Muhammad Nasir Khurshid, 04071913019

Gemstone Price Predictor

Problem

Cubic Zirconia is a gemstone which has different features. On the basis of these features, cubic zirconia is sold at different prices generating profit depending on the prices. We want to predict the prices of gemstones on the basis of features so that we can distinguish between higher profitable stones and lower profitable stones so as to have a better profit share.

Data Set

Features

1. Carat - Carat weight of the cubic zirconia
2. Cut - Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good
3. Color - Color of the cubic zirconia with D being the best and J the worst
4. Clarity - Cubic zirconia Clarity refers to the absence of the Inclusions and Blemishes
5. Depth - The Height of a cubic zirconia, measured from the Culet to the table, divided by its average Girdle
6. Table - The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter
7. X - Length of the cubic zirconia in mm
8. Y - Width of the cubic zirconia in mm
9. Z - Height of the cubic zirconia in mm

Ground Truth

- Price - The Price of the cubic zirconia

Approach

- Linear Regression & Gradient Descent (to get minimum loss)

Data Processing

- The values for ‘Cut’ and ‘Color’ are converted from string to integer to train the model as we can’t perform calculations on strings.
- ‘Clarity’ is dropped from the dataset.
- All the rows with ‘NULL’ values are deleted.
- Data normalization is performed on features and label.

Results

- Feature: Carat
- No. of features: 1

Iterations	Learning Rate	W	Loss
100	0.5	0.92213359	0.14966393890862925
100	0.1	0.92213359	0.1496639389086296
100	0.01	0.79983114	0.1652379666741372
1000	0.001	0.79757656	0.16524004301365094

- Features: Carat, Color, Cut
- No. of features: 3

Iterations	Learning Rate	W	Loss
100	0.5	0.95692962 0.09814202 0.07098246	0.1357514311226255
100	0.1	0.95692952 0.09814195 0.07098242	0.13575143112264043
100	0.01	0.79825459 0.01541505 0.02470005	0.1652379666741372
100	0.001	0.16661795 0.02155038 0.00870329	0.7183541539480602

- Features: Carat, Color, Cut, Depth, Table
- No. of features: 5

Iterations	Learning Rate	W	Loss
100	0.5	0.96085669,0.09752813, 0.05064569,0.03063951, 0.03227716	0.1348142549157066
100	0.1	0.96085174,0.09752807, 0.05068108,0.03060971, 0.03223804	0.1348142564020076
100	0.01	0.79541529,0.01479002, 0.03256153,0.01275281, 0.01406768	0.16273014176623032
100	0.001	0.16626825,0.02151172, 0.00783853.0.00072655, 0.01961347	0.7156895760027296

- Features: Carat, Color, Cut, Depth, X, Y Z
- No. of features: 7

Iterations	Learning Rate	W	Loss
100	0.5	1.05370694e+47,3.35698914e+46, 2.03541884e+46,5.30759922e+45, 1.06094716e+47,1.04240174e+47, 1.06079607e+47	2.048077891295911e+94
100	0.1	0.78200635, 0.09022727, 0.06575634, 0.01608713, 0.10031636, 0.00295004, 0.06184821	0.14240287733075724
100	0.01	0.30734046, 0.05567691, 0.04965099, 0.01346959, 0.22099967, 0.1882748, 0.2103628	0.17809895519569366
100	0.001	0.13038446, 0.01343434, 0.0038394, 0.00188877, 0.12269669, 0.11749237. 0.12091656	0.36666310913884687

Conclusion

Using gradient descent, we are able to minimize the loss with 1, 3, 5 and 7 features and 4 different learning rates in 100 iterations, as shown in tables (results) above. If we increase the iterations and reduce the learning rate then we will be able to train the model more accurately and minimize the loss more.