**Week 5: Cloud and API deployment**

Name: Sebastián J. Castro

Batch code: LISP01

Submissionn date: 13/05/2021

Submitted to: Data Glacier

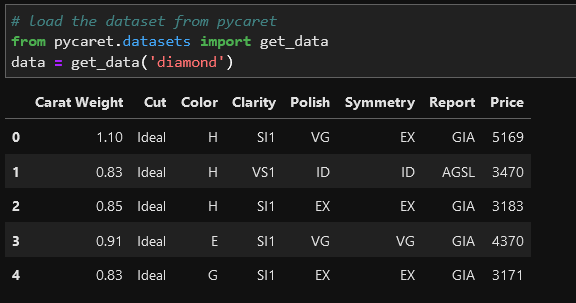
What tools will I use in this task?

1. Pycaret:   
   Pycaret is an open-source, low-code machine learning library and end-to-end model management tool built-in Python for automating machine learning workflows. It is easy to use, simple, and it enable ML model deployment quickly and efficiently.
2. FastAPI:  
   FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints. It is fast (very high performance), one of the fastest python frameworks available, indeed. It is fast and easy to code.

Steps:

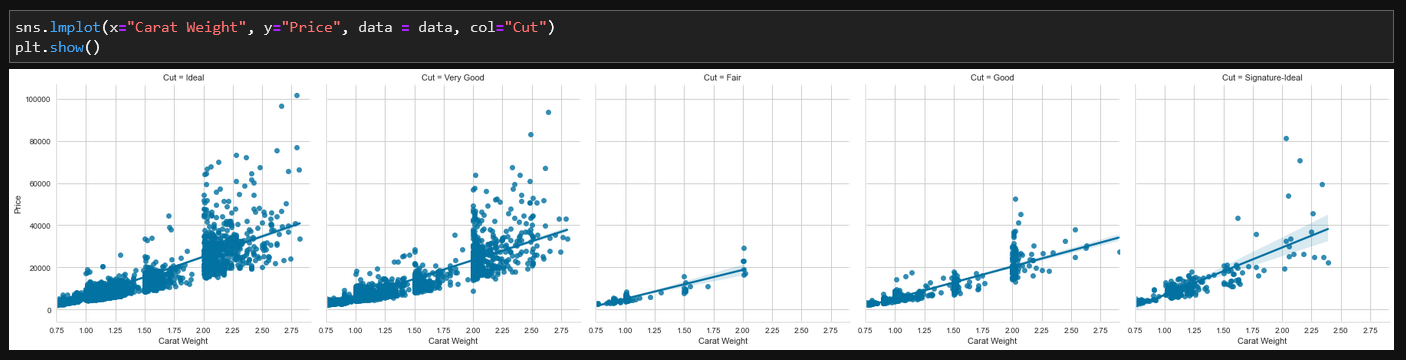
For this task, I will be using a very popular case of study by Darden School of Business, published in Harvard Business. The case is regarding the story of two people who are going to be married in the future. The guy named Greg wanted to buy a ring to propose to a girl named Sarah. The problem is to find the ring Sarah will like, but after a suggestion from his close friend, Greg decides to buy a diamond stone instead so that Sarah can decide her choice. Greg then collects data of 6000 diamonds with their price and attributes like cut, color, shape, etc.

1. Importing dataset:



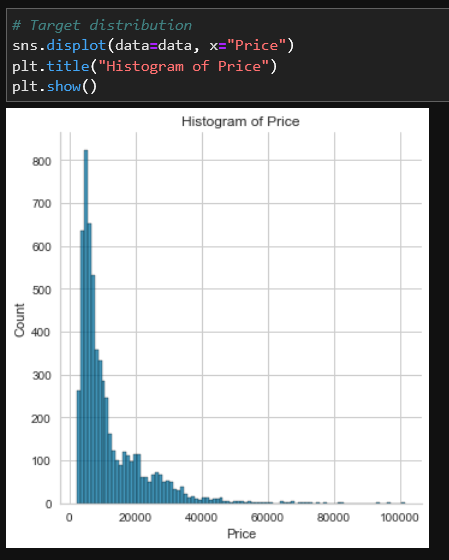
1. Quick visualization:

In this step we are looking for asses the relationship of independent features (weight, cut, color, clarity, etc.) with the target variable “Price”.



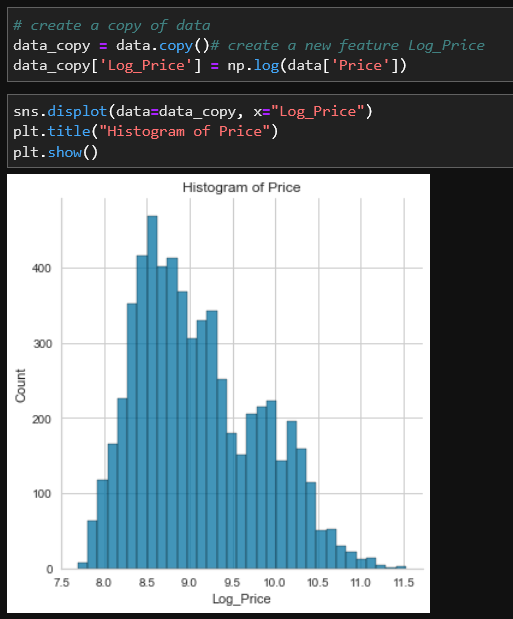
1. Target distribution:

Now we check the distribution of the target variable.



The distribution of “Price” is right-skewed, we can quickly check to see if log transformation can make “Price” approximately normal to give fighting chance to algorithms that assume normality.

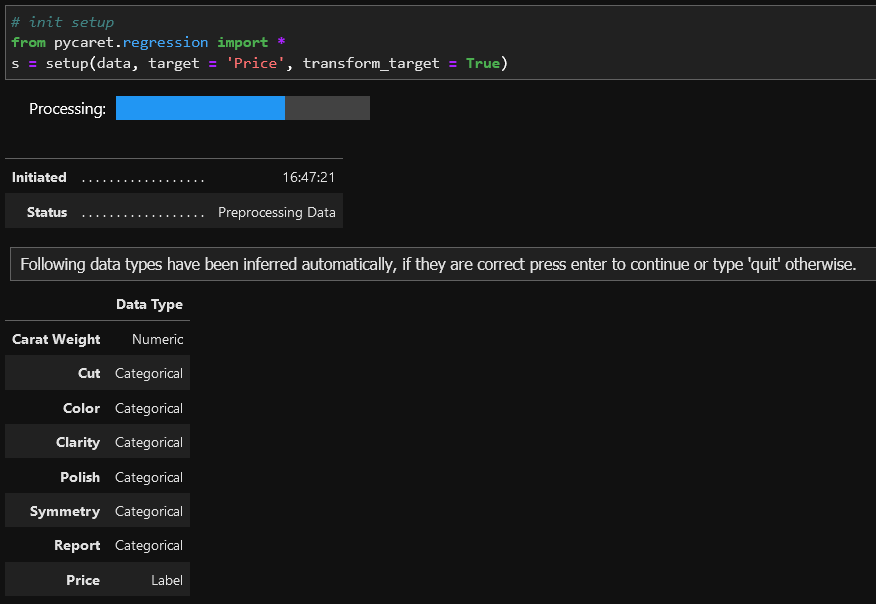
1. Target variable log transformation



This confirms our hypothesis. The transformation will help us to get away with skewness and make the target variable approximately normal. Based on this, we will transform the “Price” variable before training our models.

1. Data preparation

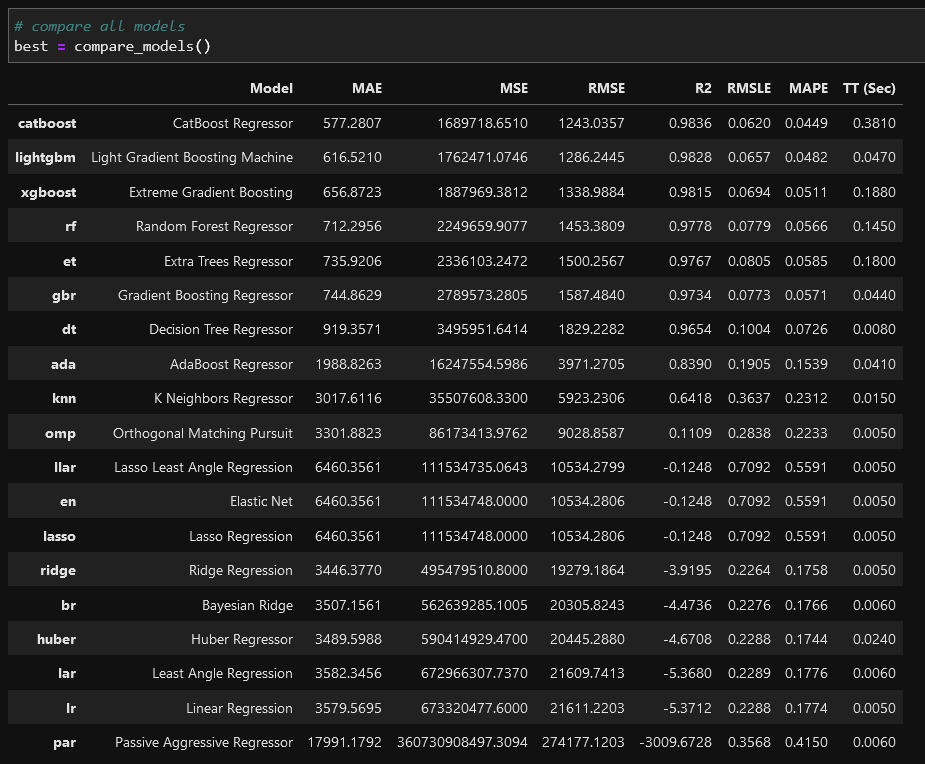
Common to all modules in PyCaret, the “setup” is the first and the only mandatory step in any machine learning experiment performed in PyCaret. This function takes care of all the data preparation required prior to training models. Besides performing some basic default processing tasks, PyCaret also offers a wide array of pre-processing features.



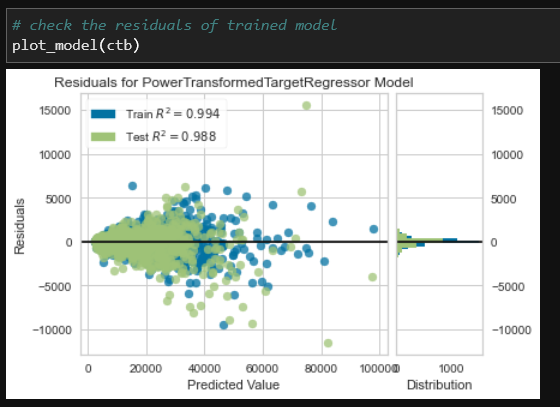
Pycaret setup function infers data types. Besides, Pycaret will transform the “Price” variable behind the scene using box-cox transformation as transform\_target = True. It will affect the distribution of the target in a similar way as log transformation.

1. Model Training and Selection

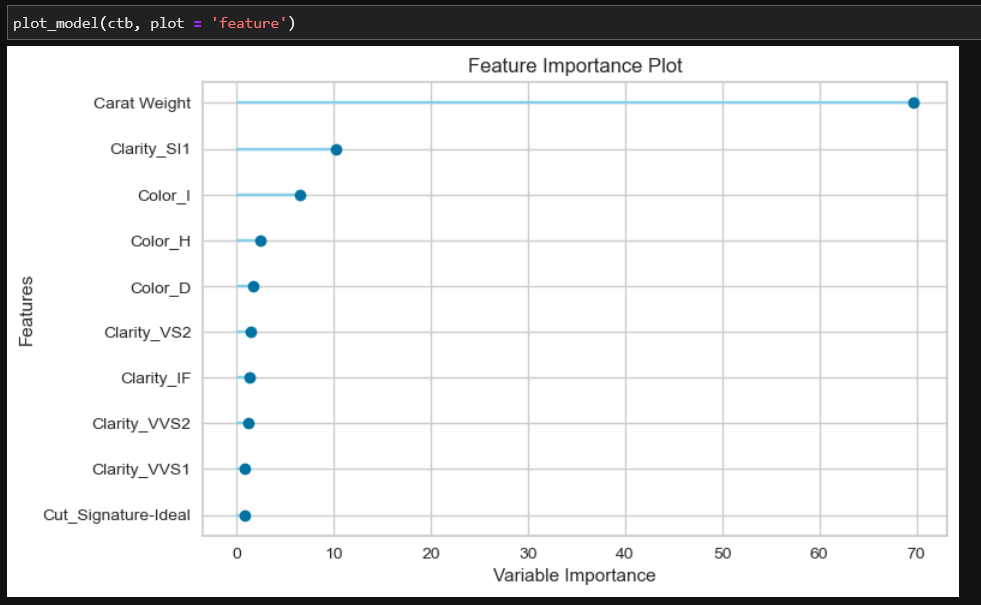
Now that data preparation is done, we can start the training process by using compare\_models functionality. This function trains all the algorithms available in the model library and evaluates multiple performance metrics using cross-validation.



The best model based on Mean Absolute Error (MAE) is CatBoost Regressor. MAE using 10-fold cross-validation is $577 compared to the average diamond value of $11,600. This is less than 5%. Not bad for the efforts we have put in so far. Let’s see the residuals of the trained model:

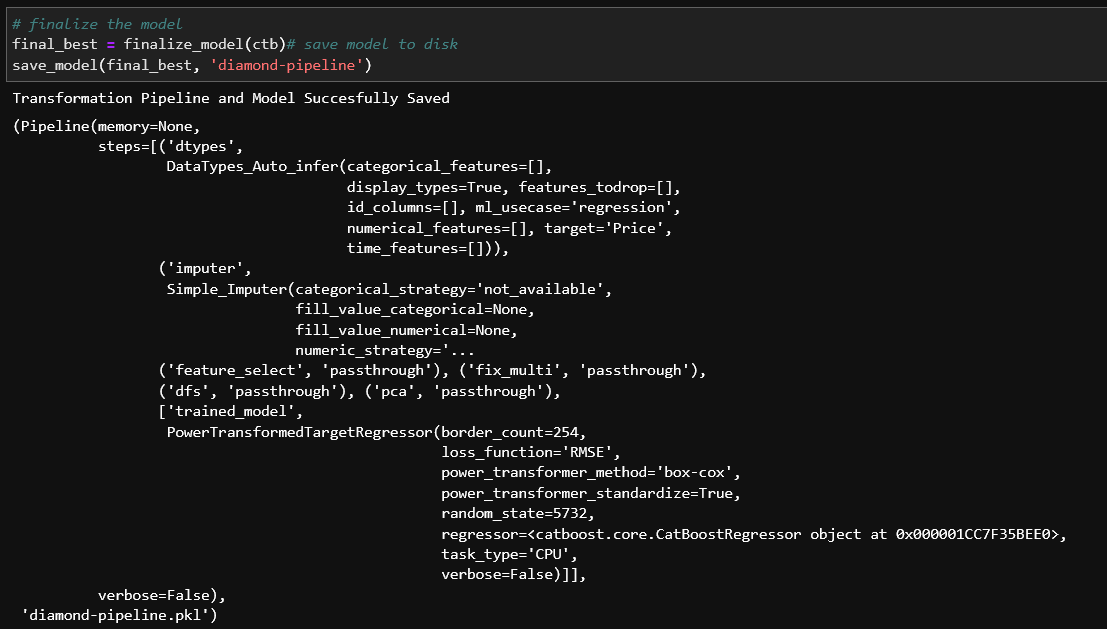


We can also check the feature importance:



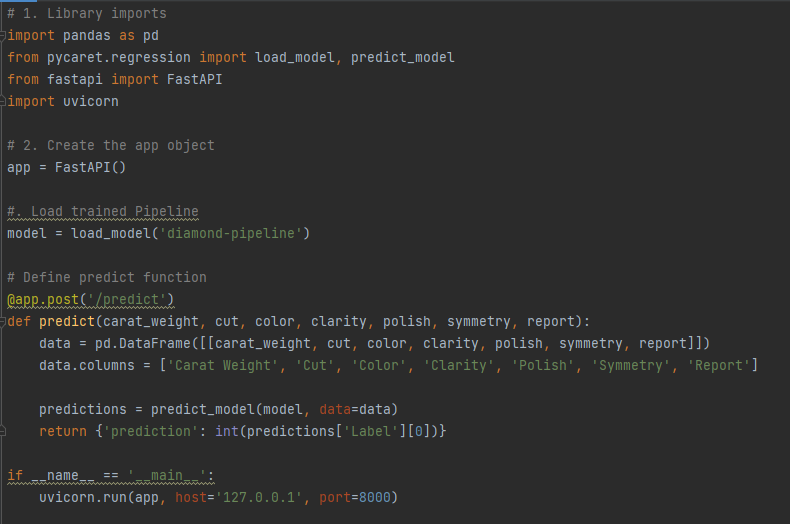
1. Finalize and Save Pipeline:

Let’s now finalize the best model i.e. train the best model on the entire dataset including the test set and then save the pipeline as a pickle file.



1. Deployment

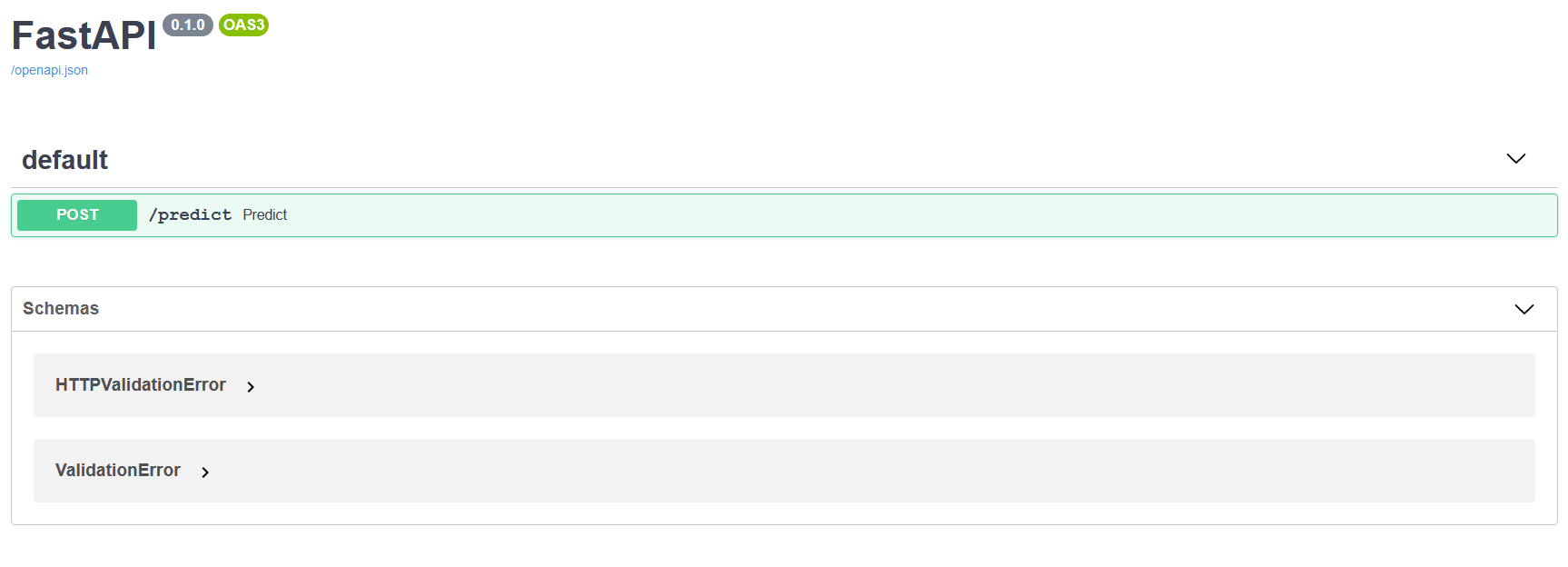
We will create an API using FastAPI. The main.py file necessary for doing that is the following:



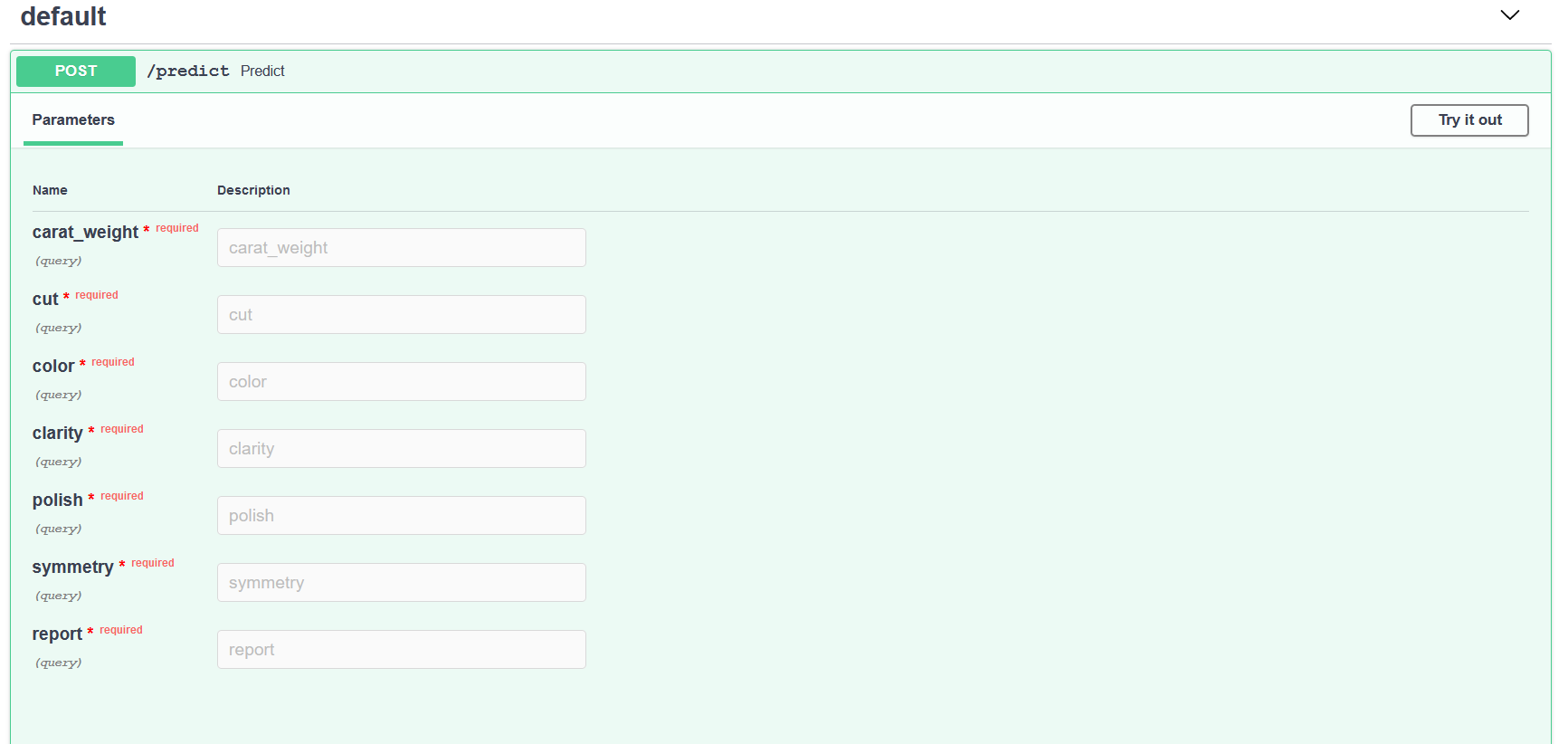
We then run this script by running the following command in the command prompt.



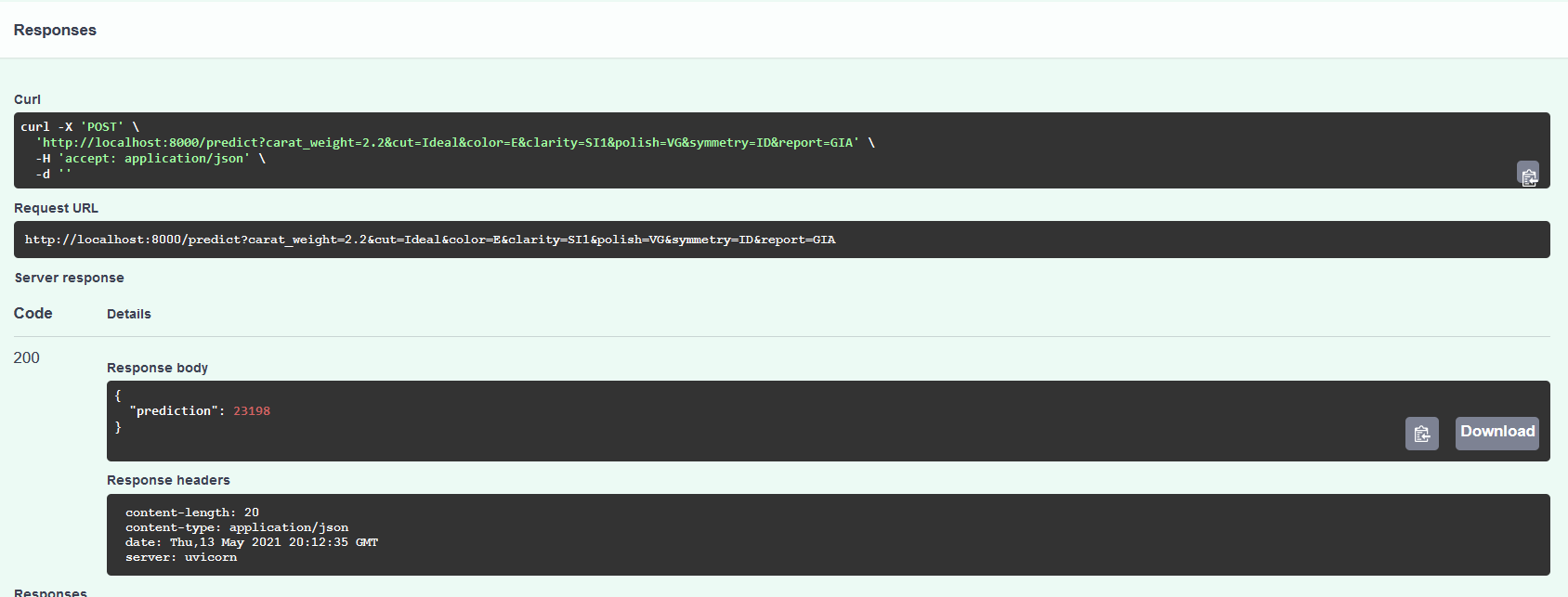
This will initialize an API service on localhost. Then, we have to type <http://localhost:8000/docs> at a browser and it should show something like this:



Then we click on “POST” button and it will open a form like this:



Then, we click on “Try it out” and fill in some values in the form and click on “Execute”.



Under the response body we have a prediction value of 23198 (this is based on values I entered in the form). This means that given all the attributes I have entered, the predicted price of this diamond is $23,198.

We can use the request library of Python to connect to API and generate predictions.

