Diamond Price Prediction

Kaggle competition organized by SHAI for AI

Team -7

Lujain Moualla

Team Member

Ahmed Alsayed

Team Leader

Mohammad Ghannam

Team Member

AGENDA

Project Life Cycle

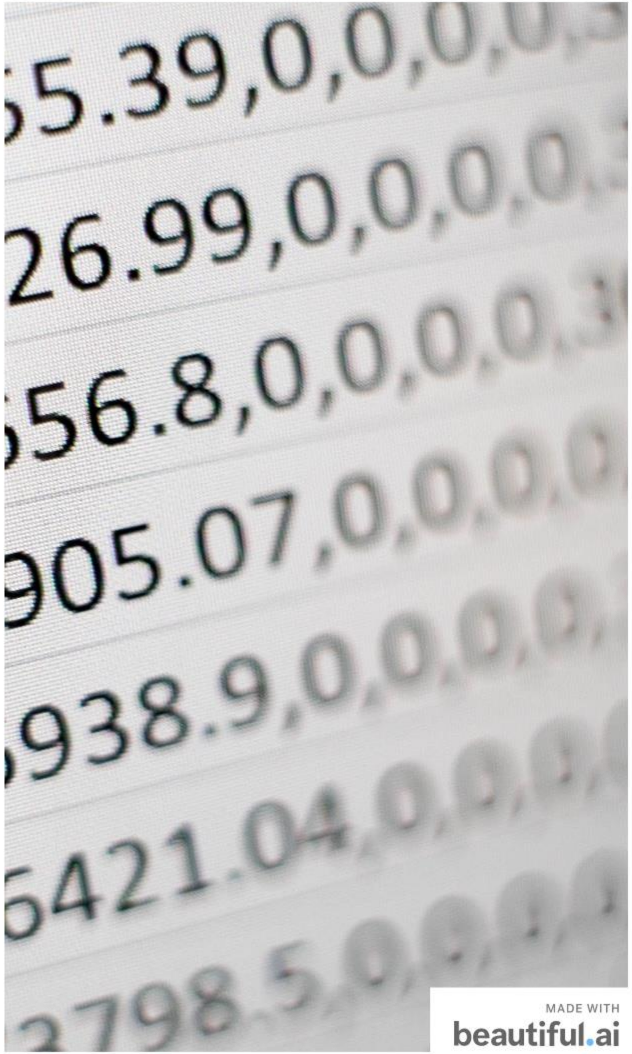
1. Look at the Big Picture
2. Getting The Data
3. Explore and Visualize the Data
4. Feature Engineering
5. Prepare The Data for Machine Learning Algorithms
6. Select a Model and Train it
7. Fine-tune The Model
8. Present Solutions

Look at the Big Picture

Data Overview

- **Price:** price in US dollars ($326 -- $18,823).
- **Carat:** weight of the diamond (0.2 -- 5.01).
- **Cut:** quality of the cut (Fair, Good, Very Good, Premium, Ideal).
- **Color:** diamond color, from J (worst) to D (best).
- **Clarity:** a measurement of how clear the diamond is (I1 (worst), SI1, SI2, VS2, VS1, VVS2, VVS1, IF (best)).
- **X:** length in mm (0 -- 10.74).
- **Y:** width in mm (0 -- 58.9).
- **Z:** depth in mm (0 -- 31.8).
- **Depth:** total depth percentage = z/ mean(x,y) = 2*z/ (x+y) (43 -- 79).
- **Table:** width of the top of diamond relative to wildest point (443 -- 95).

Get the Data



Get the Data

Reading the data:

	Id	carat	cut	color	clarity	depth	table	price	x	y	z
0	1	1.06	Ideal	I	SI2	61.8	57.0	4270	6.57	6.60	4.07
1	2	1.51	Premium	G	VVS2	60.9	58.0	15164	7.38	7.42	4.51
2	3	0.32	Ideal	F	VS2	61.3	56.0	828	4.43	4.41	2.71
3	4	0.53	Ideal	G	VS2	61.2	56.0	1577	5.19	5.22	3.19
4	5	0.70	Premium	H	VVS2	61.0	57.0	2596	5.76	5.72	3.50

Get the Data

Understanding Structure

A- Data information:

There are no null values in any column. Data types include float64 for 6 columns, int64 for 2, and object (likely categorical) for 3.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43152 entries, 0 to 43151
Data columns (total 11 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Id       43152 non-null  int64  
 1   carat    43152 non-null  float64
 2   cut      43152 non-null  object 
 3   color    43152 non-null  object 
 4   clarity  43152 non-null  object 
 5   depth    43152 non-null  float64
 6   table    43152 non-null  float64
 7   price    43152 non-null  int64  
 8   x        43152 non-null  float64
 9   y        43152 non-null  float64
 10  z        43152 non-null  float64
dtypes: float64(6), int64(2), object(3)
memory usage: 3.6+ MB
```

Get the Data

Understanding Structure

B- Statistical information:

	Id	carat	depth	table	price	x	y	z
count	43152.000000	43152.000000	43152.000000	43152.000000	43152.000000	43152.000000	43152.000000	43152.000000
mean	21576.500000	0.797855	61.747177	57.458347	3929.491912	5.731568	5.735018	3.538568
std	12457.053745	0.473594	1.435454	2.233904	3985.527795	1.121279	1.148809	0.708238
min	1.000000	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	10788.750000	0.400000	61.000000	56.000000	947.750000	4.710000	4.720000	2.910000
50%	21576.500000	0.700000	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	32364.250000	1.040000	62.500000	59.000000	5312.000000	6.540000	6.540000	4.040000
max	43152.000000	5.010000	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

Get the Data

Understanding Structure

C- Catigorical features information

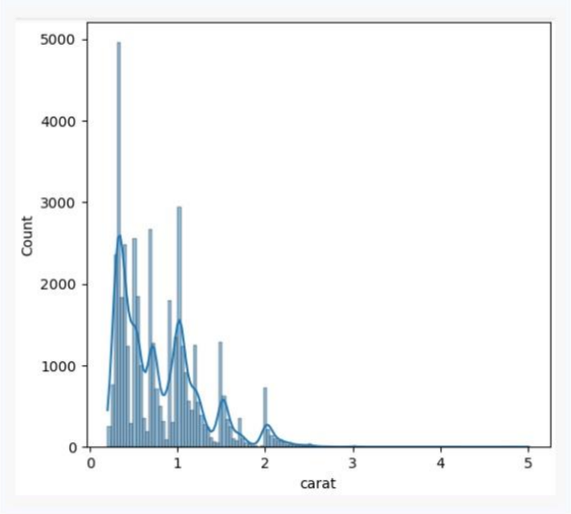
- **Cut:** Has 5 unique categories with "Ideal" being
  the most frequent (17203 instances).
- **Color:** Has 7 unique categories, with "G" being
  the most frequent (9060 instances).
- **Clarity:** Has 8 unique categories, and "SI1" appearing
  most often (10428 instances).

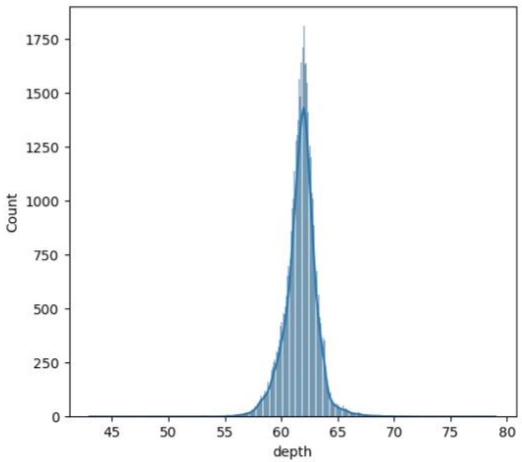
All three features have the same count (43152), indicating no missing values.

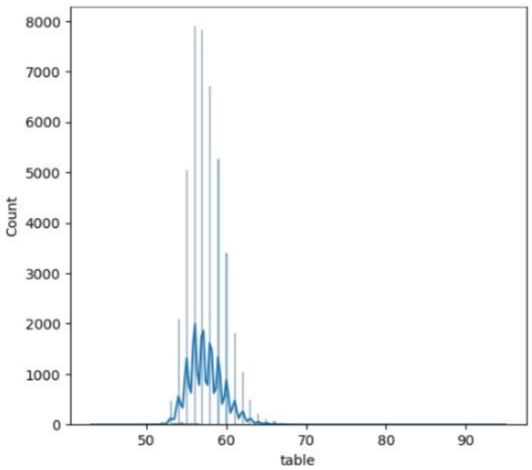
	cut	color	clarity
count	43152	43152	43152
unique	5	7	8
top	Ideal	G	SI1
freq	17203	9060	10428

Explore and Visualize the Data to Gain Insights

Histogram plot







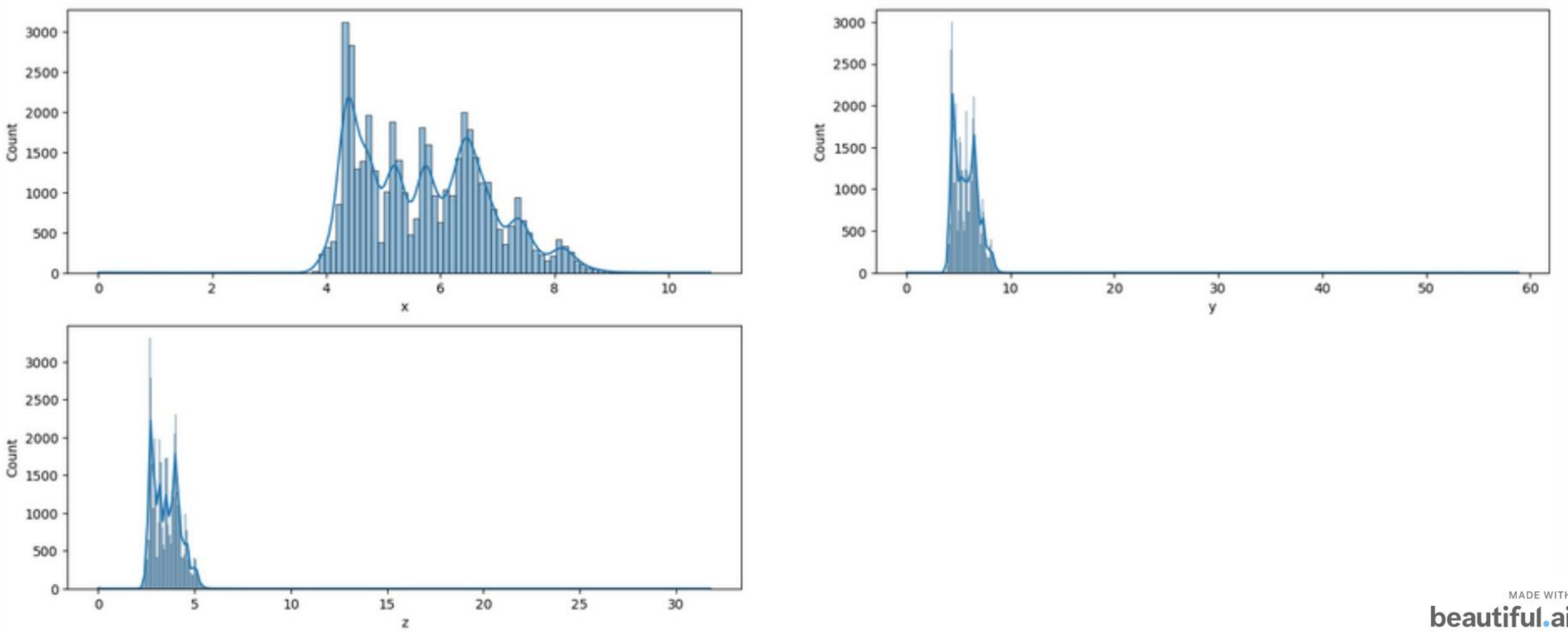
Carat: A right-skewed distribution concentrated between 0.2 to 1.5 carats, indicating most diamonds are smaller. A long tail extends towards larger carat weights, suggesting fewer larger diamonds.

Depth: A roughly normal distribution centered around 62%,suggesting most diamonds fall within a typical depth range.

Table: Shows a slight right skew with the majority of values between 55% to 60%.

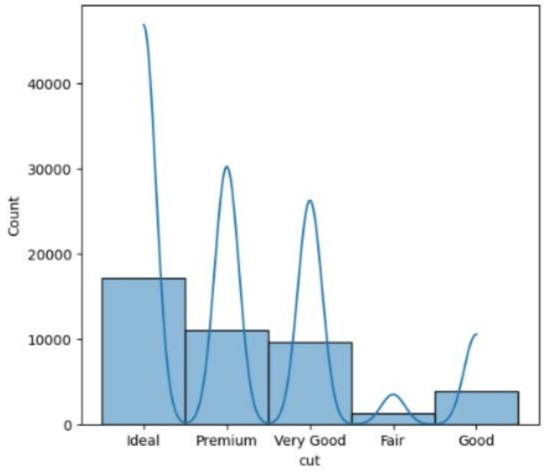
Histogram plot

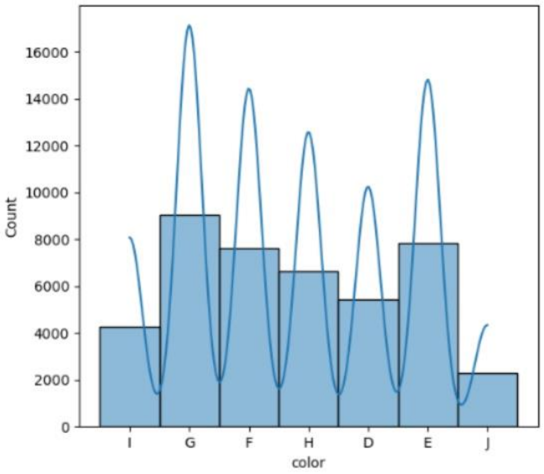
Dimensions (x, y, z): Shows a right skew (y,z) indicating most diamonds are small.

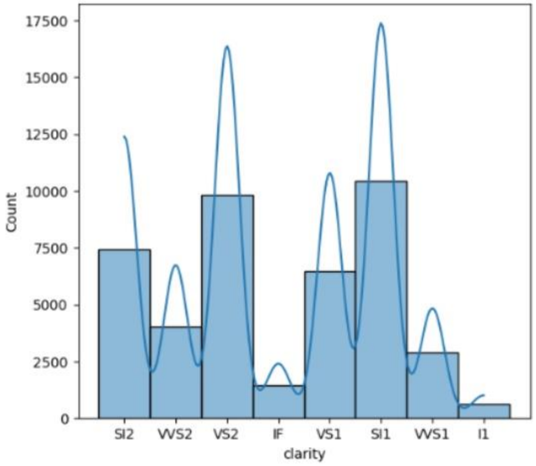


Histogram plot

Categorical Features:







Cut: 'Ideal' is the most frequent cut grade, followed by 'Premium', 'Very Good', 'Good', and 'Fair'.

Color: The distribution across color is with 'G' being the most frequent, followed by descending frequencies for other grades.

Clarity: 'SI1' clarity grade appears most often, followed by 'VS2', 'SI2', 'VS1', 'VVS2', 'VVS1', 'IF', and 'I1'.

Correlation Matrix

Strong Positive Correlations:

Carat &b Dimensions (x, y, z)

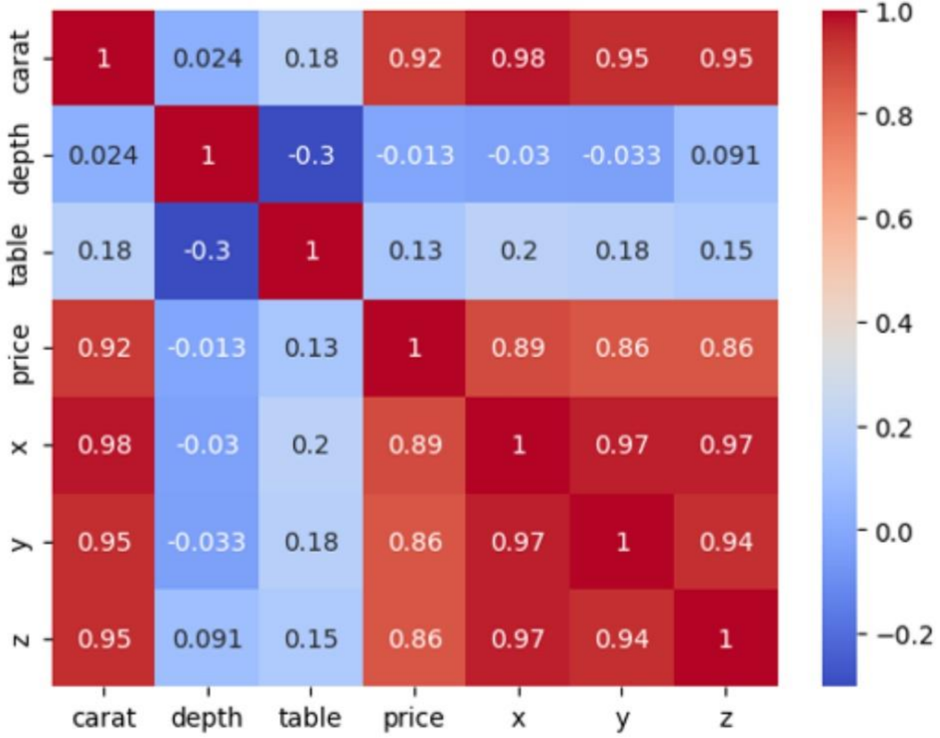
Price & Dimensions (x, y, z)

Price & Carat

Weak Correlations:

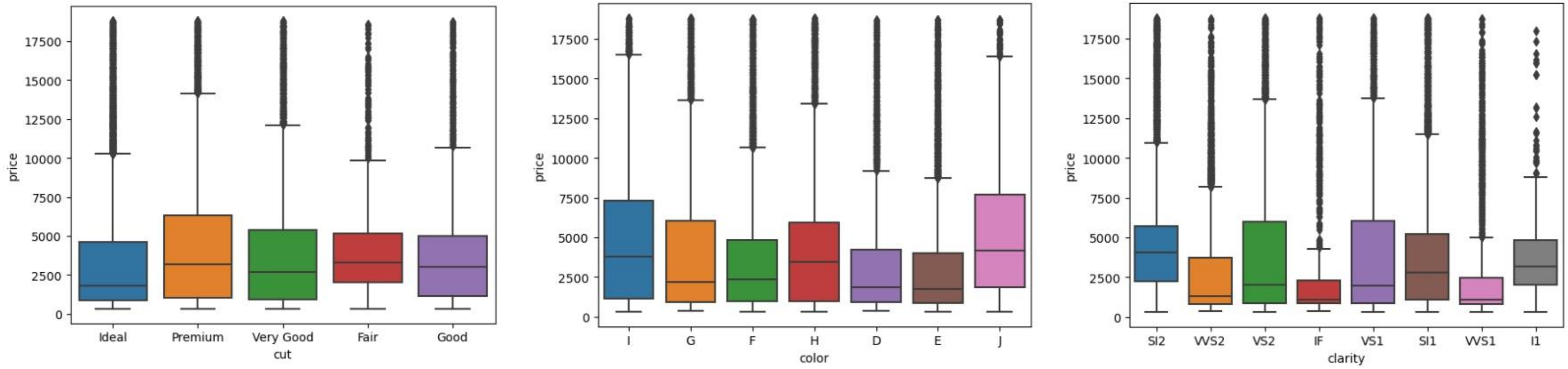
Depth & Table

Price & Depth/ Table



Boxplot

- **Price Overlap:** Significant overlap in price ranges exists across all clarity, color and cut grades, indicating that each one alone is not a strong predictor of price.

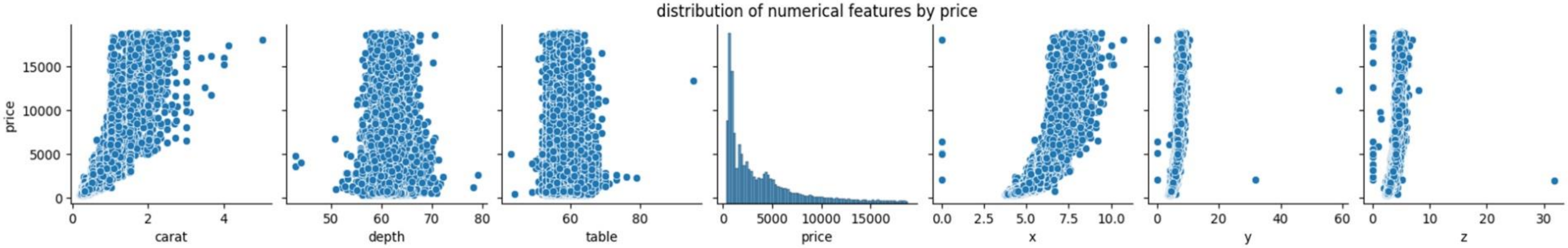


Boxplot for categorical features

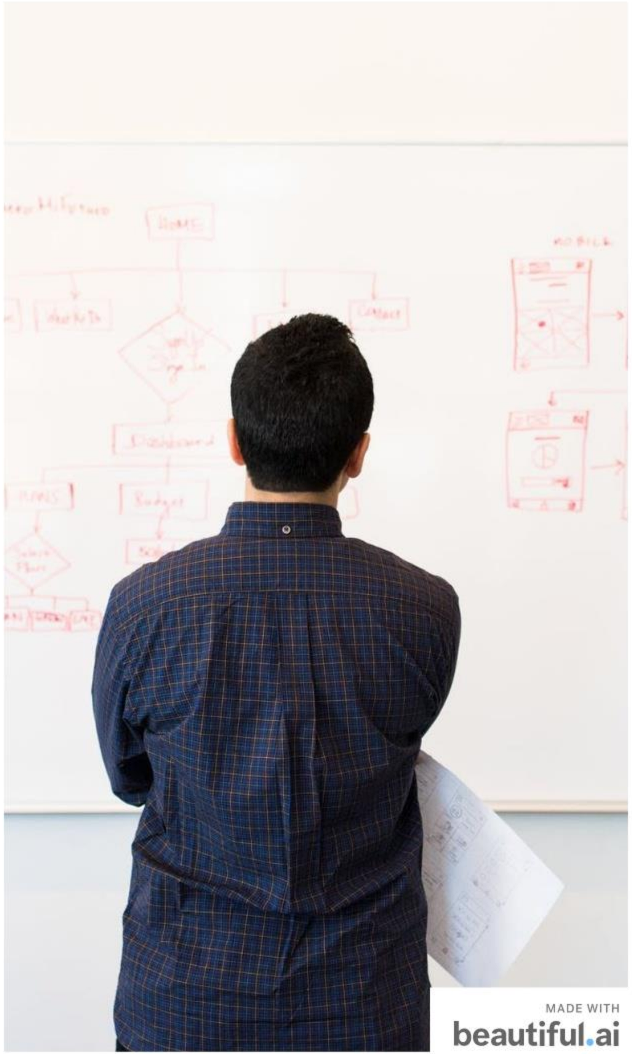
Checking for Outliers

• Across most plots, outliers are present. These are data points that deviate significantly from the general trend. Outliers could represent rare, high-value diamonds or potential data anomalies that require further investigation.

- we used **Winsorization** method to handle outliers by replacing them with the median.

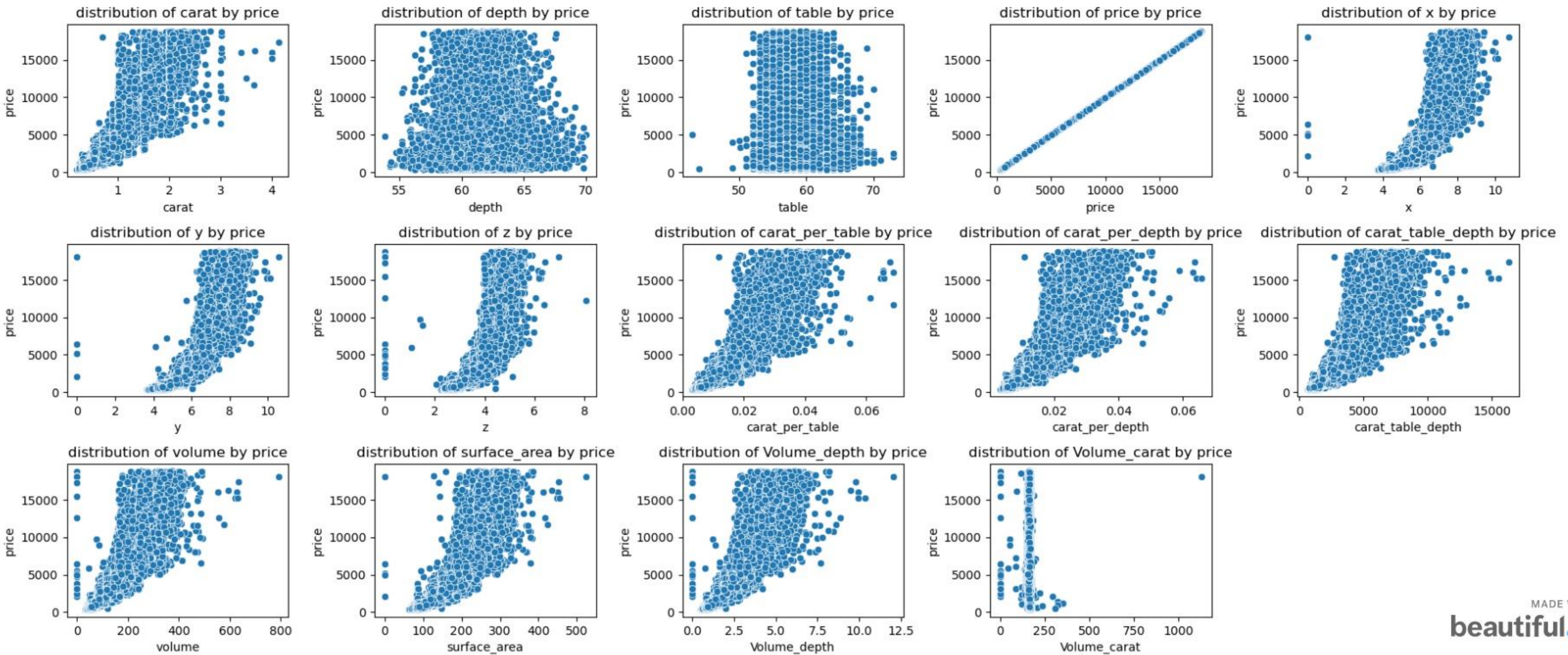


Feature Engineering



Feature Engineering

we added these new features: 'Carat_per_Table', 'Carat_per_Depth', 'Carat_Table_Depth', ' Volume', 'Surface_Area', 'Volume_carat', 'Volume_depth'. we also dropped 'Id' since it's not helpful.



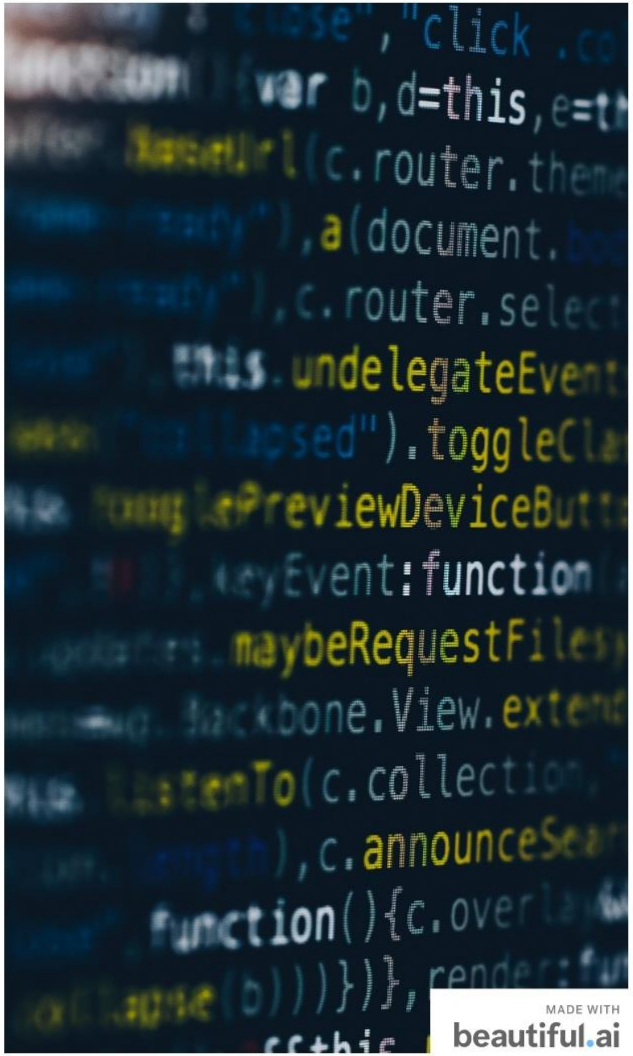
MADE WITH beautiful.ai

Prepare the Data for Machine Learning Algorithms

Prepare the Data for Machine Learning Algorithms

1. Data Splitting.
2. Encoding and Scaling the Data.
3. Grouping non-linear feature for polynomial Feature.
4. Apply a full pipeline.

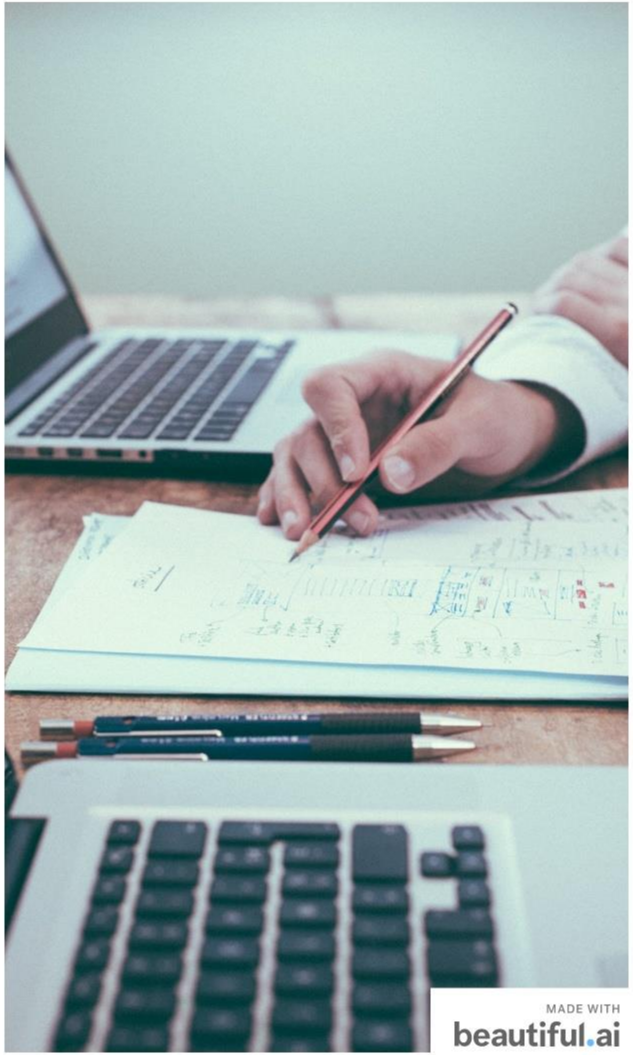
Select a Model and Train it



Select a Model and Train it

	Linear Regression	Random Forest	XGBoost	LGBM
RMSE	1,304	212	252	322

Fine-tune the Model



Fine-tune the Model

	XGBoost
RMSE	526.7

Present Solutions

Present Solutions

Private Score	Public Score
529.78931	522.97531

As a final outcome, we adopted XGBoost as the final model since it provided the lowest RMSE (Root Mean Squared Error). This resulted in a private score of 529.8, which placed us in the 15th position out of 55 competitors.


the
end