# Diamond Price Prediction

This is my first competition to predict model made using python on jupyter notebooks in kaggle with Shai.

**First** I will talk about data.

**Data Description:**

This classic dataset contains the prices and other attributes of almost **54,000** diamonds. It's a great dataset for beginners learning to work with data analysis and visualization.

Content **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), SI2, SI1, 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 top of diamond relative to widest point (43--95)

**Files:**

* **train.csv** - the training set
* **test.csv** - the test set
* **sample\_submission.csv** - a sample submission file in the correct format
* **metaData.csv** - supplemental information about the data

**Columns:**

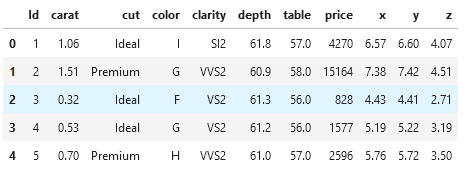
* **example\_id - definition of example\_id**
* **feature\_1 - definition of feature\_1**
* **etc.**
* Get the data

Read data train and data test from kaggle datasets by pandas library and do some basic Data Exploration.

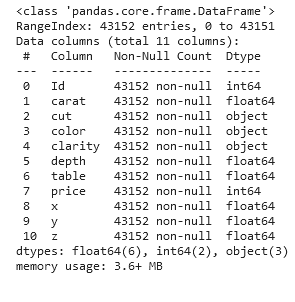
There are four commands which are used for Basic data exploration in Python

* **head()** : This helps to see a few sample rows of the data
* **info()**: This provides the summarized information of the data
* **describe()**: This provides the descriptive statistical details of the data
* **nunique()**: This helps us to identify if a column is categorical or continuous

get head ().



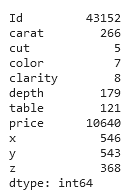
get info ():



get describe ()**:**



Get nunique ():

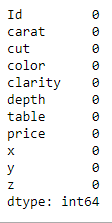


* **price**: Continuous.
* **carat**: Continuous.
* **cut**: Categorical.
* **color**: Categorical.
* **clarity**: Categorical.
* **depth**: Continuous.
* **table**: Continuous.
* **x**: Continuous.
* **y**: Continuous.
* **z**: Continuous.

we can see from our data:

* **Target** Variable: price
* **Features**: id, carat, cut, color, clarity, depth, table, x, y, z.
* **Price**: price of a diamond, and is a **numerical** variable, based on that we can understand that we need to create a **supervised ML Regression model**, as the target variable is **Continuous**.
* **Carat:** represents the weight of the diamond, and is a **numerical** variable.
* **Cut:** represents the quality of the cut of the diamond, and falls into 5 **categories**: fair, good, very good, ideal, and premium. In these categories were represented by an **ordinal** variable, 1-5.
* **Color** represents the color of the diamond, and is rated D through J, with D being the most colorless (and valuable) and J being the most yellow. In these categories were represented by an **ordinal** variable.
* **Clarity:** represents the internal purity of the diamond, and falls into 8 **categories**: I1, SI2, SI1, VS2, VS1, VVS2, VVS1, and IF (in order from least to purest). In these categories were represented by an **ordinal** variable, 1-8.
* **Depth:** represents the depth percentage of the diamond, and is a **numerical** variable.
* **Table:** represents the width of top of diamond relative to widest point, and is a **numerical** variable.
* **X:** represents the length of the diamond, and is a **numerical** variable.
* **Y:** represents the width of the diamond, and is a **numerical** variable.
* **Z:** represents the depth of the diamond, and is a **numerical** variable.

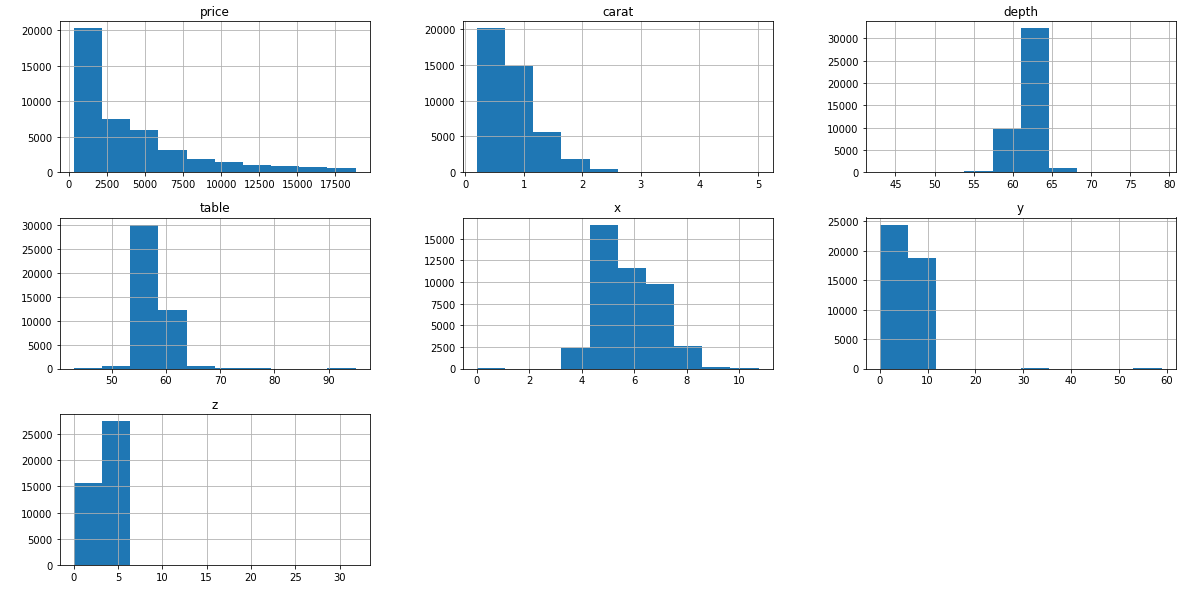
check if any null values in data and any duplicated data



And 0 duplicated row

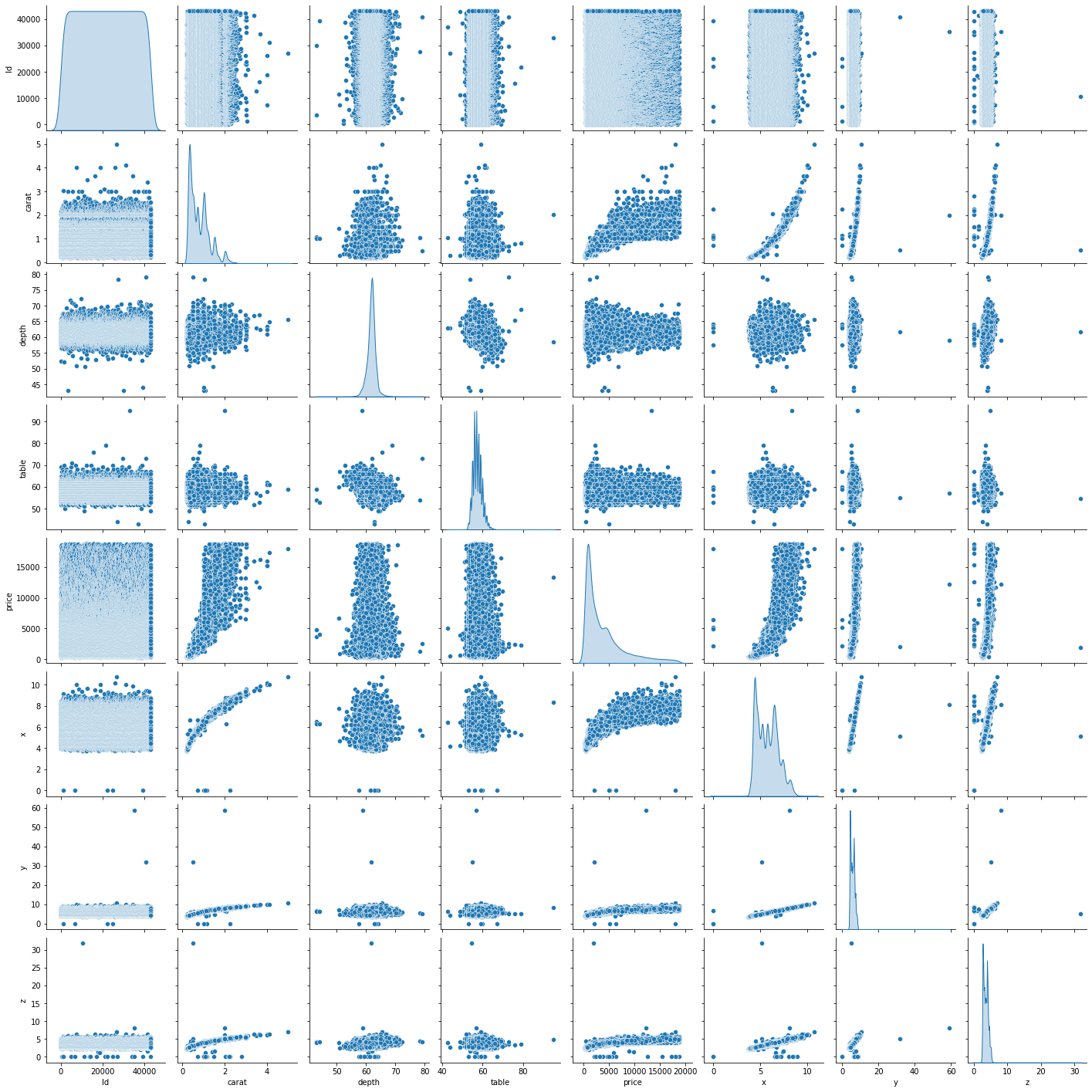
* Visual Exploratory Data Analysis

Visualize distribution of all the Continuous Predictor variables in the data using histograms

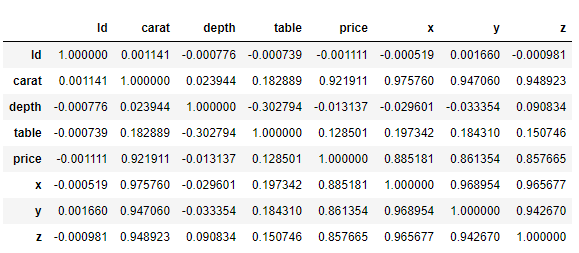


* **carat** : Selected. The distribution is good.
* **table**: Selected. The distribution is good.
* **depth**: Selected. The distribution is good.
* **x**: Selected. Outliers seen near 0, need to treat them.
* **y**: Selected. Outliers seen beyond 20, need to treat them.
* **z**: Selected. Outliers seen beyond 10, need to treat them.

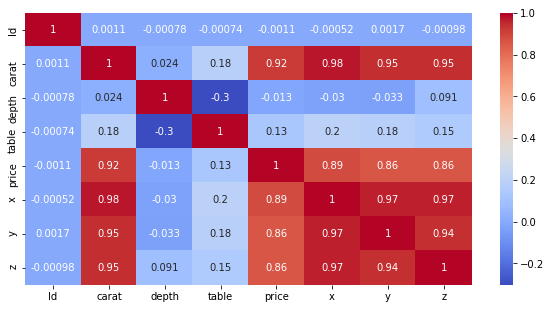
Let’s look at the pair plot of the dataset. Pair plot allows us to see both the distribution of variables and also the relationships between two variables



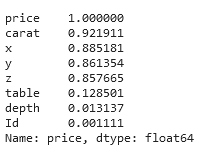
let's quanitfy that correlation by using .corr()



visualize the same using sns.heatmap() method

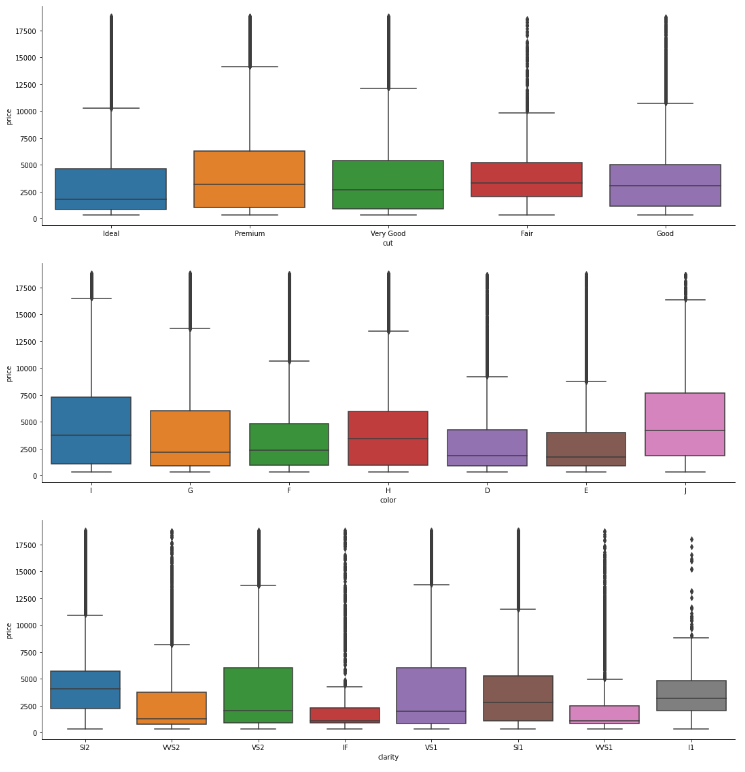


Let’s sort values with absolute correlation with target



We can conclude that carat, x, y & z features have strong correlation with price variable and Id has very weak relation with price variable.

Let's have a sense of **categorical** features with respect to target (price) variable by using box plots



I have used the boxplots to compare the data spread, central tendencies (Median), variability amongst the categories of a given categorical variable and these plots also helps to identify the outliers.

* Data preprocessing

Before begin to building the model, it is good to convert the categorical data to numerical data, so I use OrdinalEncoder to encoder categorical column because the values have ordinal

Cut: worst -> best [ 'Fair', 'Good', 'Very Good','Ideal', 'Premium']

clarity: worst -> best ['I1', 'SI2', 'SI1', 'VS2', 'VS1', 'VVS2', 'VVS1', 'IF']

color: worst -> best [ 'J', 'I', 'H','G', 'F', 'E', 'D']

do fit\_transform on train and transform on test

and create new feature volume by mult x\*y\*z

drop x,y,z

and delete id

and do MinMaxScaler on features

* Train Test Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size=0.2, random\_state=42)

* Machine Learning algorithms

Now is the time to learn regression models for diamond price prediction and start coding machine learning algorithms on this dataset.

Try train multi algorithms and measure error by RMSE

Evaluation Metric  
The evaluation metric for this competition is **Root Mean Squared Error** (RMSE). The RMSE is a commonly used measure of the differences between predicted values provided by a model and the actual observed values.



Notice that Randomforest and XGBRegressor lower error, so I built GridSearchCV for both

Get this result

**XGBRegressor**

RMSE on test = 510

**RandomForestRegressor**

RMSE on test = 620.7198339071548

After I decide the final model train model with best parameter on all data train and test on data test and submission result.

We came in first in competition on Kaggle 💪🏻 💪🏻.

https://www.kaggle.com/competitions/shai-club/leaderboard

