1. Determines the most sold types of diamonds .

----------OR----------

2. Predicting prices of various types of diamond by technical way (ML models)

Ask Questions

1. **On what price of diamond depends ?**
2. **What is the most popular type of diamond?**
3. **Why this results ?**

Obtaining the data

Diamond.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 11:48 AM

Comment Share T

+ Code + Text Connect

```
df = pd.read_csv("/content/train.csv")
df.head()
```

	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

Clean the data

1. Check null values

2. Duplicated values

3. Wrong values

4. Types of data

5. Clear names of columns

6. Irrelevant columns

Null values

```
df.isna().sum()
```

```
Id          0
carat       0
cut         0
color       0
clarity     0
depth       0
table       0
price       0
x           0
y           0
z           0
dtype: int64
```

Types of columns

```
df.info()
```

```
<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
```

Duplicated rows ,,,, names of columns,,,,, drop irrelevant columns...

```
[ ]  df.duplicated().sum()

     0

[ ]  df.columns

     Index(['Id', 'carat', 'cut', 'color', 'clarity', 'depth', 'table', 'price',
            'x', 'y', 'z'],
           dtype='object')

[ ]  df.rename(columns = {"carat":"Weight", "cut":'Quality', "x":"Length_in_mm","y": "Width_in_mm", "z":"Depth_in_mm"}, i
     df_test.rename(columns = {"carat":"Weight", "cut":'Quality', "x":"Length_in_mm","y": "Width_in_mm", "z":"Depth_in_mm

[ ]  df.drop(columns = "Id", inplace = True)
     df_test.drop(columns = "Id", inplace = True, axis = 1)
```

Understand the data

Five summary

```
df.describe()
```

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

1. **Degrees of clarity :**

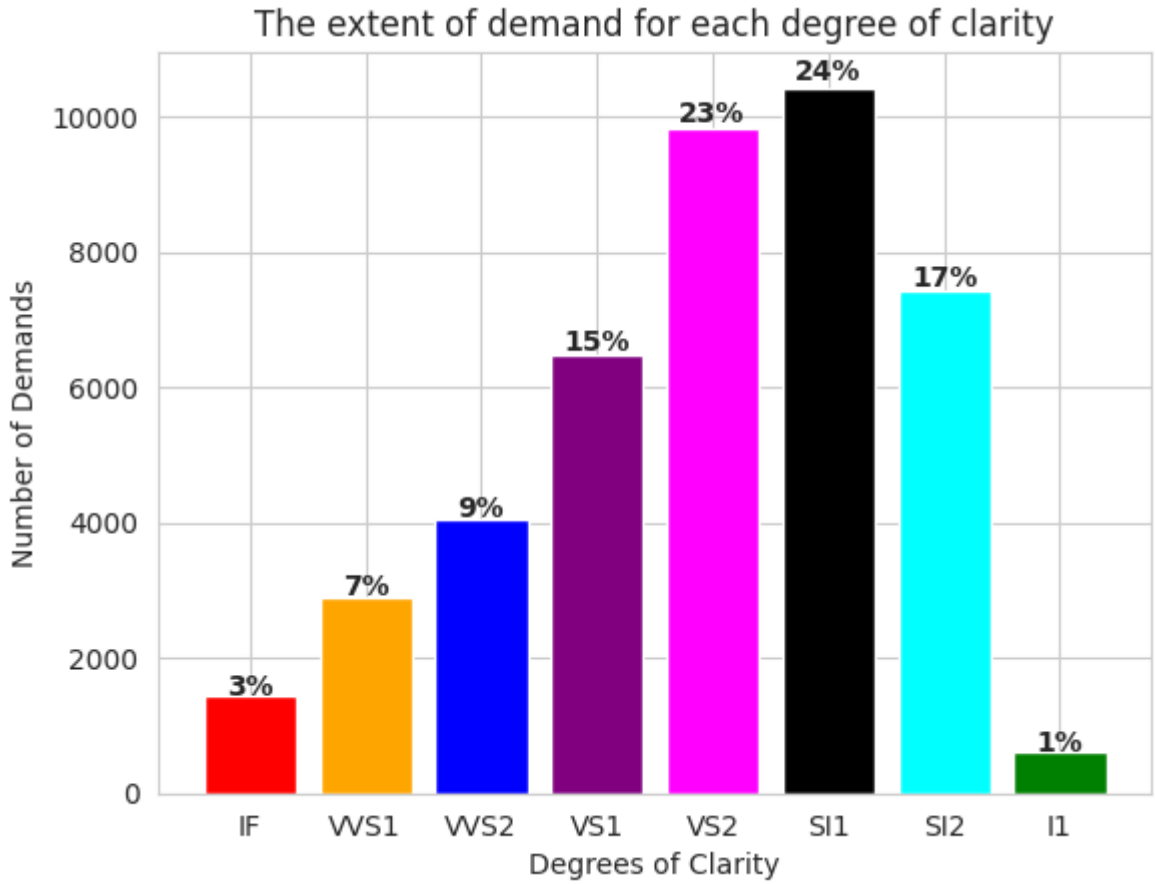
FL > VVS1 > VVS2 > VS1 > VS2 > SI1 > SI2

Why ?

most popular type of clarity (most sold type based on clarity)

Res :

although sI1 & sI2 are less clarity, the demand of them is the most.

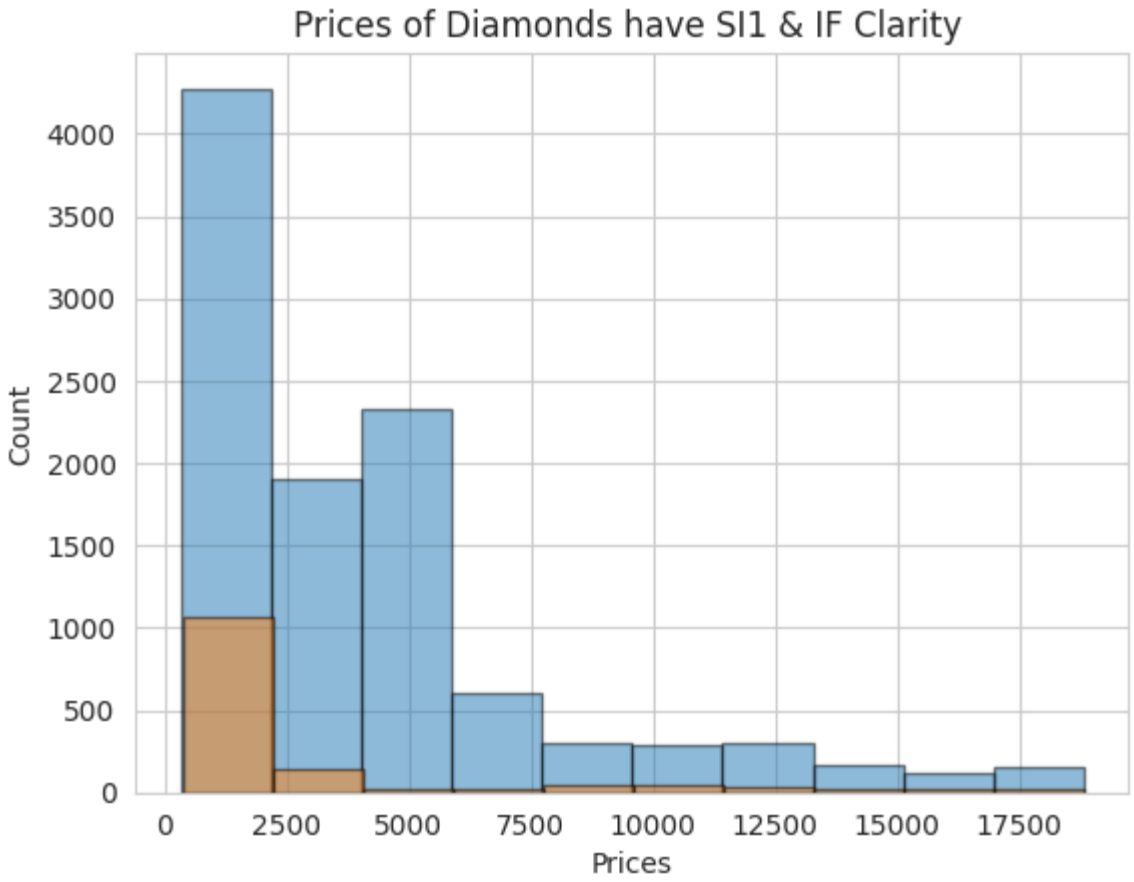


Why ?

answer of the previous question

Res :

both clarity have prices among range(2500 – 17500)

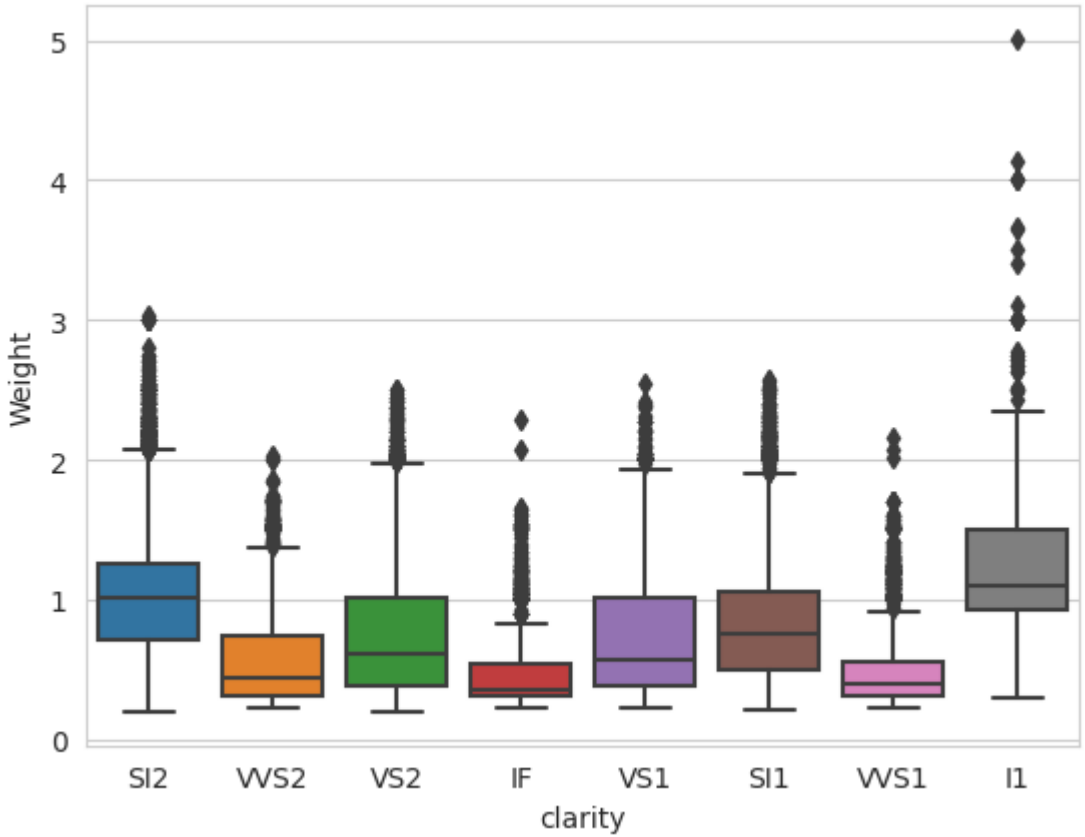


Why ?

To answer the previous question

Res :

The bad clarity sold with high weight unlike good ones

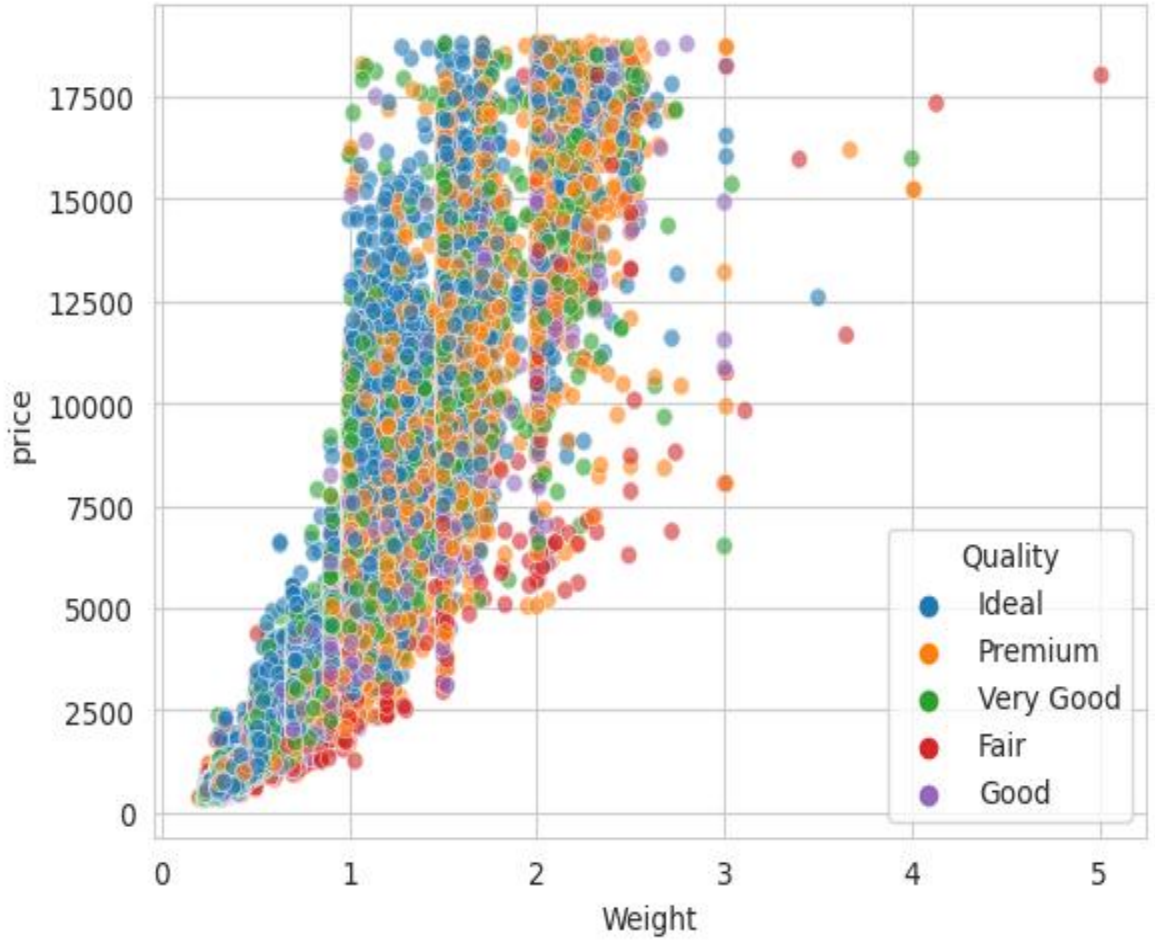


Why ?

more understand for features

Res :

Big purchasing of the best cuts, therefore their prices are high

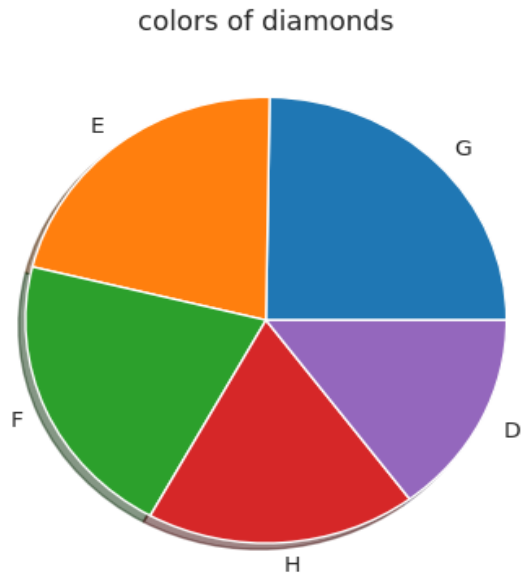


1. Degrees of clarity :

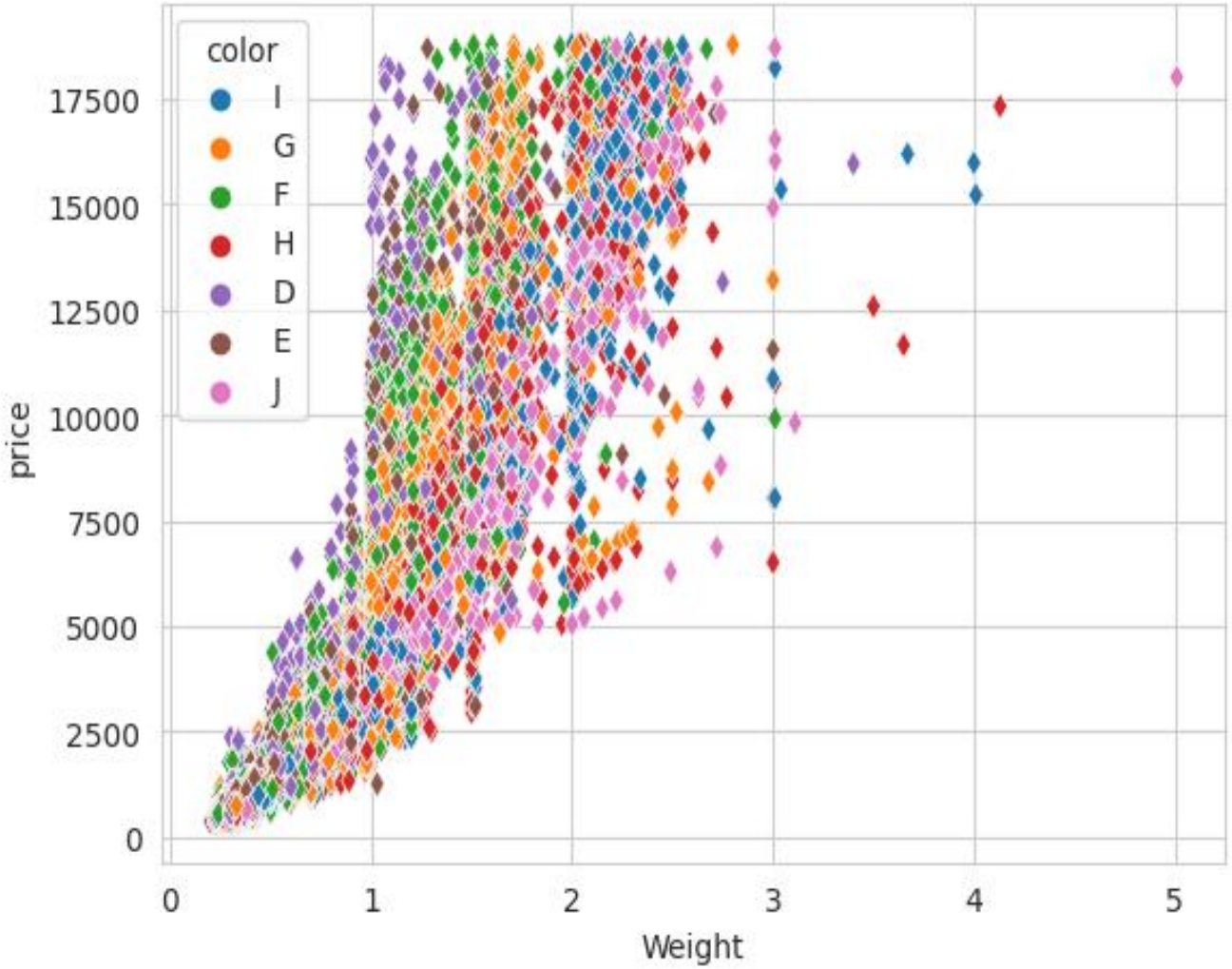
D > E > F > G > H > I > J

Res :

Best colors have high prices & low weight, almostly the all are required.



Weights of different colors of diamond




THE END