HarvardX PH125.9x Data Science Capstone Diamond Price

Stepan Kravtsov

8 March 2022

1. Overview

Dataset

The dataset used for this project was taken from Kaggle and contains information about price of diamonds and variables that can possibly influence it. The dataset can be found **here**.

This dataset contains variables such as:

- Price
- Carat
- Cut
- Color
- Clarity
- Depth
- Table
- x
- y
- z

The *price* of the diamond is what we are going to predict.

The *carat* is the mass of the diamond (in carats).

The *cut* is the way in which a diamond is faceted to catch and reflect light. It is measured from Poor to Ideal. In this dataset only the diamonds of cut Fair to Ideal are present.

The *color* refers to how clear or yellow the diamond is. It is measured on a scale from D to Z, where D is the clearest and Z the most yellow. In this dataset only the diamonds of colors J to D are present.

The *clarity* is a qualitative metric that grades the visual appearance of each diamond. It is measured from I (included) to IF (internally flawless).

The *depth* refers to the diamond's measurement from top to bottom. It is measured in percent and is calculated by dividing the diamond's total height by its total width. The depth affects the way a diamond reflects light

The *table* is size of the flat facet on the diamond's surface — the large, flat surface facet that you can see when you look at the diamond from above. As the largest facet on a diamond, the table plays a major role in determining how brilliant (sparkly) the diamond is.

X, Y, and Z are the measurements of the diamond.

Goal

The goal of this project is to build a model that would predict the diamond's price using the given parameters. The models will be assessed using Mean Absolute Percent Error (MAPE) since we are trying to predict a continuous variable, and this measure will give us the average error we make (in percent).

MAPE is calculated using the formula

$$MAPE = \frac{100\%}{n} \sum_{1}^{n} \frac{Actual - Predicted}{|Actual|} \tag{1}$$

The Metrics library will be used to calculate MAPE. There is no final goal for MAPE. The goal is just to make it as small as possible.

2. Methods and Analysis

Modifying the dataset

If we look at the dataset, we can see that the variables *cut*, *color*, and *clarity* have character values.

```
##   carat       cut color clarity depth table price    x    y    z
## 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

If we want to use them for predictions, we need to convert them to numerical (*cut* will be measured from 0 to 4 instead of Fair to Ideal, *color* will be measured from 0 to 6 instead of J to D, *clarity* will be measured from 0 to 7 instead of I1 to IF). After the changes the dataset looks like this.

```
##   carat cut color clarity depth table price    x    y    z
## 1  0.23   4     5       1  61.5    55   326 3.95 3.98 2.43
## 2  0.21   3     5       2  59.8    61   326 3.89 3.84 2.31
## 3  0.23   1     5       4  56.9    65   327 4.05 4.07 2.31
## 4  0.29   3     1       3  62.4    58   334 4.20 4.23 2.63
## 5  0.31   1     0       1  63.3    58   335 4.34 4.35 2.75
## 6  0.24   2     0       5  62.8    57   336 3.94 3.96 2.48
```

Also, a diamond can have various dimensions, but still have the same price, so it is hard to predict the price based on them. That is why we are not going to use them. After taking away *x*, *y*, and *z*, the data looks like this.

```
##   carat cut color clarity depth table price
## 1  0.23   4     5       1  61.5    55   326
## 2  0.21   3     5       2  59.8    61   326
## 3  0.23   1     5       4  56.9    65   327
## 4  0.29   3     1       3  62.4    58   334
## 5  0.31   1     0       1  63.3    58   335
## 6  0.24   2     0       5  62.8    57   336
```

Building the models

Splitting the data

The data will be split into train and test sets (90/10). We don't need a lot of testing data, so only 10% will be taken to maximize the training set.

Model 1: A first naive "mean" model

To start with, let's create a model that would just predict the mean price for every diamond. The MAPE for that model can be seen below.

```
## [1] 189.6753
```

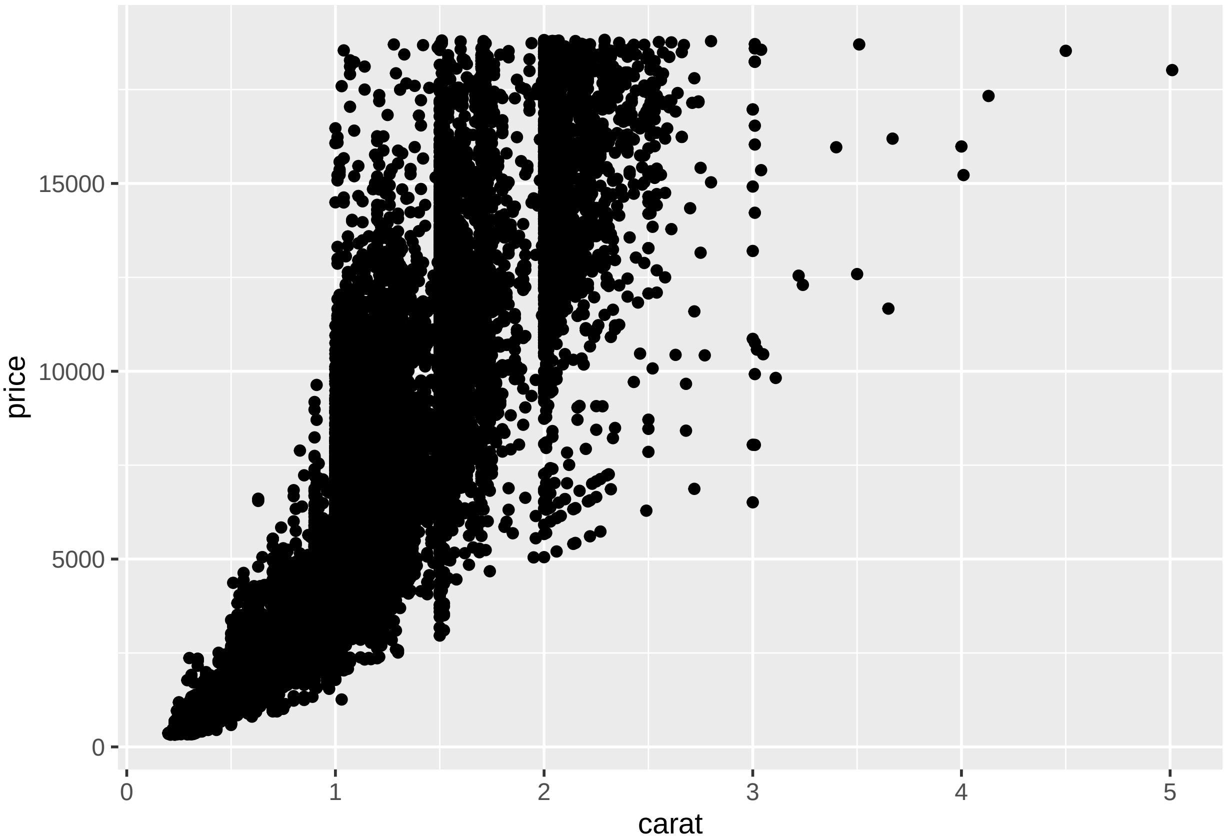
Let's create a table, where we will keep all the results and add the results of the first model to the table.

Table 1: MAPE Results Model 1

Index	Method	MAPE
1	Just the average	189.7 %

Model 2: Linear model, Carat

Let's plot the data for *carat* vs *price*.



We can see that the carat has an effect on the price. So we will create a linear model that will use the *carat* variable to predict the price of the diamond.

The MAPE for this model can be seen below.

```
## [1] 37.59466
```

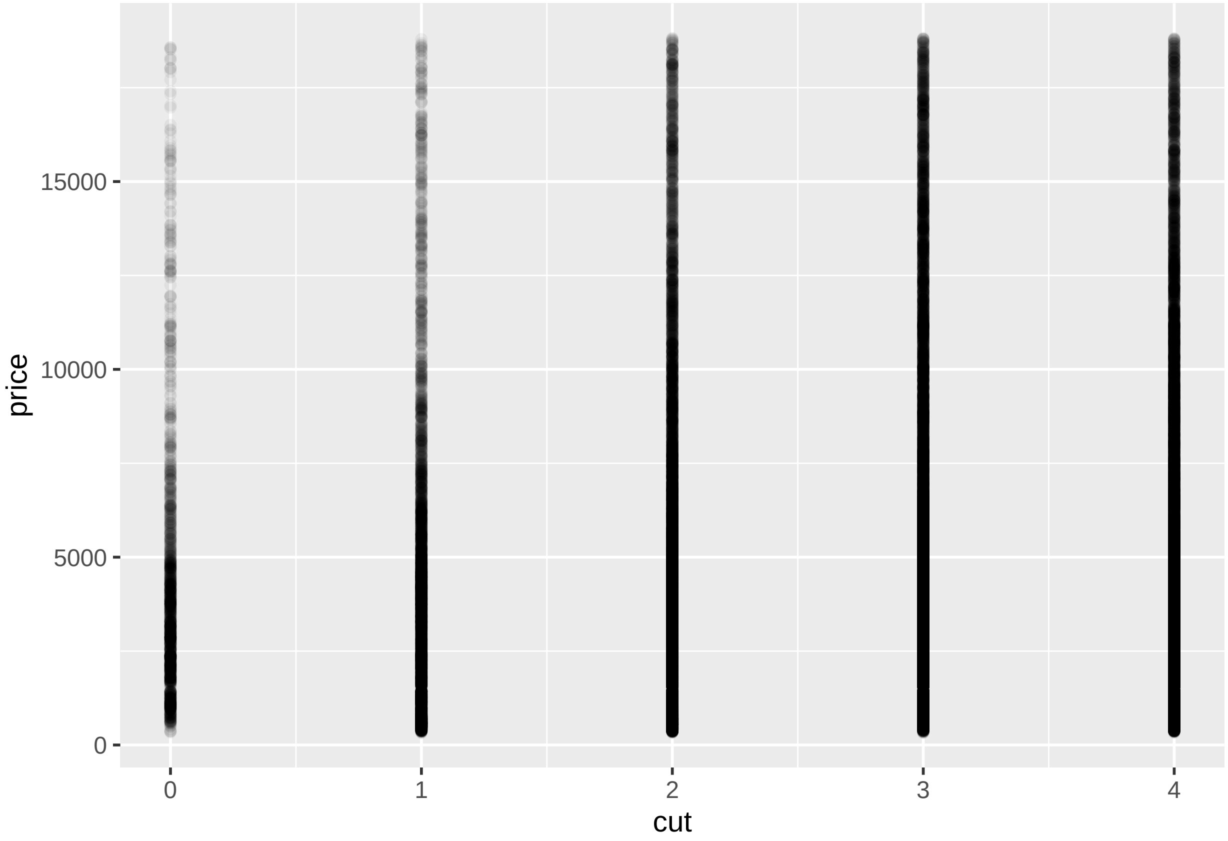
Let's add it to the table.

Table 2: MAPE Results Models 1-2

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %

Model 3: Linear model, Cut

Let's plot the data for *cut* vs *price* and set alpha to 0.05 so we can see the effect better.



We can see that the carat has some effect on the price. Diamonds with higher prices tend to have better cuts. So we will add the *cut* as a predictor to our linear model.

The MAPE for this model can be seen below.

```
## [1] 37.56877
```

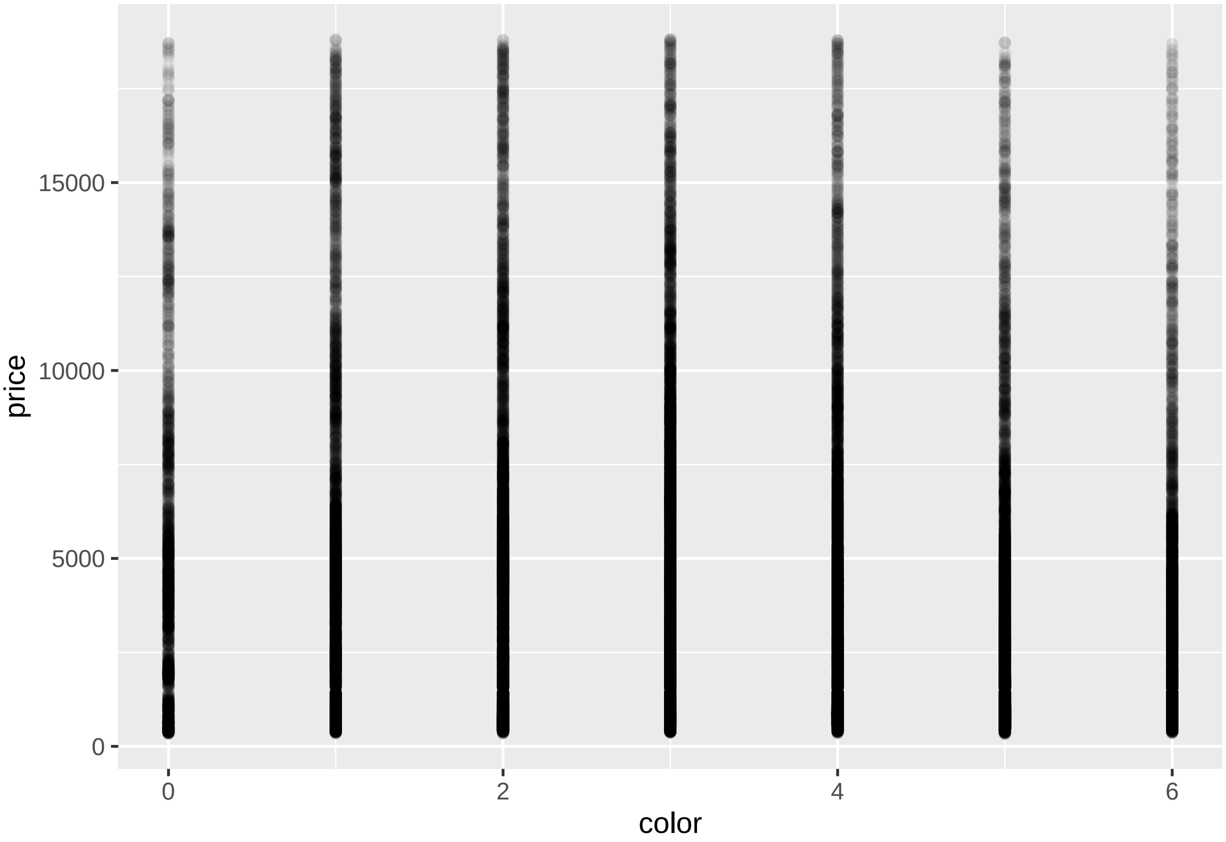
Let's add it to the table.

Table 3: MAPE Results Models 1-3

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %
3	Carat + Cut Effect Model	37.57 %

Model 4: Linear model, Color

Let's plot the data for *color* vs *price* and set alpha to 0.05 so we can see the effect better.



We can see that the carat has some effect on the price. The diamond prices are not uniformly distributed among colors. So we will add the *color* as a predictor to our linear model.

The MAPE for this model can be seen below.

```
## [1] 43.21955
```

Let's add it to the table.

Table 4: MAPE Results Models 1-4

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %
3	Carat + Cut Effect Model	37.57 %
4	Carat + Cut + Color Effect Model	43.22 %

Models 5-7: Linear models, Clarity, Depth, Table

Let's try to add the remaining variables (Clarity, Depth, Table) to our models one by one to see how MAPE would change.

The MAPE for the models can be seen below.

```
## [1] 50.35417
```

```
## [1] 50.4464
```

```
## [1] 50.46747
```

Let's add them to the table.

Table 5: MAPE Results Models 1-7

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %
3	Carat + Cut Effect Model	37.57 %
4	Carat + Cut + Color Effect Model	43.22 %
5	Carat + Cut + Color + Clarity Effect Model	50.35 %
6	Carat + Cut + Color + Clarity + Depth Effect Model	50.45 %
7	Carat + Cut + Color + Clarity + Depth + Table Effect Model	50.47 %

If we look at our results, we see that all the variables added after cut only decreased the accuracy of our model. Looks like linear models aren't the best for this dataset.

Model 8: KNN model

Let's create a KNN model that would predict the price using the neighbouring points.

The MAPE for the model can be seen below.

```
## [1] 14.90634
```

Let's add it to the table.

Table 6: MAPE Results Models 1-8

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %
3	Carat + Cut Effect Model	37.57 %
4	Carat + Cut + Color Effect Model	43.22 %
5	Carat + Cut + Color + Clarity Effect Model	50.35 %
6	Carat + Cut + Color + Clarity + Depth Effect Model	50.45 %
7	Carat + Cut + Color + Clarity + Depth + Table Effect Model	50.47 %
8	KNN Model	14.91 %

Model 9: Projection pursuit regression model

While looking through the possible models I can make, I stumbled upon a Projection pursuit regression (PPR) model, which is an extension of additive models and is really similar to a neural network. I decided to try to create one for this dataset.

The MAPE for the model can be seen below.

```
## [1] 13.01227
```

Let's add it to the table.

Table 7: MAPE Results Models 1-9

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %
3	Carat + Cut Effect Model	37.57 %
4	Carat + Cut + Color Effect Model	43.22 %
5	Carat + Cut + Color + Clarity Effect Model	50.35 %
6	Carat + Cut + Color + Clarity + Depth Effect Model	50.45 %
7	Carat + Cut + Color + Clarity + Depth + Table Effect Model	50.47 %
8	KNN Model	14.91 %
9	PPR Model	13.01 %

Model 10: Gradient boosting

Another method that I found was the gradient boosting, which is a machine learning technique that gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. I creted one for this dataset.

The MAPE for the model can be seen below.

```
## [1] 11.55999
```

Let's add it to the table.

Table 8: MAPE Results Models 1-10

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %
3	Carat + Cut Effect Model	37.57 %
4	Carat + Cut + Color Effect Model	43.22 %
5	Carat + Cut + Color + Clarity Effect Model	50.35 %
6	Carat + Cut + Color + Clarity + Depth Effect Model	50.45 %
7	Carat + Cut + Color + Clarity + Depth + Table Effect Model	50.47 %
8	KNN Model	14.91 %
9	PPR Model	13.01 %
10	Gradient Boosting Model	11.56 %

Results

The results of all the models can be seen in the table below.

Table 9: Final Reults

Index	Method	MAPE
1	Just the average	189.7 %
2	Carat Effect Model	37.59 %
3	Carat + Cut Effect Model	37.57 %
4	Carat + Cut + Color Effect Model	43.22 %
5	Carat + Cut + Color + Clarity Effect Model	50.35 %
6	Carat + Cut + Color + Clarity + Depth Effect Model	50.45 %
7	Carat + Cut + Color + Clarity + Depth + Table Effect Model	50.47 %
8	KNN Model	14.91 %
9	PPR Model	13.01 %
10	Gradient Boosting Model	11.56 %

As we can see from the table, linear model was not the best method for this dataset. The best MAPE achieved with linear models is 37.5687684. Some of the other methods used, including KNN, PPR, and Gradient Boosting, proved to be more effective in predicting the price of a diamond with the given parameters. The lowest MAPE (11.5599855) was achieved with Gradient Boosting.

Conclusion

In this project, I used different machine learning methods to predict the price of a diamond. The methods used include linear models, KNN, projection pursuit regression, and gradient boosting. Linear models weren't very effective since after a certain point adding new predictors only increased the MAPE of the model. There was no specific goal for the MAPE achieved, and I think that the MAPEs achieved were low enough. The best model was able to, on average, predict the price within 11.5599855 percent of the actual price.

One of the biggest limitations was the computation time. For example, some methods like random forests weren't tested since they took too much time to create the models (the time was tested on smaller samples of data).

The future work may include using other machine learning methods to decrease MAPE even further.