

```
In [1]: import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

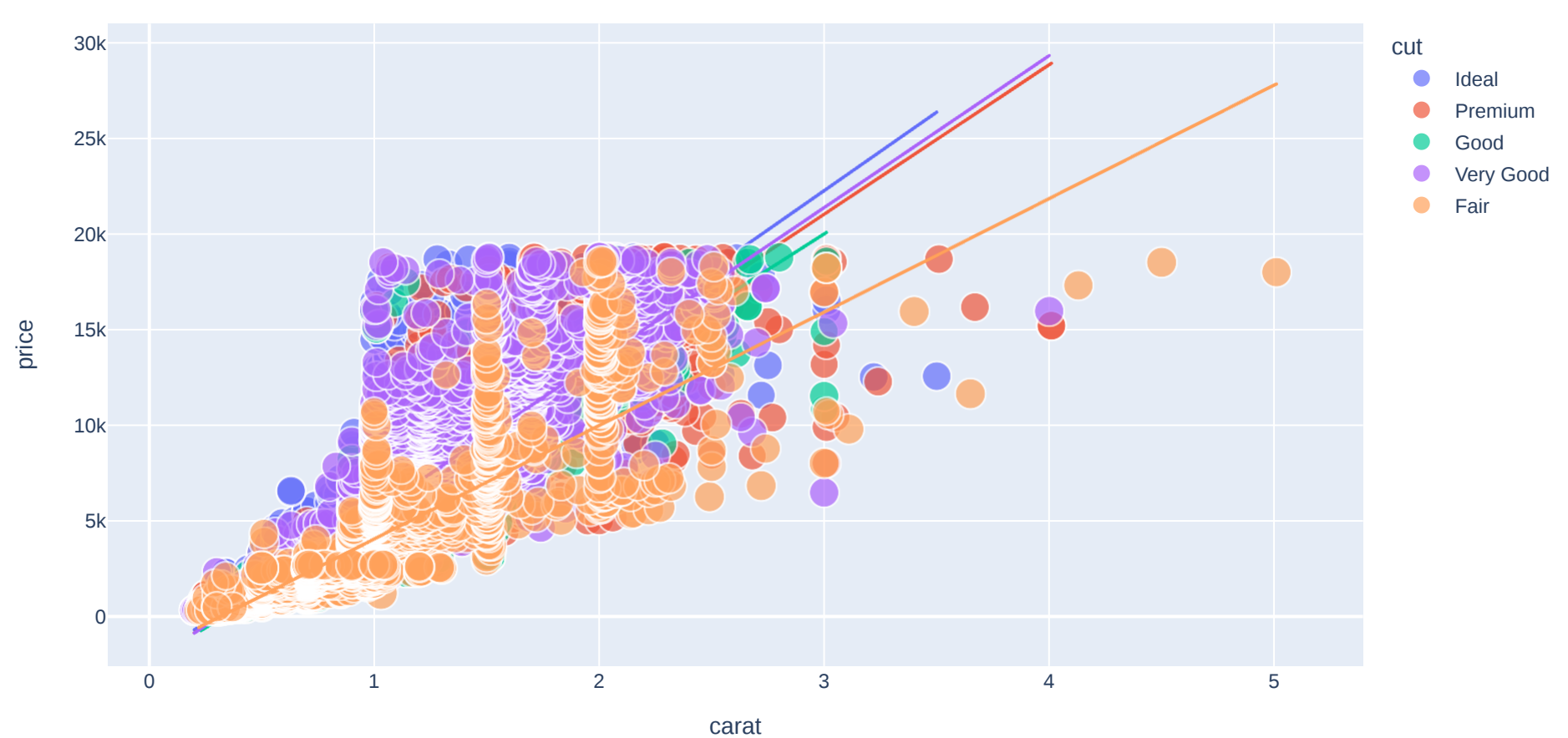
data = pd.read_csv(r"C:\Users\User21\Diamonds.csv")
print(data.head())

   Unnamed: 0  carat      cut color clarity depth  table  price     x     y \
0           1    0.23   Ideal     E   SI2    61.5   55.0   326   3.95   3.98
1           2    0.21  Premium     E  SI1    59.8   61.0   326   3.89   3.84
2           3    0.23    Good     E  VS1    56.9   65.0   327   4.05   4.07
3           4    0.29  Premium     I  VS2    62.4   58.0   334   4.20   4.23
4           5    0.31    Good     J  SI2    63.3   58.0   335   4.34   4.35

      z
0    2.43
1    2.31
2    2.31
3    2.63
4    2.75

In [2]: figure = px.scatter(data_frame = data, x="carat",
                             y="price", size="depth",
                             color= "cut", trendline="ols")

figure.show()
```



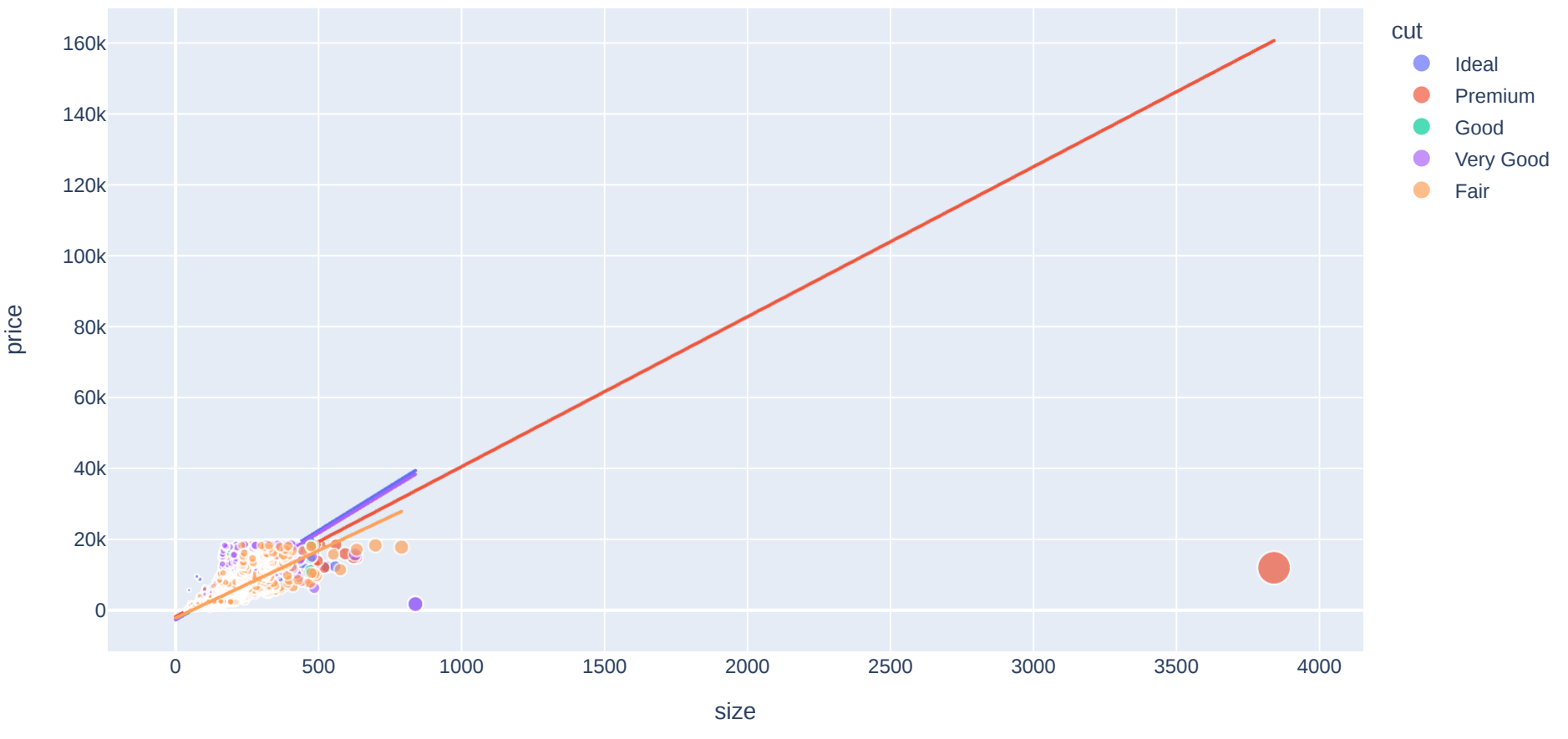
```
In [3]: data["size"] = data["x"] * data["y"] * data["z"]
print(data)

   Unnamed: 0  carat      cut color clarity depth  table  price     x     y \
0           1    0.23   Ideal     E   SI2    61.5   55.0   326   3.95   3.95
1           2    0.21  Premium     E  SI1    59.8   61.0   326   3.89   3.84
2           3    0.23    Good     E  VS1    56.9   65.0   327   4.05   4.05
3           4    0.29  Premium     I  VS2    62.4   58.0   334   4.20   4.20
4           5    0.31    Good     J  SI2    63.3   58.0   335   4.34   4.34
...         ...    ...     ...   ...   ...   ...   ...   ...   ...   ...
53935       53936  0.72   Ideal     D   SI1    60.8   57.0   2757  5.75   5.75
53936       53937  0.72    Good     D   SI1    63.1   55.0   2757  5.69   5.69
53937       53938  0.70  Very Good     D   SI1    62.8   60.0   2757  5.66   5.66
53938       53939  0.86  Premium     H   SI2    61.0   58.0   2757  6.15   6.15
53939       53940  0.75   Ideal     D   SI2    62.2   55.0   2757  5.83   5.83
...         ...    ...     ...   ...   ...   ...   ...   ...   ...   ...
53935       5.76   3.50  115.920000
53936       5.75   3.61  118.110175
53937       5.68   3.56  114.449728
53938       6.12   3.74  140.766120
53939       5.87   3.64  124.568444

[53940 rows x 12 columns]
```

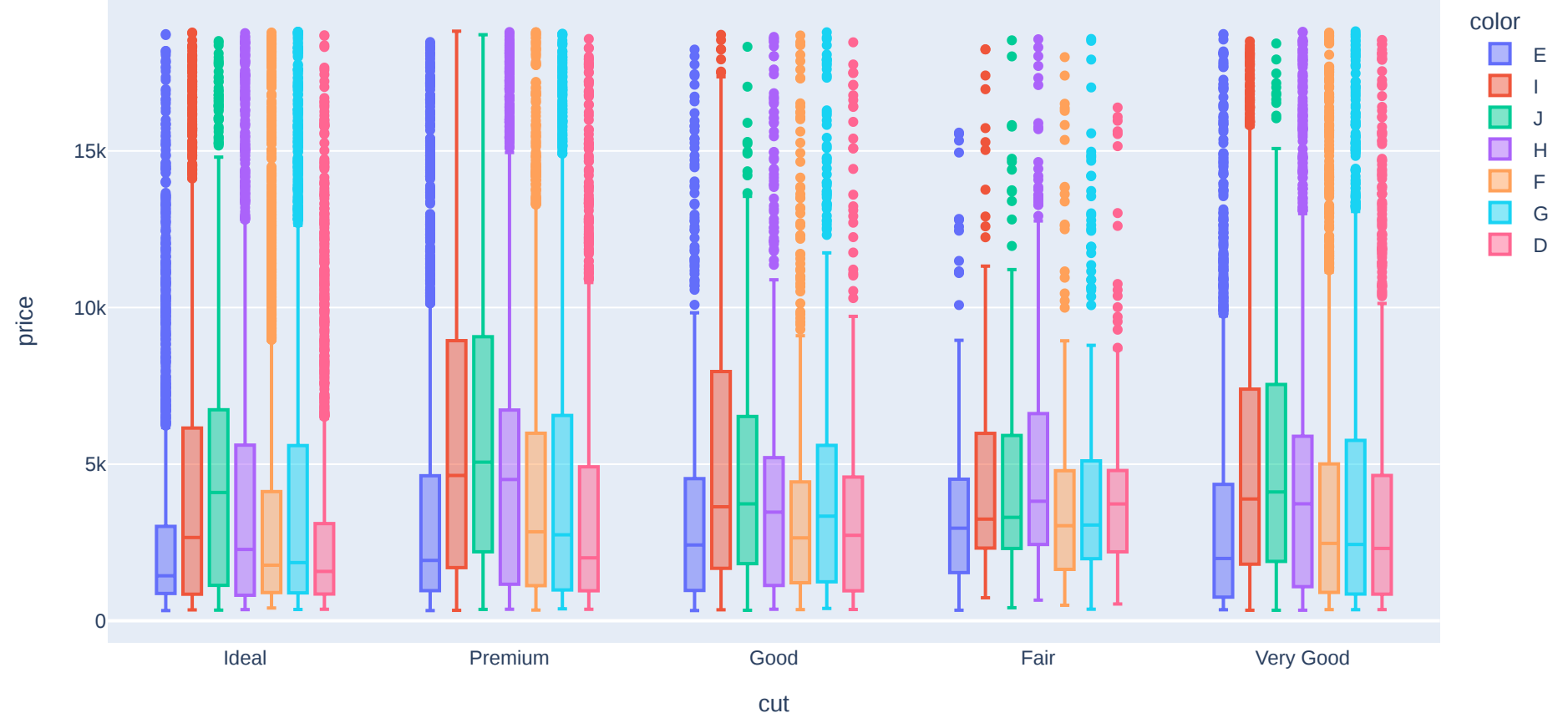
```
In [4]: figure = px.scatter(data_frame = data, x="size",
                             y="price", size="size",
                             color= "cut", trendline="ols")

figure.show()
```



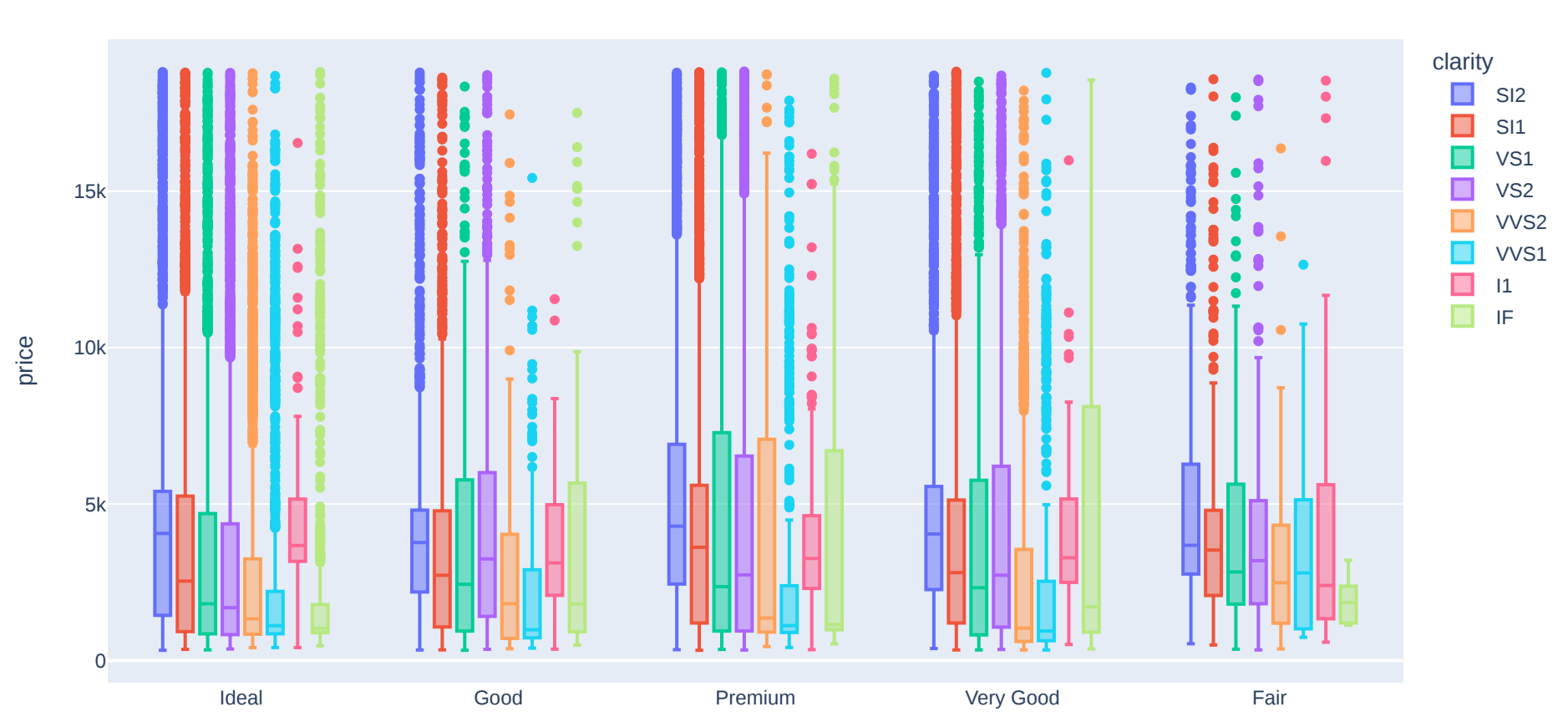
```
In [5]: fig = px.box(data, x="cut",
                    y="price",
                    color="color")

fig.show()
```



```
In [6]: fig = px.box(data,
                    x="cut",
                    y="price",
                    color="clarity")

fig.show()
```



```
In [7]: correlation = data.corr()
print(correlation["price"].sort_values(ascending=False))
```

```
price      1.000000
carat      0.921591
size       0.902385
x          0.884435
y          0.865421
z          0.861249
table      0.127134
depth     -0.010647
Unnamed: 0 -0.306873
Name: price, dtype: float64
```

```
In [8]: data["cut"] = data["cut"].map({"Ideal": 1,
                                     "Premium": 2,
                                     "Good": 3,
                                     "Very Good": 4,
                                     "Fair": 5})
```

```
In [9]: #splitting data
from sklearn.model_selection import train_test_split
x = np.array(data[["carat", "cut", "size"]])
y = np.array(data[["price"]])

xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test_size=0.10,
                                                random_state=42)
```

```
In [10]: from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(xtrain, ytrain)
```

C:\Users\User21\AppData\Local\Temp\ipykernel_6932\2944638855.py:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
Out[10]: RandomForestRegressor()
```

```
In [11]: print("Diamond Price Prediction")
a = float(input("Carat Size: "))
b = int(input("Cut Type (Ideal: 1, Premium: 2, Good: 3, Very Good: 4, Fair: 5): "))
c = float(input("Size: "))
features = np.array([[a, b, c]])
print("Predicted Diamond's Price = ", model.predict(features))

Diamond Price Prediction
Carat Size: 0.60
Cut Type (Ideal: 1, Premium: 2, Good: 3, Very Good: 4, Fair: 5): 2
Size: 40
Predicted Diamond's Price = [905.81666667]
```

```
In [ ]:
In [ ]:
```