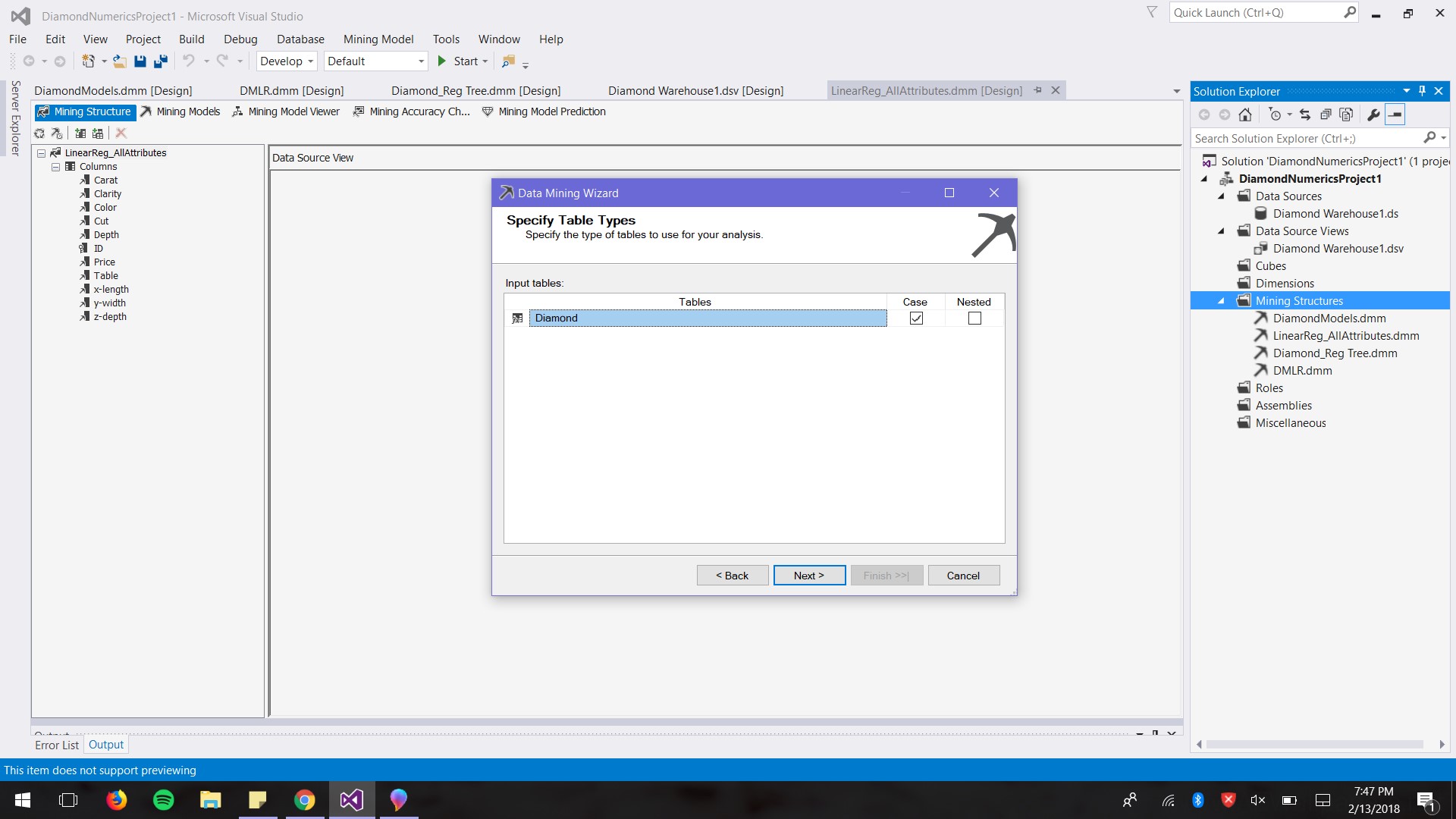


Step 1: Right click on the Mining Structure, and choose New Mining Structure

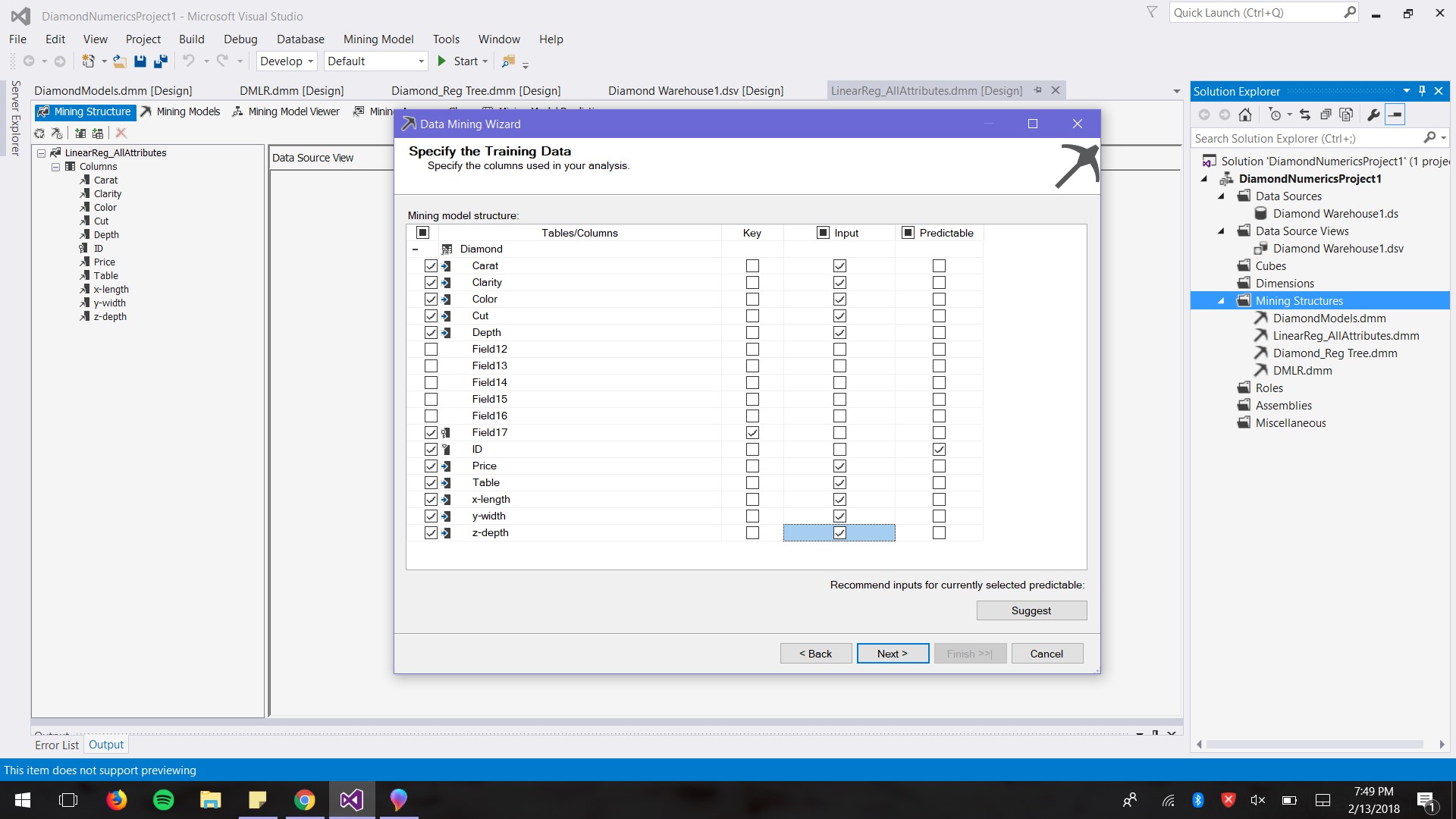




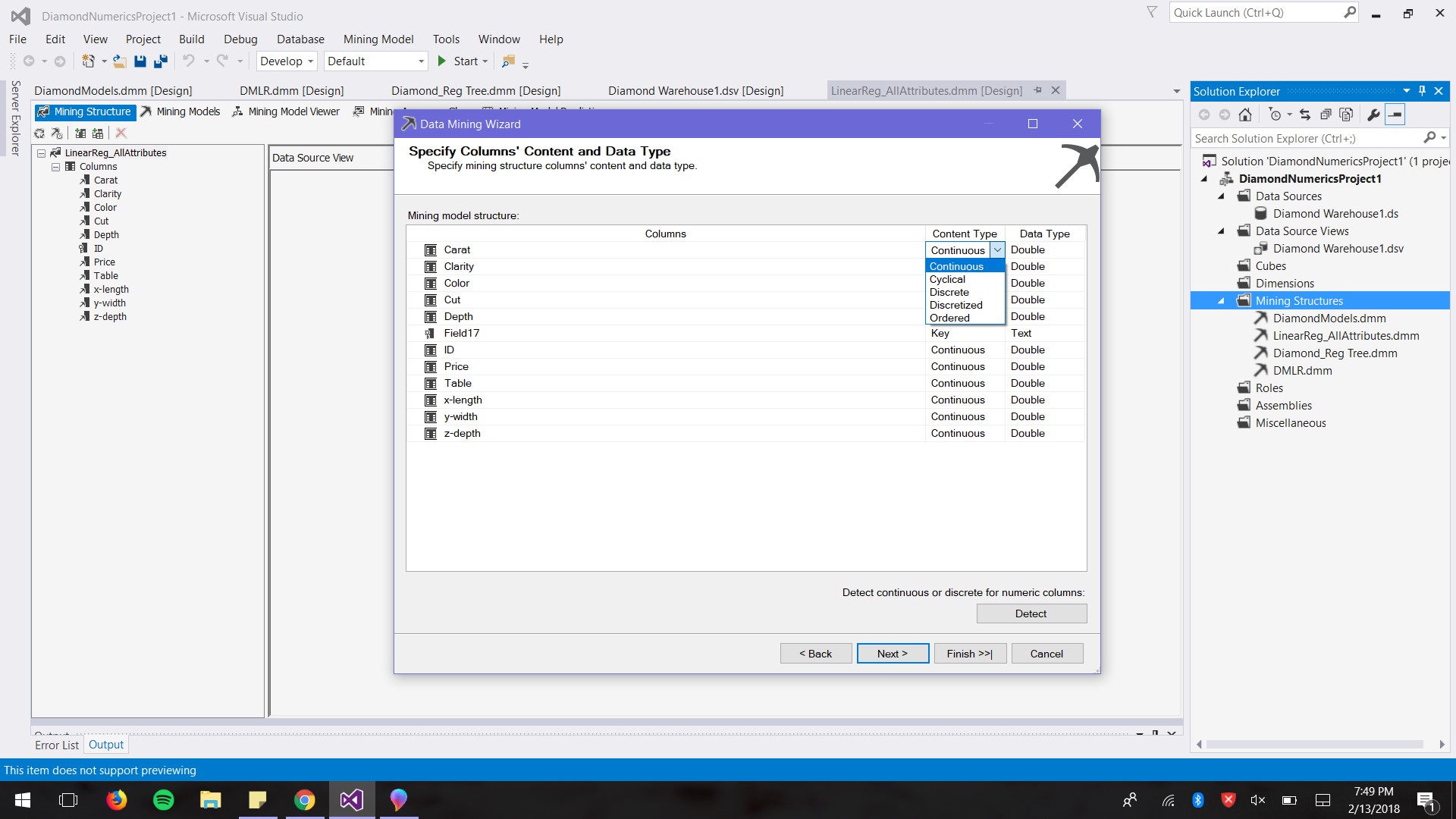
Step 2: We can choose different mining models from the drop bar.



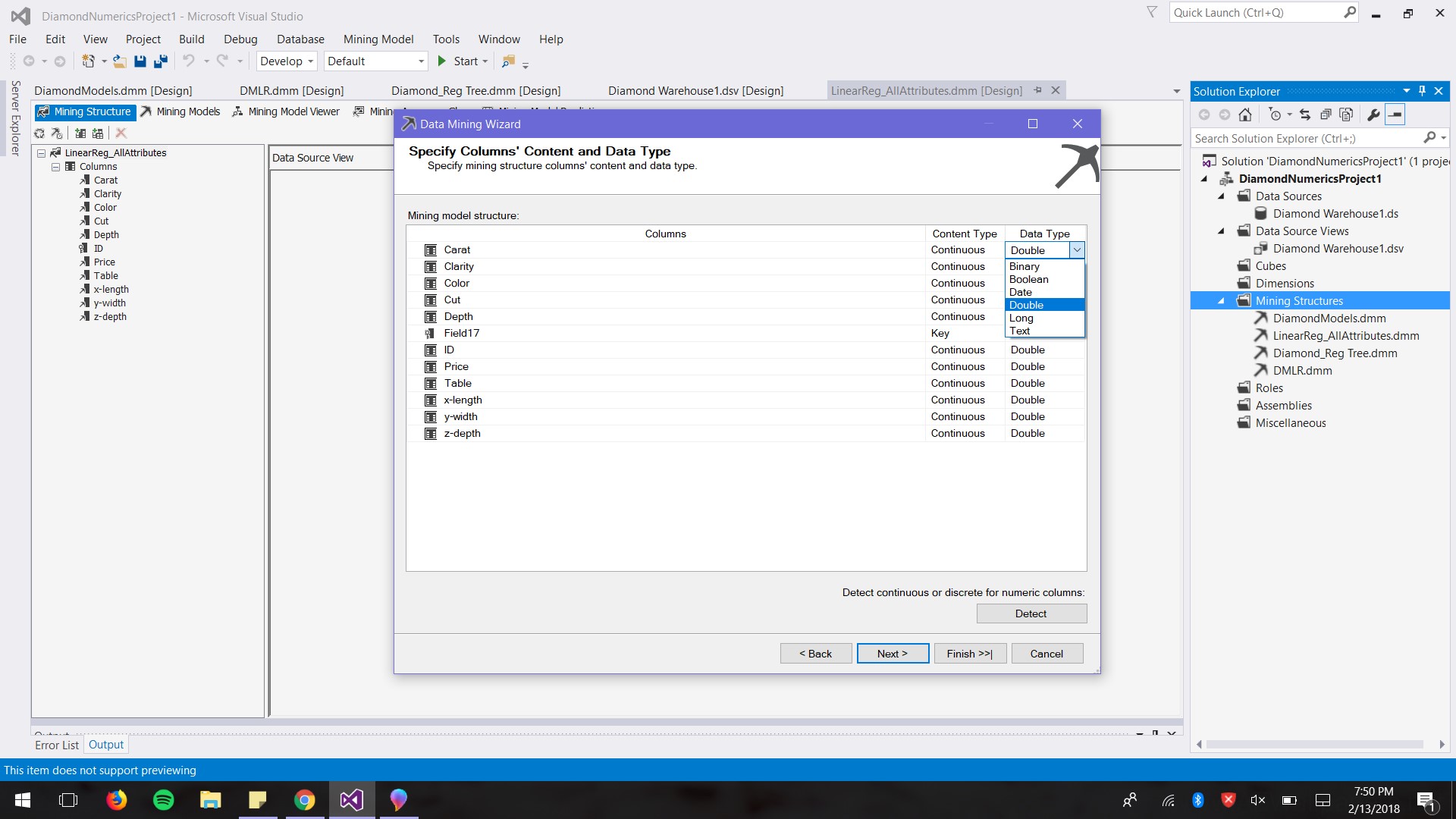
Step 3: Since we only have one table, the Diamond table will be our Case table



Step 4: We choose which attributes to be the inputs, ID key and predictable attribute. If we click Suggest at the bottom right, MSS will provide a list of attributes that are more significant relevant to the predictable value.



Step 5: If we click the drop bar, we can change the Content Type of the data



Step 6: We can also change the data type with the drop bar. Once we hit Finish, we have a mining model.

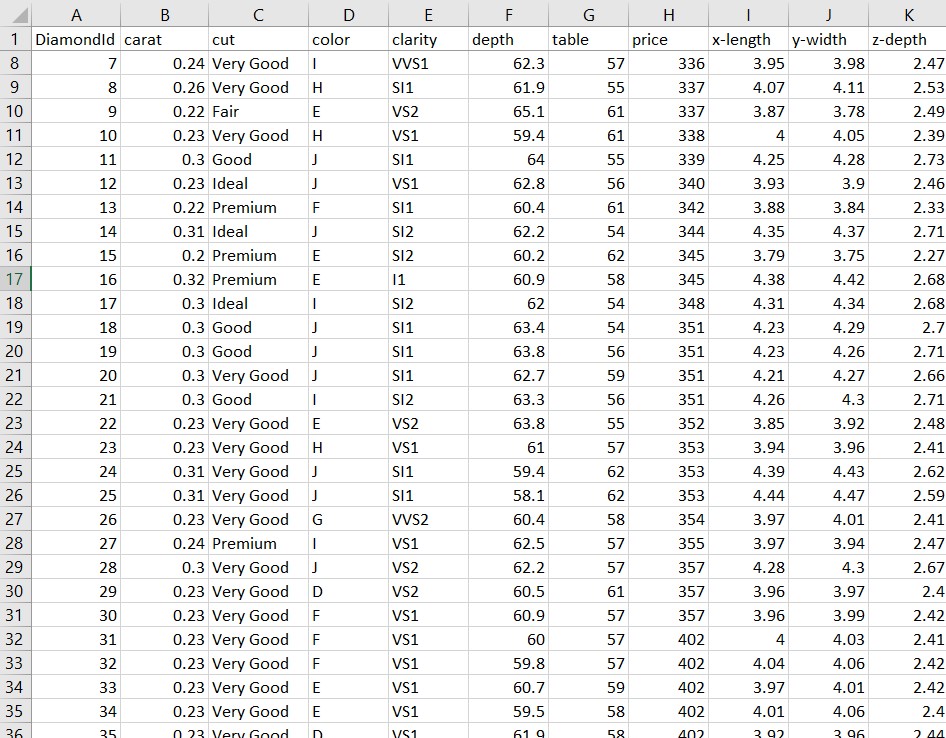
We decided to go with Linear Regression since it was the algorithm that met the requirements of our dataset (only numerical data) and our prediction value (numeric). We wanted to show the correlation between the significant attributes in our dataset and its effect on the price of the diamond. We also wanted to understand which attributes are the most important and if there is one single attribute that dominates the decision making when it comes to diamond pricing.

**III. Data source and Data preparation**

We found the dataset of diamond on Kaggle at

<https://www.kaggle.com/shivam2503/diamonds/version/1>

The size of the data is appropriate for the scope of the project which was 54000 rows. We also agreed that the data is relatively clean for us to start our first data mining project. We examined three datasets and chose this diamond dataset because of its simplicity. This allowed us to spend more time to learn how to use Microsoft SQL Server, and learn how to interpret the results in this tool. Below is a screenshot of the original data.



We had to prepare the data by changing the following attributes to numerics.

**Cut**

|  |  |
| --- | --- |
| Fair (worst) | 1 |
| Good | 2 |
| Very Good | 3 |
| Premium | 4 |
| Ideal (best) | 5 |

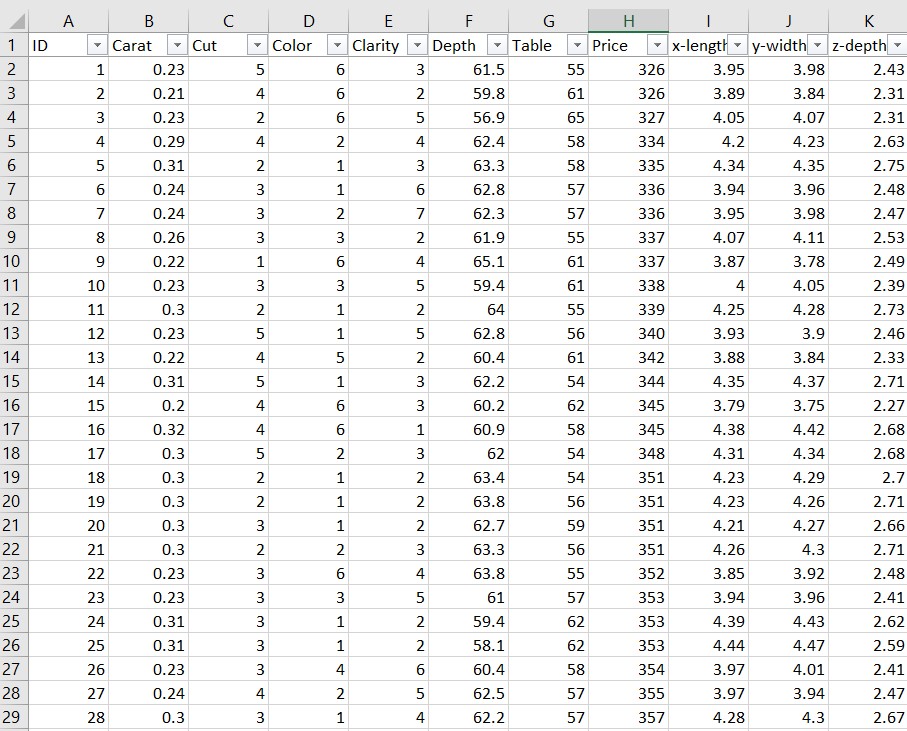
**Color**

|  |  |
| --- | --- |
| J(worst) | 1 |
| I | 2 |
| H | 3 |
| G | 4 |
| F | 5 |
| E | 6 |
| D(best) | 7 |

**Clarity**

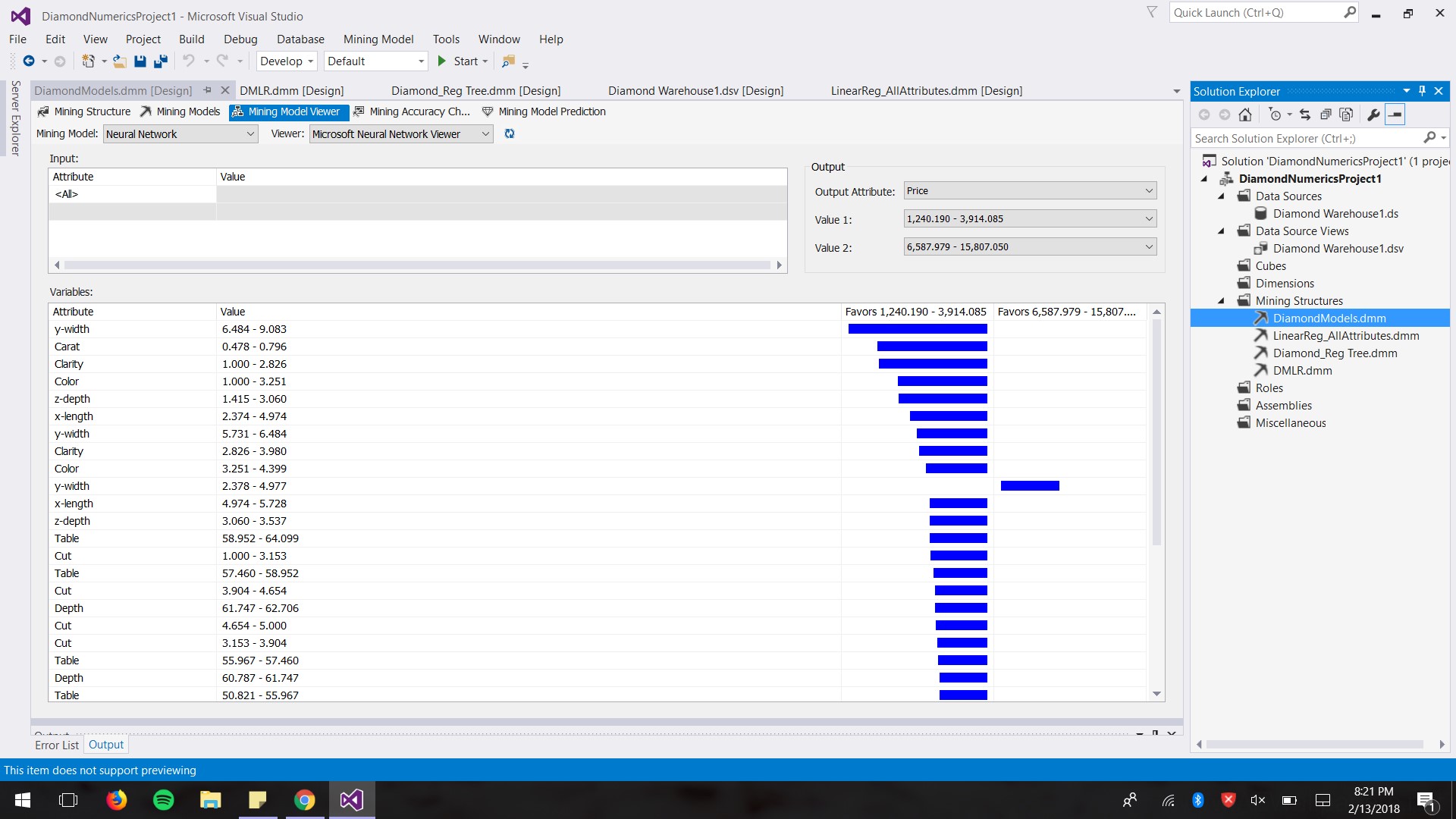
|  |  |
| --- | --- |
| 1I (worst) | 1 |
| SI2 | 2 |
| SI1 | 3 |
| VS2 | 4 |
| VS1 | 5 |
| VVS2 | 6 |
| VVS1 | 7 |
| IF (best) | 8 |

This is how the data looks after we cleaned it.



**IV. When you choose a data mining algorithm(s) for you mining model, tell us why you chose that one (or that category of algorithms)**

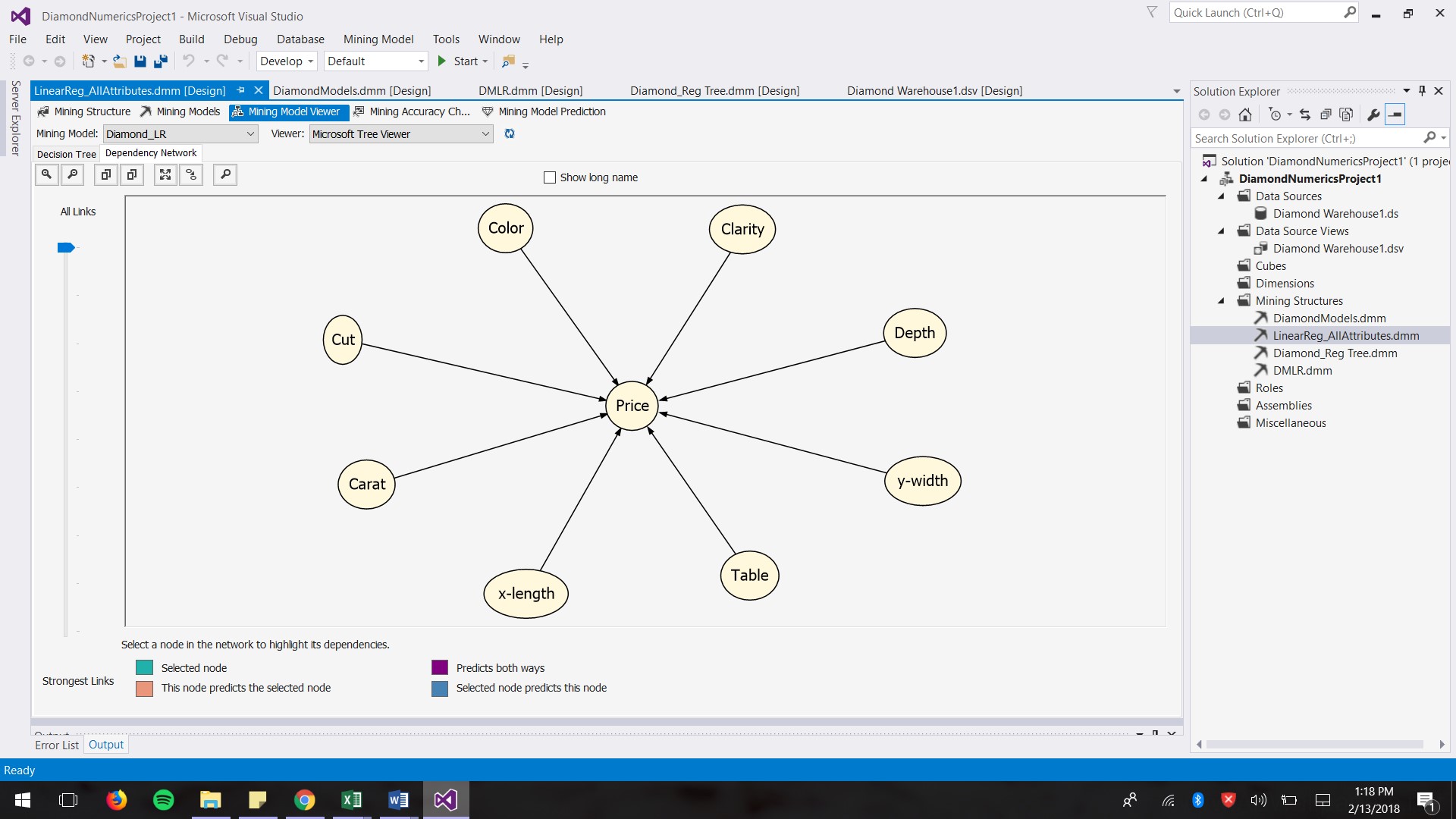
We tried regression tree and neural network. With the result of the regression tree, the errors of the predicted prices were very big compared to the actual prices. The neural network model in Microsoft SQL Server did not provide any helpful insight with the prediction of diamond price.



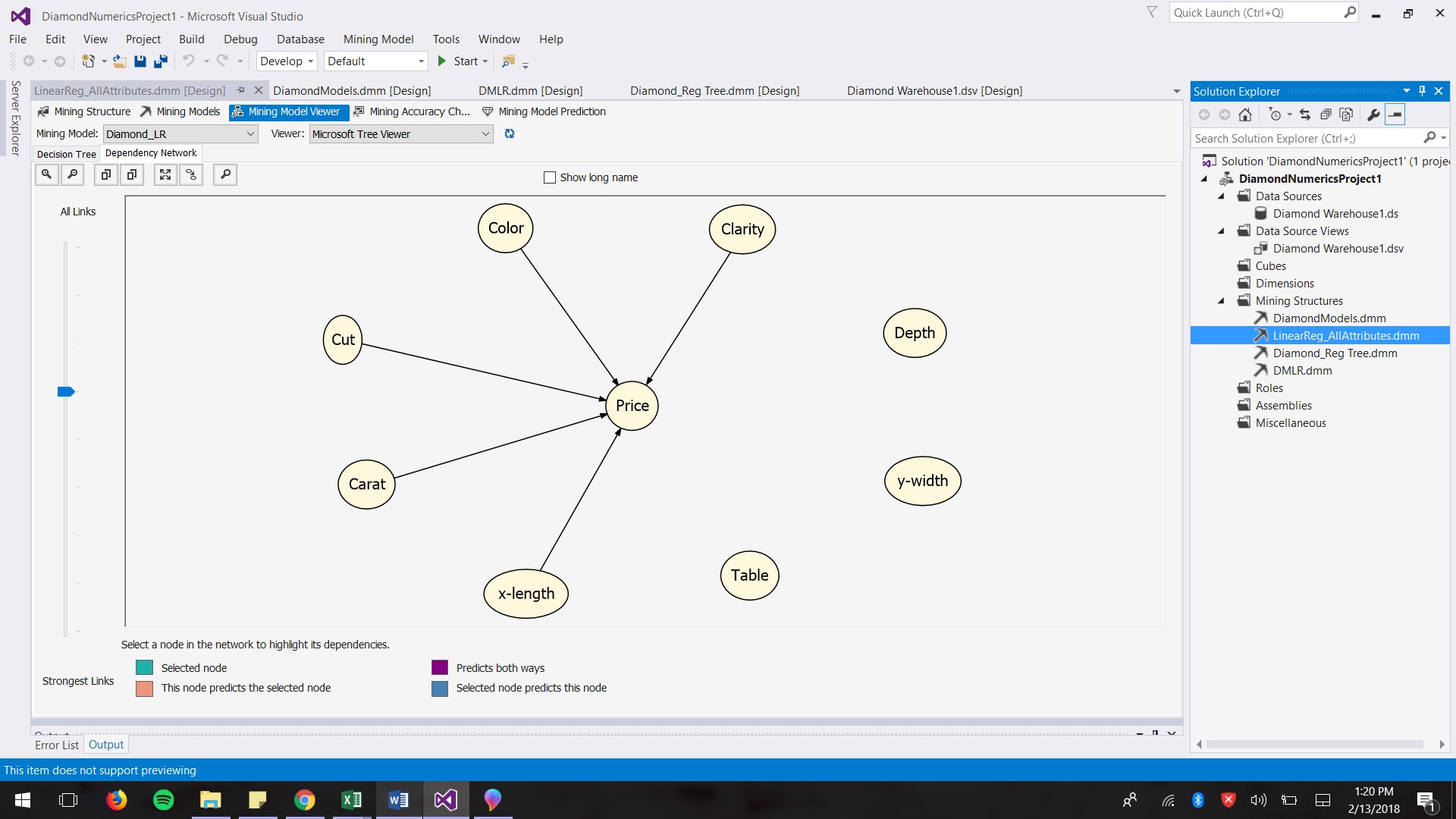
This screenshot is the neural network (NN). Based on what we learned from the tutorials on Youtube, the NN model is helpful to determine which attributes are significant to predictable value. However, for this diamond dataset, there were too many ranges of values for each individual attribute. The price ranges are also broad. Therefore, we did not see any significant correlation between the inputs and the different ranges of price.

The third algorithm we ran was linear regression because regression is a data mining function that predicts a number. Regression task begins with a data set in which the target values are known. A regression algorithm estimates the value of the target attribute as a function of the input attributes for each case. These relationships between target and input attributes are summarized in a model, which can then be applied to a different data set in which the predicted values not known. Since we wanted to predict the diamond price and our attributes are numerical, linear regression was the best model for our data.

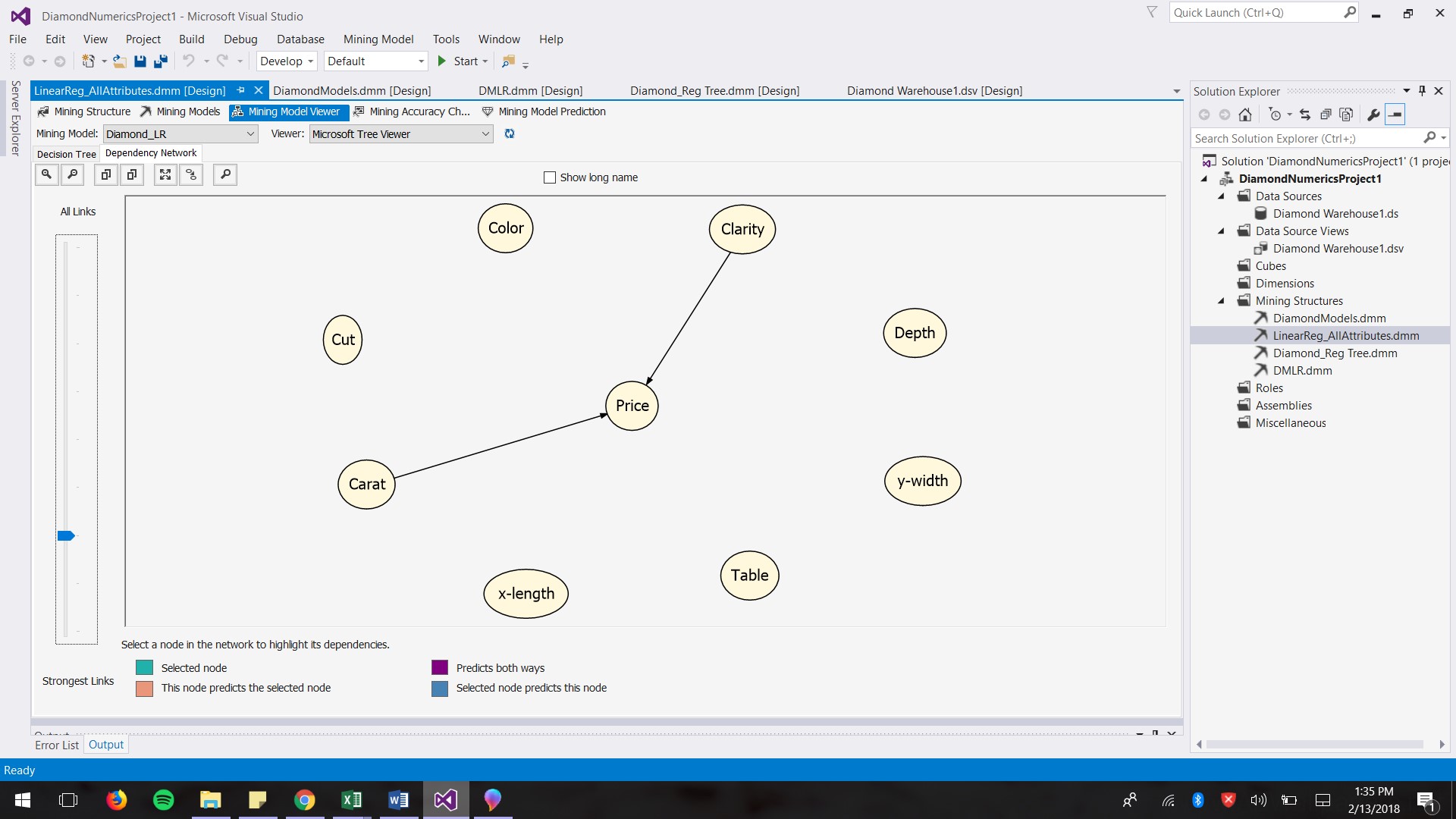
We used the Dependency Network function to determine the correlation between the continuous attributes and the predictive attributes.

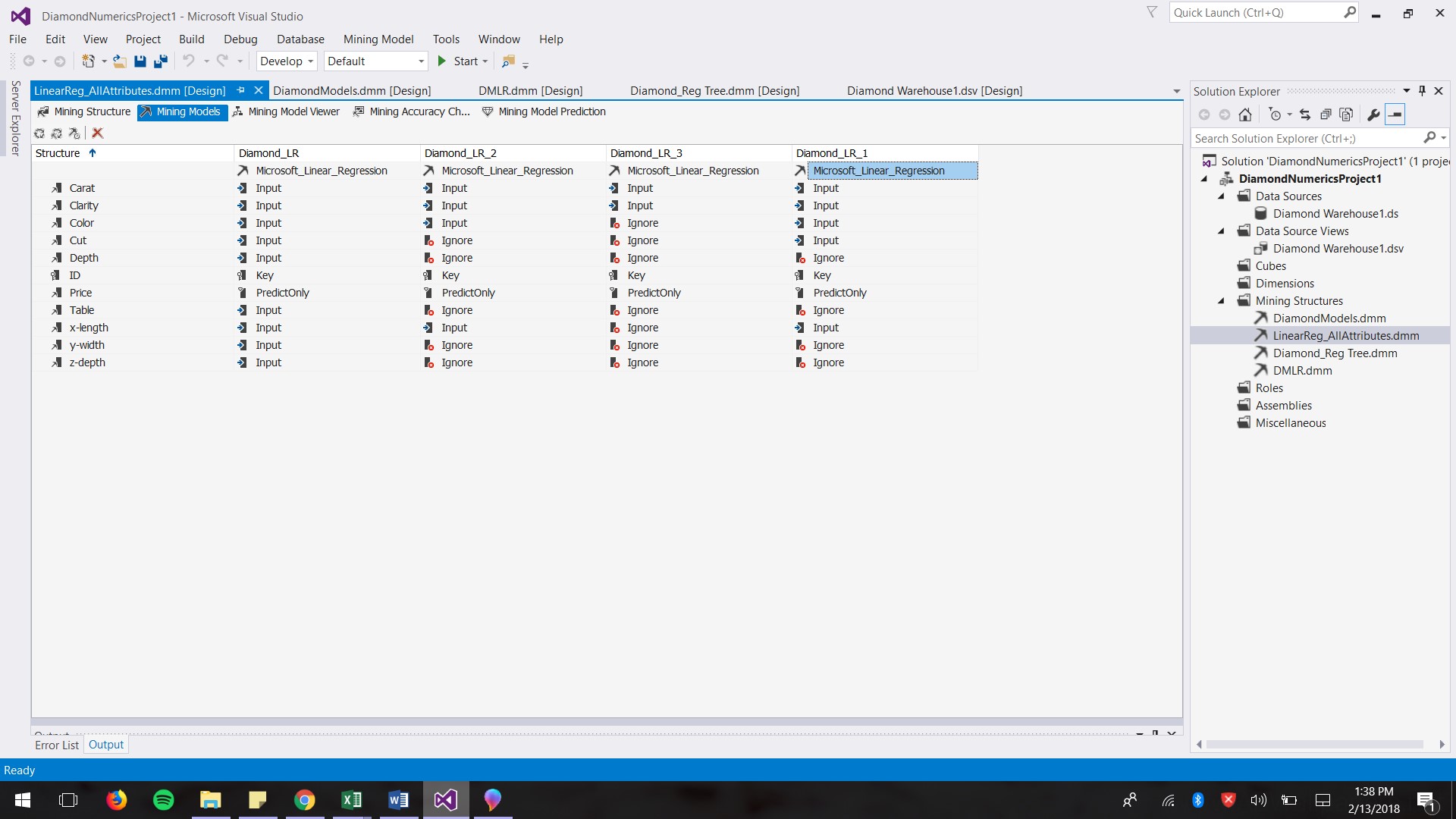
 **Picture A**

As we can see in Picture A, Microsoft SQL Server automatically removed attribute Z-depth from the equation, which means Z is not significantly relevant to the price. When we dragged down the scale on the left side, it will show us which attribute has the stronger link or is more significantly relevant to the predictive attribute, price. (Picture B).

  **Picture B**

For the first linear regression model, we include all the attributes as our inputs for the model. Microsoft SQL Server produced a formulas, not including the Z-depth. With the information from Dependency Network, we ran three more regression models. Each time we eliminated one attribute that is less significant than others. The 4th model only included Carat and Clarity, since these two had the strongest links to price. See Picture C and picture D for more references.

  **Picture C**

** Picture D**

**Diamond\_LR**

Attributes Carat, Clarity, Color, Cut, Depth, Table, x-length

Price = 3972.08 + 10673.837\*(Carat) + 396.976\*(Clarity) + 301.695\*(Color) + 134.608\*(Cut) – 84.747\*(Depth) – 27.009\*(Table) -1104.033(x) + 195.129(y)

**Diamond\_ LR \_1**

Attributes Carat, Clarity, Color, Cut,x-length

Price = 10440.272\*(Carat) + 404.419\*(Clarity) + 304.223\*(Color) + 179.191\*(Cut) – 812.536\*(X) – 3393.263

**Diamond\_LR\_2**

Attributes: Carat, Clarity, Color, x-length

Price = 10309.547\*(Carat) + 422.715\*(Clarity) + 303.575\*(Color) – 772.747\*(x) – 2888.908

**Diamond\_LR\_3**

Attributes: Carat, Clarity

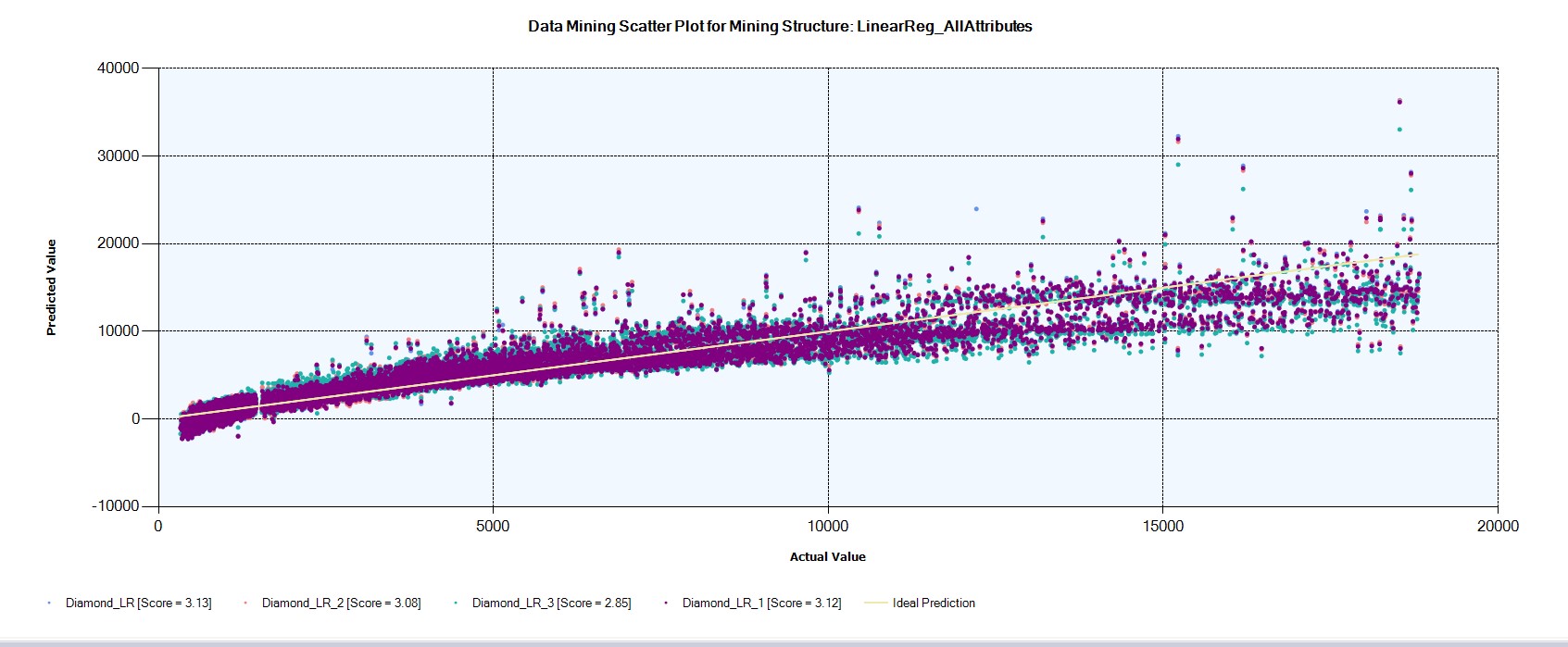
Price = 8189.855\*(Carat) + 401.454\*(Clarity) – 4203.851

We used three different methods to verify the results. First, we used the Mining Model Prediction function in Visual Studio to predict prices for 10 different diamonds. Then we compared these predicted prices with the actual prices in order to see the percentage difference between these two values in Excel.



By doing this, we were able to have a quick examination on which model has the highest frequency of low percentage difference. The result shows that Diamon\_LR\_3 has the most of low percentage difference. Model Diamond\_LR\_1 comes in the second place.

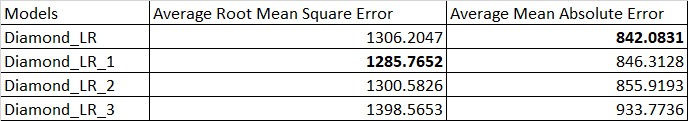
Secondly, we used the Lift Chart function to see the visual fit of these models. This function produced a scatter plot graph with the y-axis representing Predicted Value and x-axis representing Actual Value. It also gave us a score for each model. The higher the score is the better fit the model will be. Based on this Lift Chart, Diamond\_LR and Diamond\_LR\_2 have the highest scores, and therefore, are better fit to the testing dataset.



Thirdly, we used Cross Validation to verify the accuracy rate of these models. The parameters are presented below:

* Training data set = 70%
* Testing data set = 30%
* Max cases = 40,0000
* Fold Count = 10
* Target Attribute = Price

The results include values of Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Log Score for 10 different partitions. We decided to look at the average values of RMSE and MAE to compare the accuracy between 4 linear regression models. The results are shown in the table below.



RMSE is the square root of the mean error for all partition cases, divided by the number of cases in the partition, excluding rows that have missing values for the target attribute. MAE is the average error when predicted values are compared to actual values, calculated as the mean of the absolute sum of errors. A smaller RMSE (or MAE) means predictions were more accurate. Diamond\_LR has the smallest value of Average MAE, while Diamond\_LR\_1 has the lowest value of Average RMSE. The MAE is very similar to the RMSE, but is less sensitive to large errors. Therefore, we think RMSE is a better indication for the accuracy of a model. For this Cross Validation method, Diamond\_LR\_1 has the highest average RMSE.

Overall, the first method confirms that Diamond\_LR\_1 model predicts a price that has low percentage difference compared to the actual price. The Lift Chart method confirms that Diamond\_LR\_1 is a good fit to the testing data. This model also scores the highest in RMSE, meaning it is more accurate than other models. We come to the conclusion that linear regression model Diamond\_LR\_1 is best one to predict the diamond price.

**V. Limitation of Microsoft SQL Server Analysis Services**

1. The R Square value is only available in he paid version of the tool which made it difficult for us to compare our different regression model.
2. SVM not available in the the unpaid version of MS SQL server analysis, and so we were not able to use that our project.
3. Interpreting the results was not easy (no p-value). Unlike in RapidMiner, where p-value is shown in the analysis, the SQL server does not do so. But at the same time SQL Server has more ways to showing and interpreting the result. You can drill down further to understand the results and the factors that predict the outcome.
4. Cannot use the classification matrix to evaluate the model since we are not answering a yes/no question.

Overall, MS SQL Server is not the best tool in its current form with limited feature. It is also not very user-friendly in terms of using it or the visualization of the results. The Azure version is said to be much advanced in features and easier to use.

**VI. The 3W Questions**

**What Went Well**

1. Getting our dataset imported into SQL server and getting it to work was easier than in Orange.
2. Getting our regression equation to predict the price that was pretty close to the actual prices.
3. We have a fair understanding of SQL Server Analysis tool now, which was the goal when we started this project, to tie it in with our KMBI class where we learnt to build the multidimensional cube.

**What went wrong?**

1. We were limited in our analysis because of the dataset and the tool we chose.
2. It was difficult to understand the tool and the interpret the results. SSAS is not the most user-friendly and visually appealing analysis tool, hence we had to go through a lot of blogs and Youtube tutorials to understand how to use it and make sense of the results.
3. We would have liked to explore more classification algorithms like Decision trees, Naive Bayes, etc. It was due to the dataset we chose which all numeric was.

**What would be do differently?**

1. Choose a dataset with nominal values as well where more classification models could be used.
2. If we use the same dataset, we would choose a different tool that is easier and more informative with linear regression model, in terms of statistical implications (i.e R-squared, p-value). We would look into more open-source tools such as Weka, SPSS, KNMIE to see if these tools work better for linear regression model
3. If we have more time, we will explore more with the SQL query in the Visual Studio. We saw a couple a blog posts about how to calculate R-square value by writing a SQL query. However, the dataset demonstrated in these examples are different, and we do not have the knowledge of SQL query with this tool. We failed to explore the tool in this aspect.