Kaggle Diamonds Competition 💎

Ana Walsh

Index

1. GOAL 🏁

The goal of this project is to find the best machine learning model for a given dataset from kaggle.

The model should be able to make predictions for the price of diamonds.

We were asked to do this project at the Ironhack Data Analytics Bootcamp.

2. STRUCTURE OF THE PROJECT FILES

- Notebooks: one for cleaning the data and another one for doing predictions.
- data: Diamonds dataset from kaggle.
- Predictions: csv with the results of the predictions that were made.

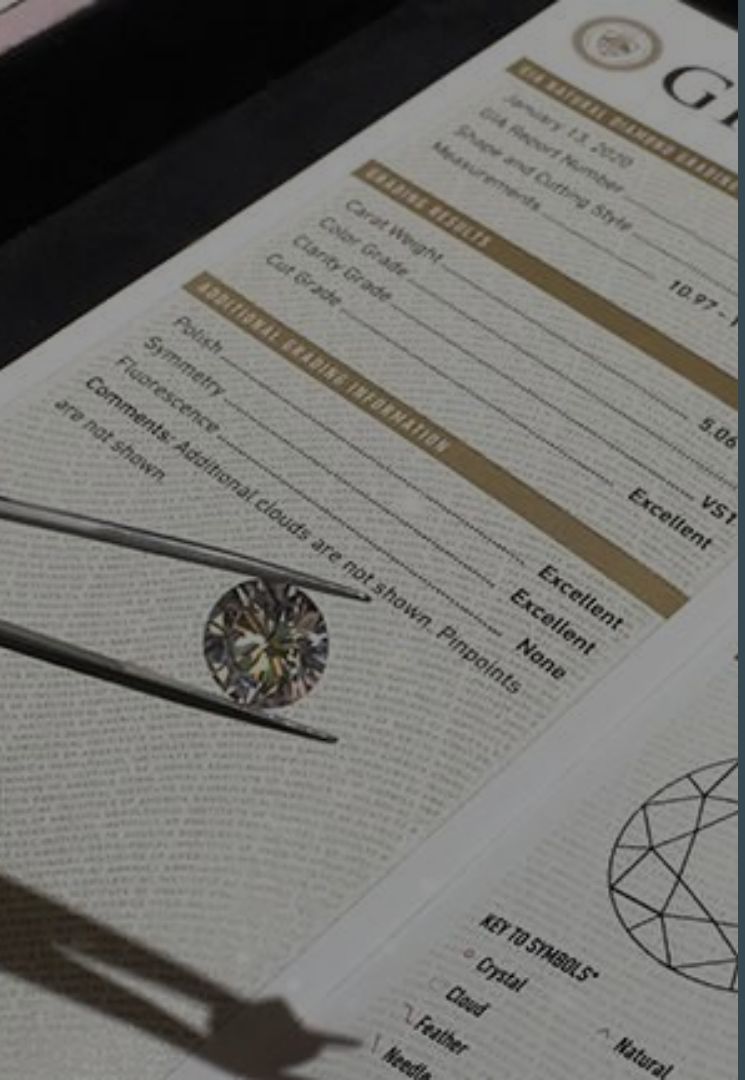
3. WORKING PLAN 👩‍💻

This is a sample of our initial dataset, of which we will have one for training the model and one for testing. The first thing we need to do is to adapt our dataset. To do this we will make 2 changes:

1 - Convert categorical variables to ordinal variables.

2 - Delete those columns that present collinearity.

	id	carat	cut	color	clarity	depth	table	x	y	z	price
0	0	1.14	Ideal	G	VVS2	61.0	56.0	6.74	6.76	4.12	9013
1	1	0.76	Ideal	H	VS2	62.7	57.0	5.86	5.82	3.66	2692
2	2	0.84	Ideal	G	VS1	61.4	56.0	6.04	6.15	3.74	4372
3	3	1.55	Ideal	H	VS1	62.0	57.0	7.37	7.43	4.59	13665
4	4	0.30	Ideal	G	SI2	61.9	57.0	4.28	4.31	2.66	422



3.1. EXPLORE AND CLEAN DATASET 🧐

We have followed the criteria provided by the GIA (Gemological Institute of America) to re-categorize some columns of our dataset:

3.1.1. Re-categorize columns: “CUT”

Cut quality is the factor that fuels a diamond’s fire, sparkle and brilliance. The allure and beauty of a particular diamond depends more on cut quality than anything else.

1. Premium
2. Ideal
3. Very good
4. Good
5. Fair

3.1. EXPLORE AND CLEAN DATASET 🧐

3.1.1. Re-categorize columns

“COLOR”

GIA categorizes diamond according to colour into five groups:

1. "Colorless" (D-F): These are the most rare, and therefore the most valuable.
2. "Near colorless" (G-J): Color is often unnoticeable except by trained graders.
3. "Faint" (K-M) : Color is still difficult to see by the untrained eye.
4. "Very light" (N-R) : Subtle color can be Seen in larger stones by an untrained eye.
5. "Light" (S-Z) : Color can be seen in stones of different sizes. The diamonds appear slightly yellow or brown but do not have sufficient color to be considered a "fancy" colored diamond.

3.1. Explore and clean dataset 🧐

3.1.1. Re-categorize columns: “CLARITY”

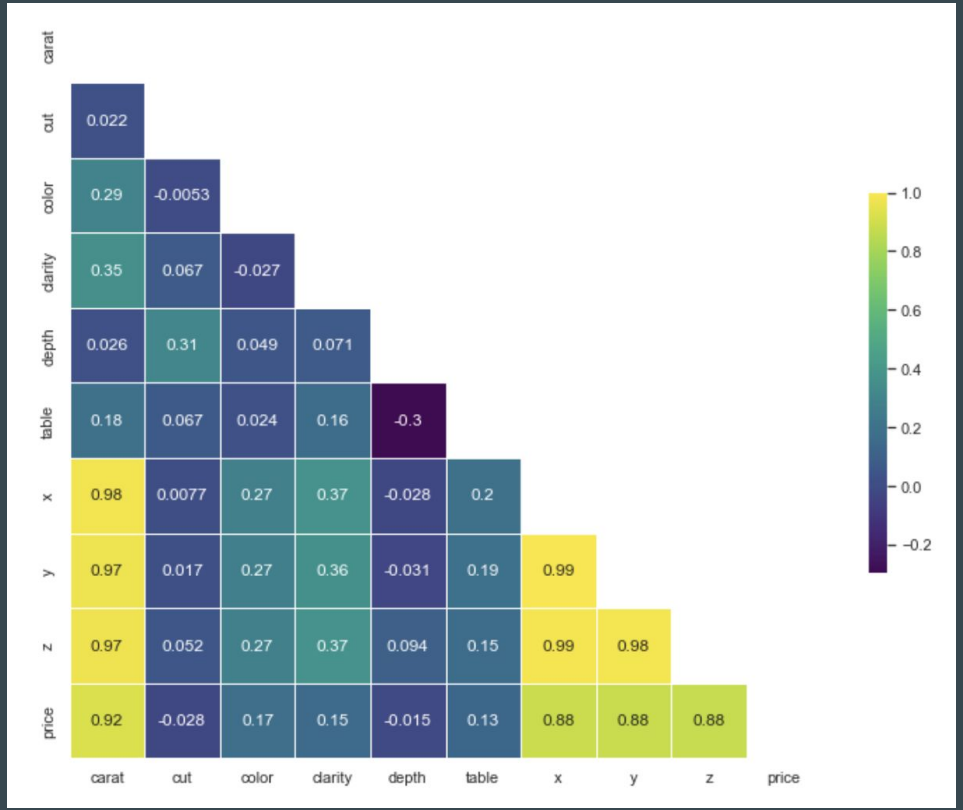
The GIA Clarity Scale contains 11 grades, with most diamonds falling into the VS (very slightly included) or SI (slightly included) categories. In determining a clarity grade, the GIA system considers the size, nature, position, color or relief, and quantity of clarity characteristics visible under 10× magnification

1. Flawless (FL)
2. Internally Flawless (IF)
3. Very, Very Slightly Included (VVS1 and VVS2)
4. Very Slightly Included (VS1 and VS2)
5. Slightly Included (SI1 and SI2)
6. Included (I1, I2, and I3)

3.1. Explore and clean dataset 🧐

	id	carat	cut	color	clarity	depth	table	x	y	z	price
0	0	1.14	Ideal	G	VVS2	61.0	56.0	6.74	6.76	4.12	9013
1	1	0.76	Ideal	H	VS2	62.7	57.0	5.86	5.82	3.66	2692
2	2	0.84	Ideal	G	VS1	61.4	56.0	6.04	6.15	3.74	4372
3	3	1.55	Ideal	H	VS1	62.0	57.0	7.37	7.43	4.59	13665
4	4	0.30	Ideal	G	SI2	61.9	57.0	4.28	4.31	2.66	422

According to the collinearity matrix, the columns showing a sufficient value are "x", "y" and "z".



3.1. Explore and clean dataset 🧐

According to the collinearity matrix, the columns showing a sufficient value are "x", "y" and "z".

- Train : part of the dataset we use to train our models.
- Test: part of the dataset that we use to make diamond price predictions.

This is what our resultant dataset looks like:

	carat	cut	color	clarity	depth	table	price
0	1.14	2	4	3	61.0	56.0	9013
1	0.76	2	5	5	62.7	57.0	2692
2	0.84	2	4	4	61.4	56.0	4372
3	1.55	2	5	4	62.0	57.0	13665
4	0.30	2	4	7	61.9	57.0	422

3.2. MAKING PREDICTIONS 🤔

Model	RMSE
Linear Regression	1244.55
Ridge	1244.53
Lasso	1244.48
SGD	99594178.25
KNeighbors	1898.34
Random Forest	557.09
Gradient	595.05

We use the RMSE as an indicator to select the best model. In this case we use the Random Forest and Gradient models with the best parameters:

- Random Forest Regressor Best Parameters:

max_depth: 200, max_features: auto, min_samples_leaf: 2, n_estimators: 300

- Gradient Boosting Regressor:

leaning_rate: 0.1, max_depth: 4, max_features:1.0, min_samples_leaf:3, n_estimators: 500

3.2. MAKING PREDICTIONS 🤔

Below are the price predictions obtained with each model:

1. Random Forest

id	price
0	3297.821046
1	2999.356409
2	3457.443203
3	3127.053393
4	5335.527565

2. Gradient

id	price
0	3297.821046
1	2999.356409
2	3457.443203
3	3127.053393
4	5335.527565

4. Tools and technologies 🛠️

- Pandas
- Matplotlib
- Seaborn
- Numpy
- Sklearn

TO DO 📝

1. Compare test and train to analyze overfitting.
2. Train models without previously deleting columns that showed collinearity.
3. Give H2O a try.