Final Project - Diamonds

Yuqing Wang 001443291

Content

1.1. Data Analysis - Ten variables

There are ten variables in the diamond data. In this project's data analysis part, we study mainly about how "color", "cut", "clarity", "carat" affect the price of a diamond.

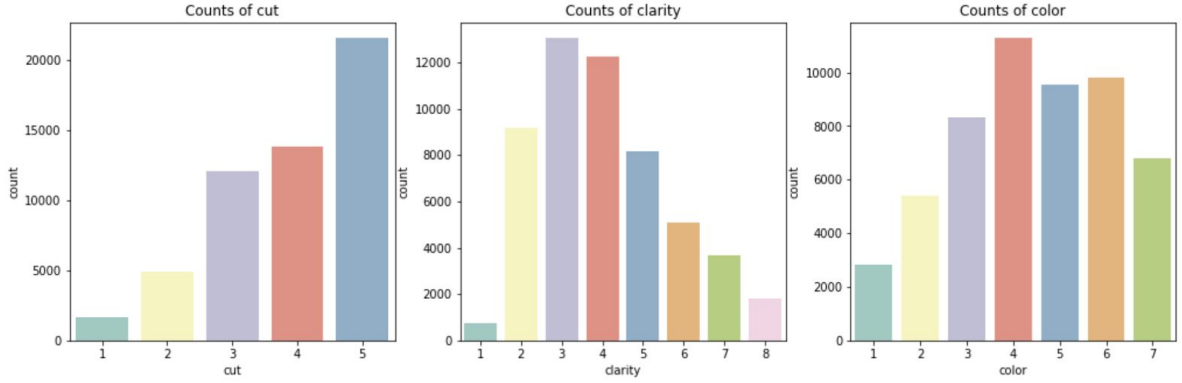
	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

According to the quality of cut, color and clarity, we convert string type data to float:

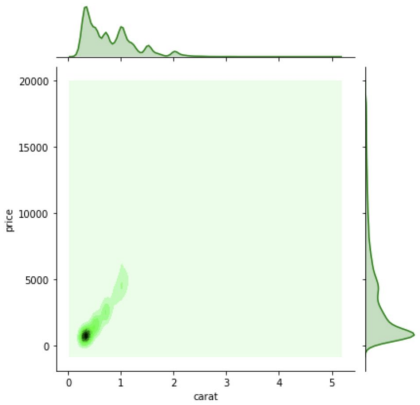
```
cut:  ['Ideal', 'Premium', 'Good', 'Very Good', 'Fair']
clarity:  ['SI2', 'SI1', 'VS1', 'VS2', 'VVS2', 'VVS1', 'I1', 'IF']
color:  ['E', 'I', 'J', 'H', 'F', 'G', 'D']
```

1.2. Data Analysis - Distribution

The quantitative distribution (count distribution) by cuts, claritys and colors:



The count distribution in this dataset of price & carat:



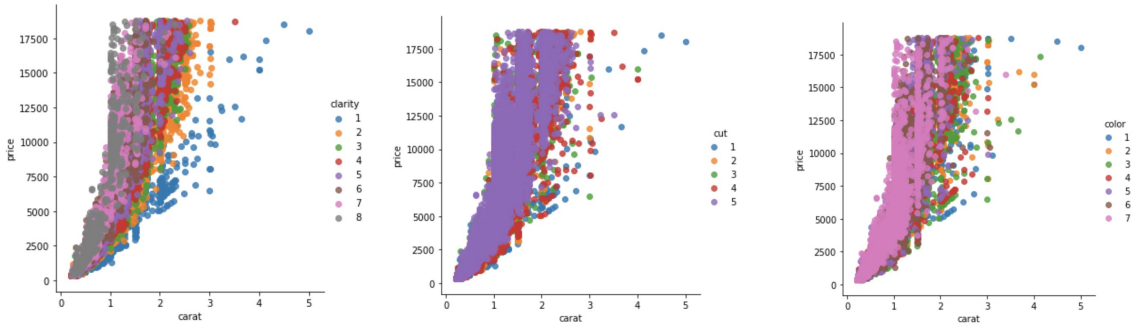
In this data, most diamonds' price are lower than $5000, and weight less than 1 carat.

1.3. Data Analysis - Price per Carat

In order to investigate the effect of variables on diamond value, sometimes we need the “price per carat” rather than “price”. So we add a new column - price_per_carat:

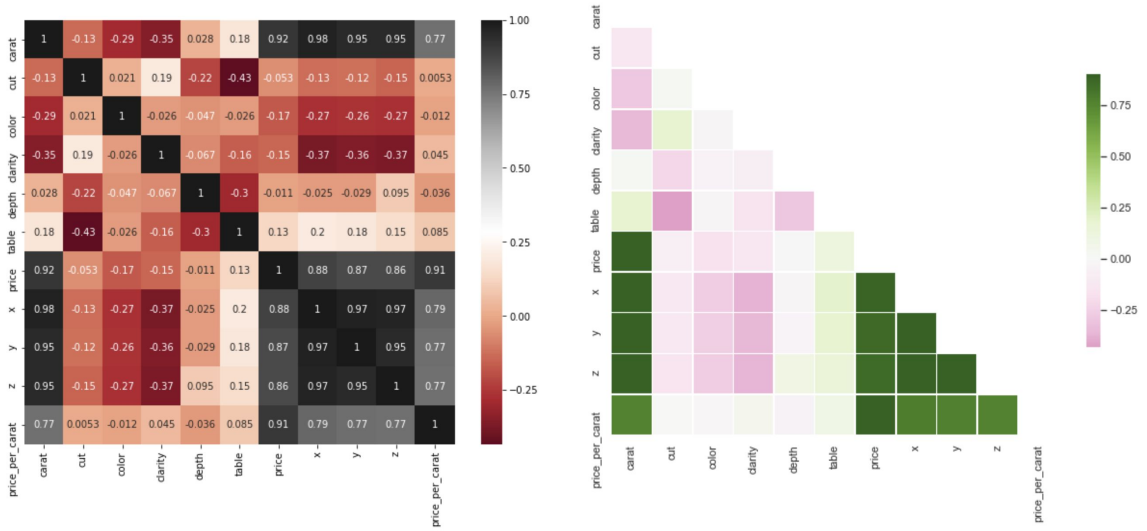
	carat	cut	color	clarity	depth	table	price	x	y	z	price_per_carat
0	0.23	5	6	2	61.5	55.0	326	3.95	3.98	2.43	1417.391304
1	0.21	4	6	3	59.8	61.0	326	3.89	3.84	2.31	1552.380952
2	0.23	2	6	5	56.9	65.0	327	4.05	4.07	2.31	1421.739130
3	0.29	4	2	4	62.4	58.0	334	4.20	4.23	2.63	1151.724138
4	0.31	2	1	2	63.3	58.0	335	4.34	4.35	2.75	1080.645161

As we all know, a big whole diamond is more pricey than a bunch of small diamonds in the same weight, so the slope of carat-price curve is not linear:



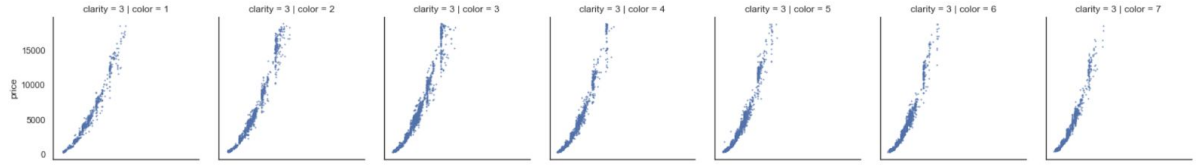
The better the diamond is, the more apparent this phenomenon gets. But a “better” diamond here has four dimensions: clarity, cut, color and weight(size). As we can see in the image: When clarity=8(best), the price is impacted by carat most. When clarity=1(poor clarity), the weight of a diamond has less influence on price.

1.4. Data Analysis - Correlations

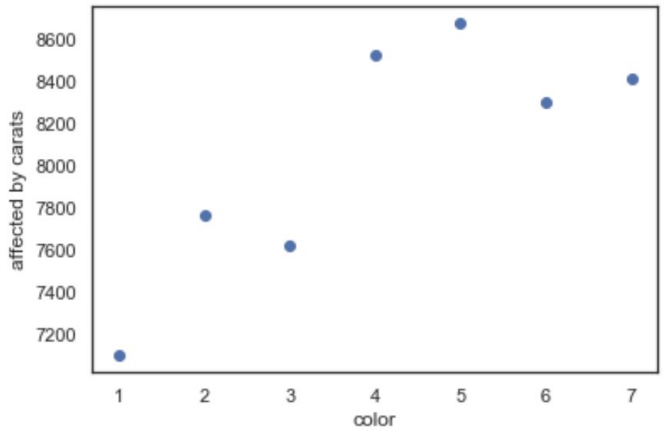


Except for the size of a diamond, the other variables can't affect the price or price_per_carat individually. Next, we need to interpret how exactly these variables affect the price.

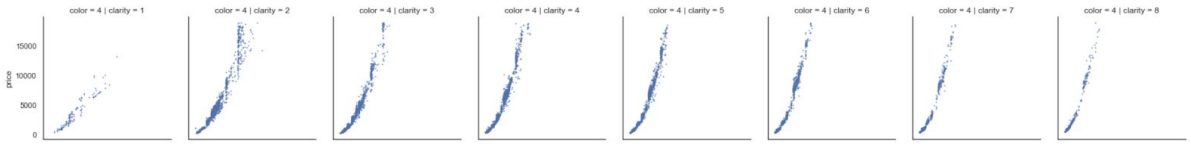
2.1. Interpreting Data - Clarity and Color



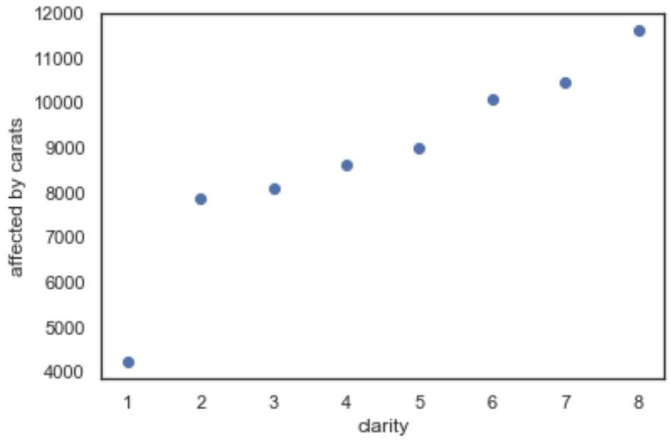
This picture shows that when clarity is the same (clarity = 3), the slope of price-carat is influenced by color. When we use Linear Regression to estimate the slope roughly, the coefficients as shown:



Even though the overall trend is increased, the coefficients are not exactly increased with the increase of the color level.



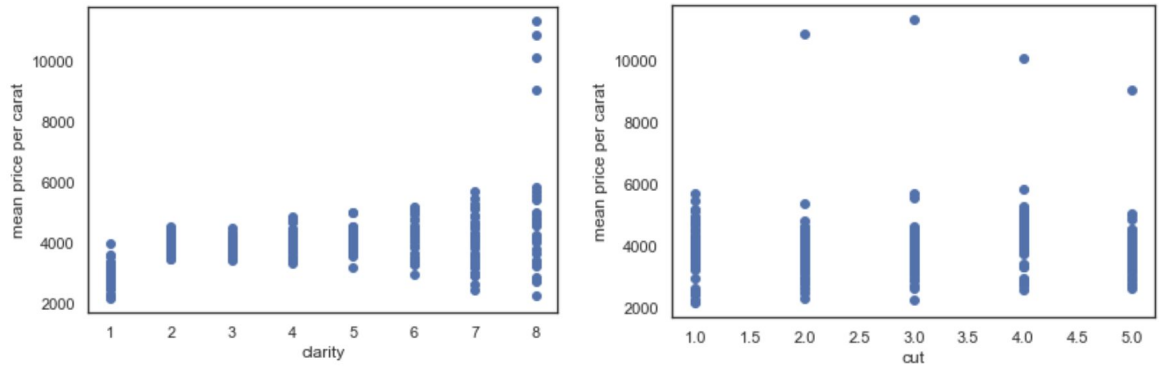
This picture shows that when color is the same (color = 4), the slope of price-carat is influenced by clarity. When we use Linear Regression to estimate the slope roughly, the coefficients as shown:

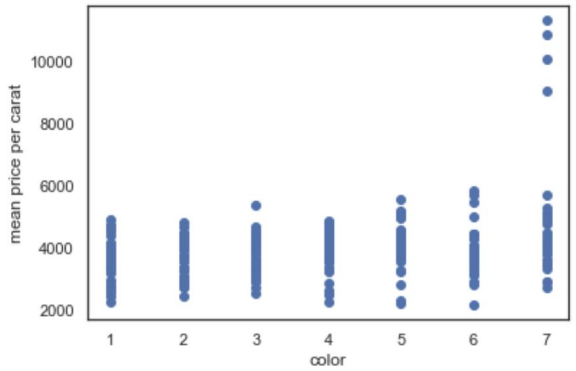


This picture shows that the clarity level drives the value of a diamond more than colors and quality of cut.

2.2. Interpreting Data - Classify data by the combinations

There are 5 levels of cut quality, 8 levels of clarity and 7 levels of colors. So as we classify them by the combination of cut, clarity and color, we have a Dataframe of 280 rows. (5*8*7) Calculate each category's mean price, mean carat and mean price per carat, we can get these data:





Like what we find out before, these variables can't significantly impact the values individually. Which means there's another variable(carat) need to be considered. We also find out there are four outliers in three images, and they are:

	cut	clarity	color	mean_price	mean_price_per_carat	mean_carat
112	2.0	8.0	7.0	10030.333333	10876.804720	0.786667
168	3.0	8.0	7.0	10298.260870	11346.512102	0.803043
224	4.0	8.0	7.0	9056.500000	10099.077901	0.708000
280	5.0	8.0	7.0	6567.178571	9034.176510	0.615714

All of them are top-clarity and top-color, but not necessarily top-cut.

2.3. Interpreting Data - Classify data by single variable

There are 3 variables we want to investigate(cut, clarity and color). When we classify the data by a single variable, we can get 3 Dataframes, each of them represent a certain classification of the data.

Classify data by the quality of cut:
Mean price and mean price per carat didn't increase with the quality of cut. But the mean carat is decreasing with the quality of cut.

	cut	mean price	mean price per carat	mean carat
1	1.0	4358.757764	3767.255681	1.046137
2	2.0	3928.864452	3860.027680	0.849185
3	3.0	3981.759891	4014.128366	0.806381
4	4.0	4584.257704	4222.905374	0.891955
5	5.0	3457.541970	3919.699825	0.702837

Classify data by clarity:
Mean price and mean price per carat didn't increase with the level of clarity. But the mean carat is decreasing with the level of clarity.

	clarity	mean price	mean price per carat	mean carat
1	1.0	3924.168691	2796.296437	1.283846
2	2.0	5063.028606	4010.853865	1.077648
3	3.0	3996.001148	3849.078018	0.850482
4	4.0	3924.989395	4080.526787	0.763935
5	5.0	3839.455391	4155.816808	0.727158
6	6.0	3283.737071	4204.166013	0.596202
7	7.0	2523.114637	3851.410558	0.503321
8	8.0	2864.839106	4259.931736	0.505123

Classify data by color:
Mean price and mean price per carat didn't increase with the level of color. But the mean carat is decreasing with the level of color.

	clarity	mean price	mean price per carat	mean carat
1	1.0	5323.818020	3825.649192	1.162137
2	2.0	5091.874954	3996.402051	1.026927
3	3.0	4486.669196	4008.026941	0.911799
4	4.0	3999.135671	4163.411524	0.771190
5	5.0	3724.886397	4134.730684	0.736538
6	6.0	3076.752475	3804.611475	0.657867
7	7.0	3169.954096	3952.564280	0.657795

2.4. Interpreting Data - Clarity vs Cut

Mean Price:
Column [clarity=2] has the highest mean price. (Reason: the weight/size of diamonds in this column is high.)

clarity	1	2	3	4	5	6	7	8
cut								
1	3703.53	5173.92	4208.28	4174.72	4165.14	3349.77	3871.35	1912.33
2	3596.64	4580.26	3689.53	4262.24	3801.45	3079.11	2254.77	4098.32
3	4078.23	4988.69	3932.39	4215.76	3805.35	3037.77	2459.44	4396.22
4	3947.33	5545.94	4455.27	4550.33	4485.46	3795.12	2831.21	3856.14
5	4335.73	4755.95	3752.12	3284.55	3489.74	3250.29	2468.13	2272.91

Mean Price per Carat:
The better the clarity is, the valuable the diamond is.

clarity	1	2	3	4	5	6	7	8
cut								
1	2408.68	3849.52	3881.12	4125.85	4097.22	4225.64	4804.24	3941.26
2	2732.78	3790.38	3647.98	4195.3	4055.9	3975.91	3538.73	5072.77
3	2948.74	4025.61	3823.95	4204.89	4104.37	3939.28	3695.18	5399.24
4	2810.76	4162.31	4043.92	4350.01	4444.21	4486.35	4008.32	4849.73
5	3287.59	3947.69	3775.63	3814.12	4042.25	4259.98	3884.91	3850.83

Mean Carat:
Poor clarity diamonds usually have a bigger size.

clarity	1	2	3	4	5	6	7	8
cut								
1	1.361	1.20384	0.964632	0.885249	0.879824	0.691594	0.664706	0.474444
2	1.20302	1.03523	0.830397	0.850787	0.757685	0.61493	0.502312	0.616338
3	1.2819	1.06434	0.845978	0.811181	0.733307	0.566389	0.494588	0.618769
4	1.28702	1.14416	0.908601	0.833774	0.793308	0.654724	0.534821	0.603478
5	1.22267	1.00793	0.801808	0.670566	0.674714	0.586213	0.49596	0.455041

Count Diamonds:
In the clarity range of (1,4], the better the clarity the diamond has, the better cut it gets.

clarity	1	2	3	4	5	6	7	8
cut								
1	210	466	408	261	170	69	17	9
2	96	1081	1560	978	648	286	186	71
3	84	2100	3240	2591	1775	1235	789	268
4	205	2949	3575	3357	1989	870	616	230
5	146	2598	4282	5071	3589	2606	2047	1212

2.5. Interpreting Data - Clarity vs Color

Mean Price:
Column [clarity=2] has the highest mean price. (Reason: the weight/size of diamonds in this column is high.)

clarity	1	2	3	4	5	6	7	8
color								
1	5254.06	6520.96	5186.05	5311.06	4884.46	5142.4	4034.18	3363.88
2	4302.18	7002.65	5355.02	5690.51	4633.18	2968.23	2034.86	1994.94
3	4453.41	6099.9	5032.41	4722.41	3780.69	2649.07	1845.66	2287.87
4	3545.69	5021.68	3774.79	4416.26	4131.36	3845.28	2866.82	2558.03
5	3342.18	4472.63	3714.23	3756.8	3796.72	3475.51	2804.28	2750.84
6	3488.42	4173.83	3161.84	2750.94	2856.29	2499.67	2219.82	3668.51
7	3863.02	3931.1	2976.15	2587.23	3030.16	3351.13	2947.91	8307.37

Mean Price per Carat:

Mean price per carat is the most important indicator of the value of diamonds.

clarity	1	2	3	4	5	6	7	8
color								
1	2710.08	4091.03	3775.64	3860.45	3750.35	4064.89	3569.89	3220.23
2	2742.59	4408.04	4075.54	4316.25	3998.21	3334.66	2964.64	3013.21
3	2904.85	4319.84	4198.8	4240.2	3898.53	3401.57	3161.01	3566.94
4	2617.48	3971	3737.55	4405.77	4420.88	4515.09	4011.67	4007.01
5	2768.38	3910.3	3875.62	4198.9	4419.34	4552.22	4243.09	4375.38
6	2906.26	3829.41	3654.5	3734.33	3928.46	3935.71	3918.26	5220.98
7	3064.48	3755.89	3644.01	3758.9	4204.89	4749.58	4835.82	9937.42

In this image, clarity=2&color=1, clarity=2&color=2, clarity=2&color=3 have high indicators. This is because the weight of diamonds influence the value:

```
color=1&clarity=2 but price per carat is high: the mean carat of this kind of diamonds is 2.4414285714285713
color=2&clarity=2 but price per carat is high: the mean carat of this kind of diamonds is 2.2021428571428565
color=3&clarity=2 but price per carat is high: the mean carat of this kind of diamonds is 2.1063559322033902
```

In order to illustrate this, we extract all data of carat=1.0 to eliminate the disturbance of carats. And we got a new mean price per carat chart:

clarity	1	2	3	4	5	6	7	8
color								
1	1732.5	3200.58	3630.83	3724.42	3814.78	3564.5	4675	nan
2	1987.25	3495.05	3961.97	4267.53	4457.41	4032	4445	nan
3	2665.57	3797.34	4502.85	4840.91	5099.11	5445.8	6233.6	7235
4	2591.71	3835.44	4570.8	5785.85	6232.77	7318.35	7801.8	7832
5	2780.83	4075.12	4857.32	6133.95	6733.77	8454.83	9238.22	9833.4
6	3366.29	4189.79	5188.92	6479.21	7369.09	8663.39	9010	11084
7	2657	4324.8	5454.81	6639.54	7694.06	9712	12000.7	16156.7

At this time, in the same weight, the better the clarity&color the diamond has, the more valuable the diamond is.

Mean Carat:
Poor clarity diamonds usually have a bigger size.

clarity / color	1	2	3	4	5	6	7	8
1	1.7506	1.42426	1.17283	1.13465	1.01744	1.02847	0.843243	0.703922
2	1.43924	1.39505	1.07687	1.06302	0.903077	0.678411	0.55493	0.515944
3	1.43833	1.24109	0.990945	0.898704	0.754209	0.582089	0.480496	0.505385
4	1.2226	1.06747	0.81961	0.797946	0.728254	0.655107	0.536116	0.491821
5	1.08559	0.9867	0.800849	0.696311	0.681723	0.589877	0.495327	0.460909
6	1.10618	0.921576	0.711303	0.592231	0.573411	0.475621	0.425808	0.506266
7	1.11714	0.872168	0.668401	0.558321	0.583021	0.52859	0.478849	0.698767

Count Diamonds:

clarity / color	1	2	3	4	5	6	7	8
1	50	479	750	731	542	131	74	51
2	92	912	1424	1169	962	365	355	143
3	162	1563	2275	1643	1169	608	585	299
4	150	1548	1976	2347	2148	1443	999	681
5	143	1609	2131	2201	1364	975	734	385
6	102	1713	2426	2470	1281	991	656	158
7	42	1370	2083	1697	705	553	252	73

2.6. Interpreting Data - Cut vs Color

Mean Price:
When the quality of cut is 5 (top-quality), the size of a diamond will be accordingly small, because it will sacrifice the weight of a diamond to get a perfect shape. Meanwhile, the weight of a diamond is crucial to the mean price.

cut	1	2	3	4	5
color					
1	4975.66	4574.17	5103.51	6294.59	4918.19
2	4685.45	5078.53	5255.88	5946.18	4451.97
3	5135.68	4276.25	4535.39	5216.71	3889.33
4	4239.25	4123.48	3872.75	4500.74	3720.71
5	3827	3495.75	3778.82	4324.89	3374.94
6	3682.31	3423.64	3214.65	3538.91	2597.55
7	4291.06	3405.38	3470.47	3631.29	2629.09

Mean Price per Carat:

cut	1	2	3	4	5
color					
1	3345.94	3524.27	3792.5	4140.52	3733.77
2	3514.65	3907.22	4111.27	4267.22	3808.07
3	3831.51	3825.52	4034.24	4278.49	3846.07
4	3699.34	4087.37	4054.56	4320.51	4164
5	3788.02	3820.56	4142.19	4357.8	4097.52
6	3820.46	3806.25	3821.53	3987.88	3683.17
7	4244.56	3846	4072.81	4111.56	3806.53

Mean Carat:

Poor quality of cut can keep a high weight of a diamond.

cut	1	2	3	4	5
color					
1	1.34118	1.09954	1.13322	1.29309	1.06359
2	1.19806	1.05722	1.04695	1.14494	0.913029
3	1.21917	0.914729	0.915948	1.01645	0.799525
4	1.02382	0.850896	0.766799	0.841488	0.700715
5	0.904712	0.77593	0.740961	0.827036	0.655829
6	0.856607	0.745134	0.676317	0.717745	0.578401
7	0.920123	0.744517	0.696424	0.721547	0.565766

Count Diamonds:

Most diamonds in this data have ideal cut shapes.

cut	1	2	3	4	5
color					
1	119	307	678	808	896
2	175	522	1204	1428	2093
3	303	702	1824	2360	3115
4	314	871	2299	2924	4884
5	312	909	2164	2331	3826
6	224	933	2400	2337	3903
7	163	662	1513	1603	2834

3. Prediction

In Supervised Learning, when the dependent variable (here is price) is a continuous numerical variable, we use regression analysis. If the dependent variable is a categorical variable, we should use classification analysis. So here, I chose four regression model to predict the price of diamonds. And then use cross-validation to score the models.

3.1. Prediction - Polynomial Regression

```
# Polynomial Regression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline
def PolynomialRegression(degree=1, **kwargs):
    return make_pipeline(PolynomialFeatures(degree),
                         LinearRegression(**kwargs))
model = PolynomialRegression(1)
model.fit(Xtrain, ytrain)
y_model = model.predict(Xtest)
from sklearn.metrics import r2_score
r2_score(ytest, y_model)
```

```
0.9794356636239624
```

3.2. Prediction - Linear Regression

```
# Linear Regression
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import KFold

lr = LinearRegression()
kf = KFold(n_splits=5, shuffle=True, random_state=11)
scores = cross_val_score(lr, Xtrain, ytrain, scoring='r2', cv=kf, n_jobs=5)
print('linear regression:')
print(scores)
print('Average R-Squared Score:', np.mean(scores),'\n')
```

```
linear regression:
[0.98071809 0.97688003 0.98012449 0.97036095 0.9754577 ]
Average R-Squared Score: 0.976708252352482
```

3.3. Prediction - Ridge Regression

```
# Ridge Regression
from sklearn.linear_model import Ridge
from sklearn.model_selection import KFold
rg = Ridge()
kf = KFold(n_splits=5, shuffle=True, random_state=11)
scores = cross_val_score(rg, Xtrain, ytrain, scoring='r2', cv=kf, n_jobs=5)
print('ridge regression:')
print(scores)
print('Average R-Squared Score:', np.mean(scores),'\n')
```

```
ridge regression:
[0.98067411 0.97684154 0.98008439 0.97044294 0.9754912 ]
Average R-Squared Score: 0.97670683631277
```

3.4. Prediction - Random Forest Regression

```
# RandomForestRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import KFold
rf = RandomForestRegressor()
kf = KFold(n_splits=5, shuffle=True, random_state=11)
scores = cross_val_score(rf, Xtrain, ytrain, scoring='r2', cv=kf, n_jobs=5)
print('random forest regression:')
print(scores)
print('Average R-Squared Score:', np.mean(scores),'\n')
```

```
random forest regression:
[0.99959508 0.99961065 0.9997544  0.99941997 0.99958551]
Average R-Squared Score: 0.9995931236675709
```