Predicting Diamond Prices

(a very important life skill)

Shelley Jones
(owner of very few diamonds)

Question

“Using data that is already publicly available, can we establish a predictive model using Linear Regression that accurately estimates diamond prices using available features such as carat, cut, color, clarity, and other attributes, and achieve a target R^2 score of at least 0.85 within the next 4 weeks?"

- **Specific**: Focuses on establishing a predictive model to estimate diamond prices using linear regression and available features.
- **Measurable**: The success criterion is achieving a target R^2 score of at least 0.85.
- **Achievable**: Assuming that a target R^2 score of 0.85 is feasible based on historical data and model capabilities.
- **Relevant**: Directly related to improving the accuracy of diamond price predictions, which is the goal.
- **Time-bound**: Sets a deadline of 4 weeks to achieve the target R^2 score.

Overview of Features

Data Dictionary

The dataset contained 10 features in which 'Price (in US dollars)' was the dependent feature.

1.	Carat (Weight of Diamond):	Weight of Diamond
2.	Cut (Quality):	Quality of cut (Fair, Good, Very Good, Premium, Ideal)
3.	Colour:	Diamond Colour (from J: 'worst' to D: 'Best')
4.	Clarity:	Measurement of Transparency from I1 (worst quality) through to IF(best quality)
5.	Table:	Width of top of a Diamond
6.	Price (in US dollars):	Price of Diamond in US dollars
7.	X(length):	Length of Diamond in mm
8.	Y(width):	Width of Diamond in mm
9.	Z(depth):	Depth of Diamond in mm
10.	Depth:	Total depth percentage where Total Depth % = z / mean(x , y) or z * 2 / (x + y)

Initial Exploratory Steps

	Carat(Weight of Daimond)	Cut(Quality)	Color	Clarity	Depth	Table	Price(in US dollars)	X(length)	Y(width)	Z(Depth)
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

- 53,940 rows: No missing data
- Variables were a mixture of integers, objects and floats
- Renamed columns for ease of coding
- Checked categorical columns for ‘unexpected’ entries: none found
- Checked for duplicate rows: 146 were identified and were removed

	carat	cut	colour	clarity	calcdepth	table	price	length	width	depths
1005	0.79	Ideal	G	SI1	62.3	57.0	2898	5.90	5.85	3.66
1006	0.79	Ideal	G	SI1	62.3	57.0	2898	5.90	5.85	3.66
1007	0.79	Ideal	G	SI1	62.3	57.0	2898	5.90	5.85	3.66

Final headers (here showing some of the duplicate values before cleaning)

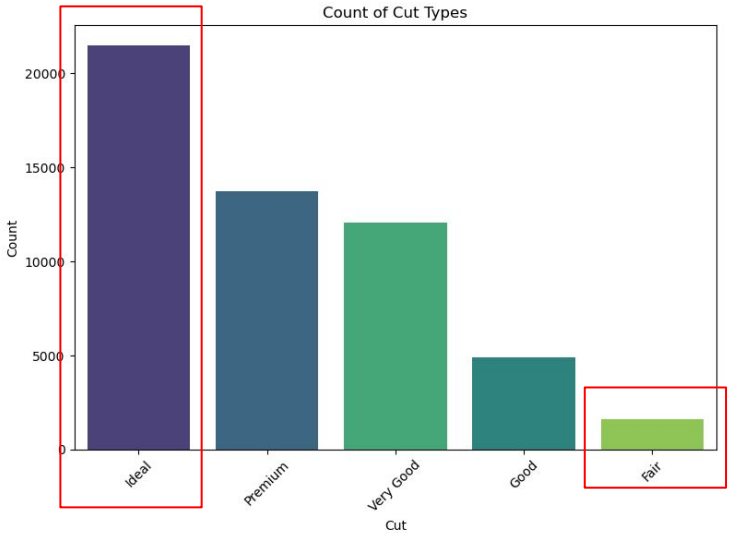
EDA Cont'd

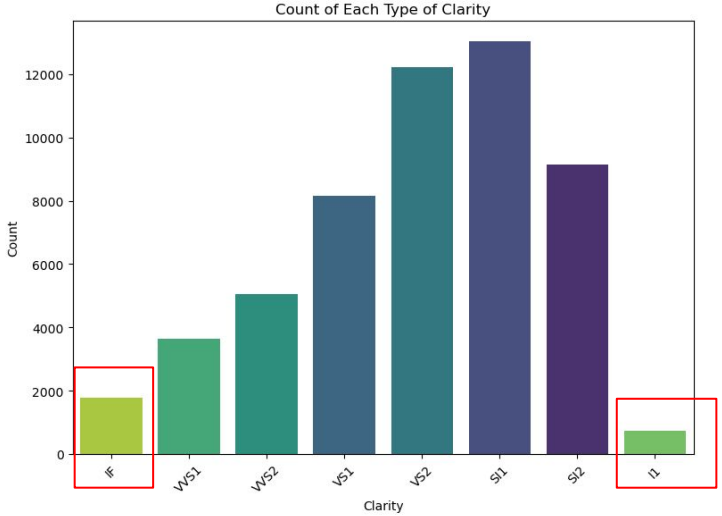
- Ran initial stats (df.describe) and identified min values of '0' for length, width and depth, which should not be possible
  - Identified it affected 19 rows, so opted to delete those rows
- Also appeared to be a high 'max' compared to 75% quartile for all variables - maybe check for outlier(s)?

	count	mean	std	min	25%	50%	75%	max
carat	53794.0	0.797780	0.473390	0.2	0.40	0.70	1.04	5.01
calcdepth	53794.0	61.748080	1.429909	43.0	61.00	61.80	62.50	79.00
table	53794.0	57.458109	2.233679	43.0	56.00	57.00	59.00	95.00
price	53794.0	3933.065082	3988.114460	326.0	951.00	2401.00	5326.75	18823.00
length	53794.0	5.731214	1.120695	0.0	4.71	5.70	6.54	10.74
width	53794.0	5.734653	1.141209	0.0	4.72	5.71	6.54	58.90
depths	53794.0	3.538714	0.705037	0.0	2.91	3.53	4.03	31.80

EDA cont’d

- Value counts of categorical data vs Price
- e.g. Quality of the Cut vs Price and Clarity vs Price

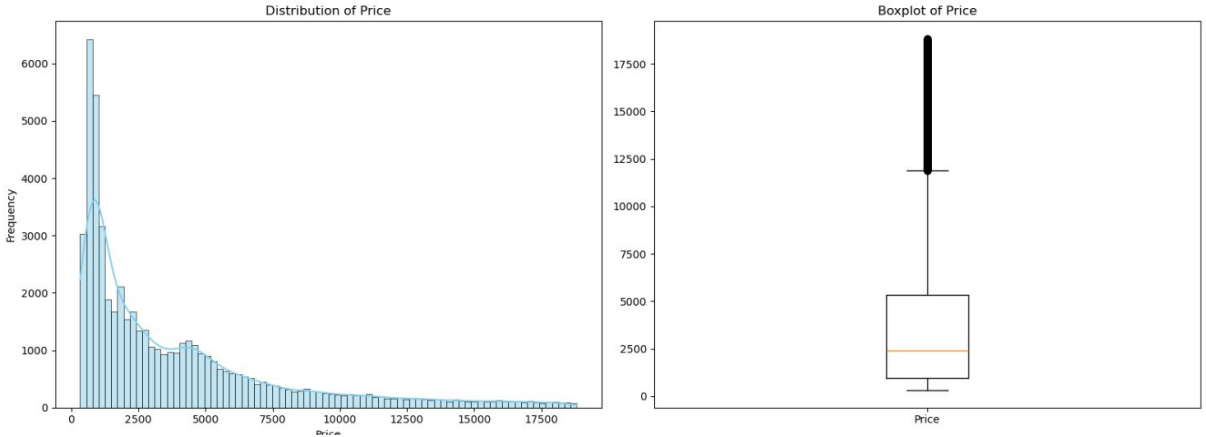




Count sizes for Cut and Clarity vary greatly

EDA Cont’d

Target feature: Price

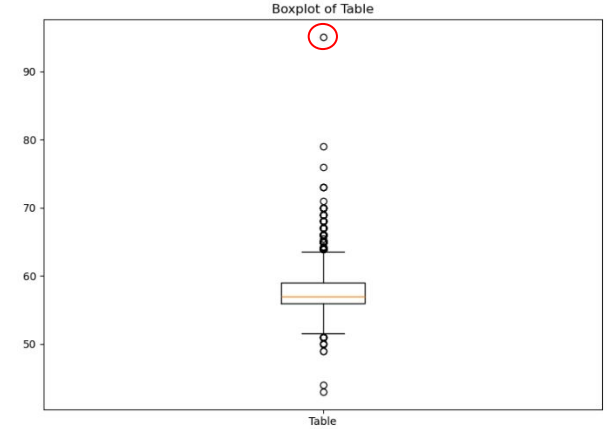


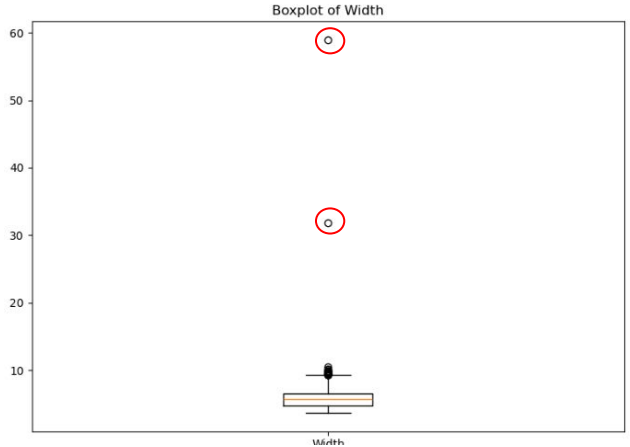
- Target variable (Price) was not normally distributed
- Very right skewed and could affect the modelling. May need to consider transforming the data e.g. BoxCox
- Boxplot indicated potential ‘outliers’, but given the number and lack of domain knowledge, I opted to keep them in the data set

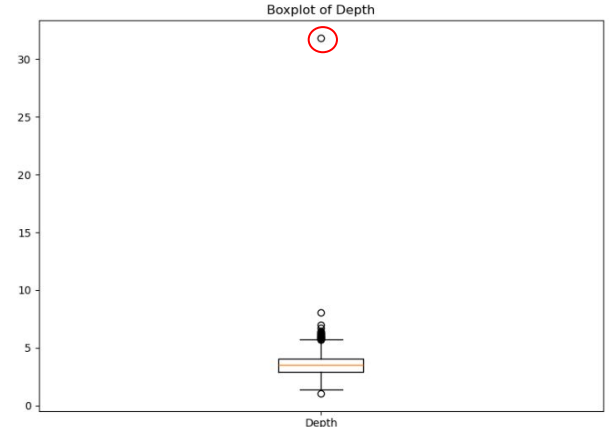
EDA Cont’d

Other Features:

- Similar potential ‘outliers’ seen across the majority of the features
- Boxplots for Width, Depth and Table however contained values that could be considered for exclusion



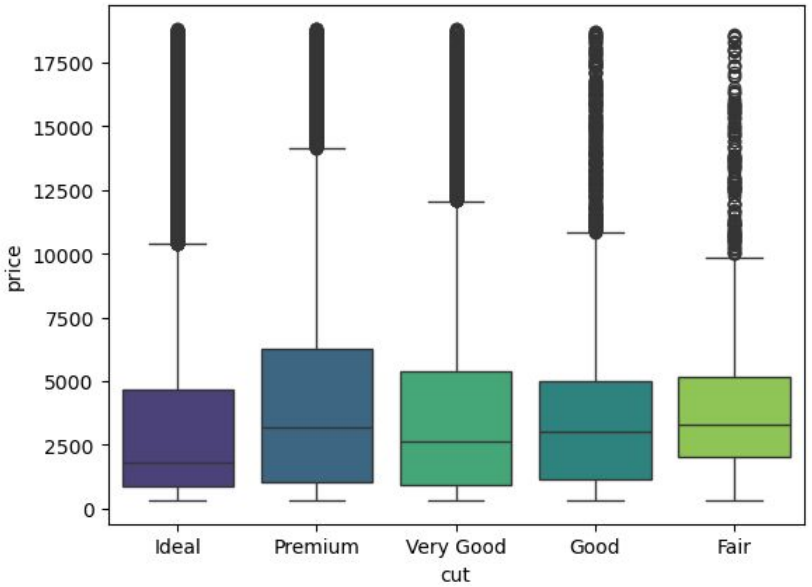




EDA Cont’d

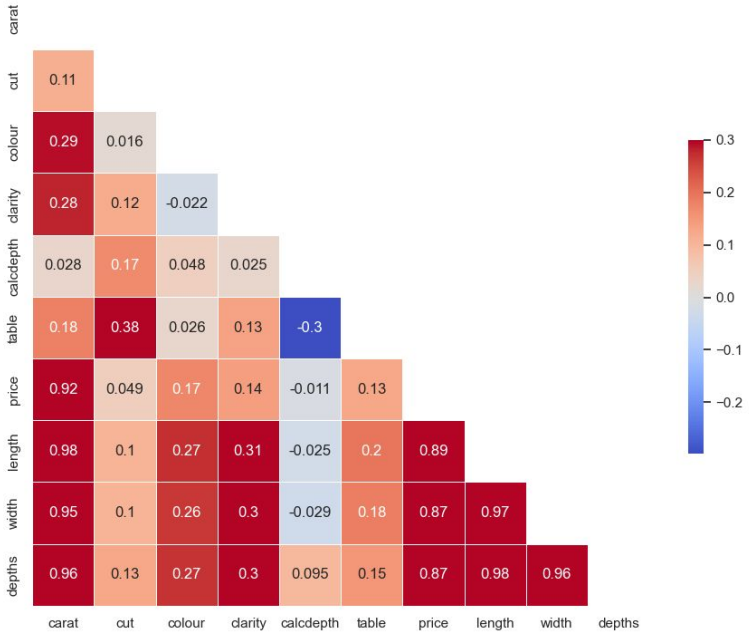
Categorical data:

- Seeing the same potential ‘outliers’ as for numerical features
- ‘Ideal’ cut had a lower mean price than all other cut types
  - Same was seen for best ‘Colour’ and best ‘Clarity’
  - Why was a perfectly cut stone | best colour | best clarity worth less?
  - Maybe they are not strong predictors of price?
- Finally I converted categorical data to numbers to allow for further analysis e.g. pairplot



Feature Selection

Feature correlation

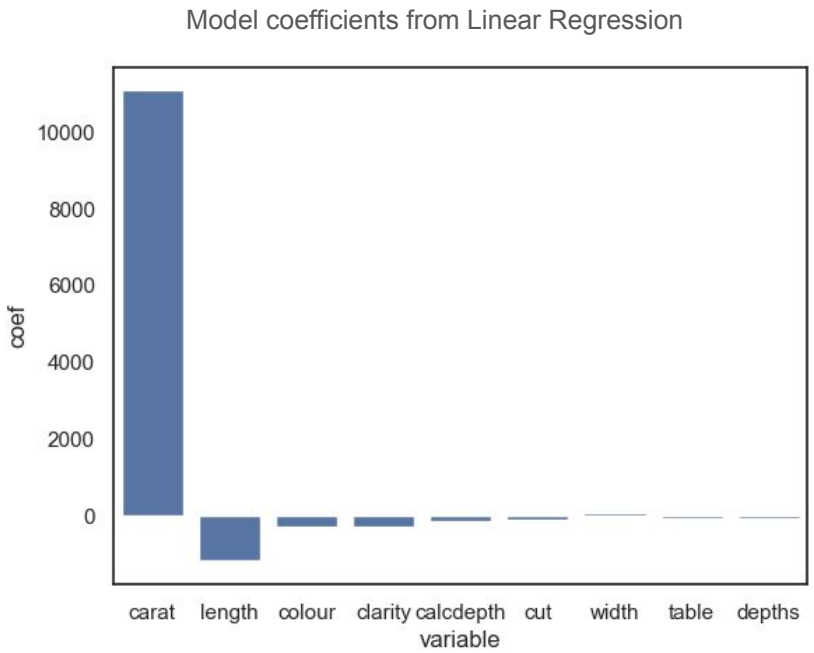


- First thoughts were that Carat, Length, Width and Depths were most strongly positively correlated
- Concerned that Length, Width, Depths (which are all indicators of diamond size) will have a strong influence on the predicted price (multicollinearity)
- In future it might be worth considering using an engineered feature e.g. 'volume' (L*W*D)
- Forward Feature Selector resulted in: Carat, Clarity, Colour, Length, Cut, CalcDepth, Table and Width!

Modelling

Set up

- Used all the data ‘as is’
  - With no domain knowledge I decided not to exclude any data at this point
- Created a model for Linear Regression
- Defined the target and predictor features:
  - Target = Price
  - Predictor features: All other columns
- Performed Train-Test split
  - Test size= 0.20
  - Set Random State for reproducibility
  - Checked shape of train / test arrays
- Fitted the Linear Regression Model from SciKit
- Calculated the model coefficients using model.coef_
  - Charted on the right for visualisation
  - Carat has a very high coefficient compared to all other features



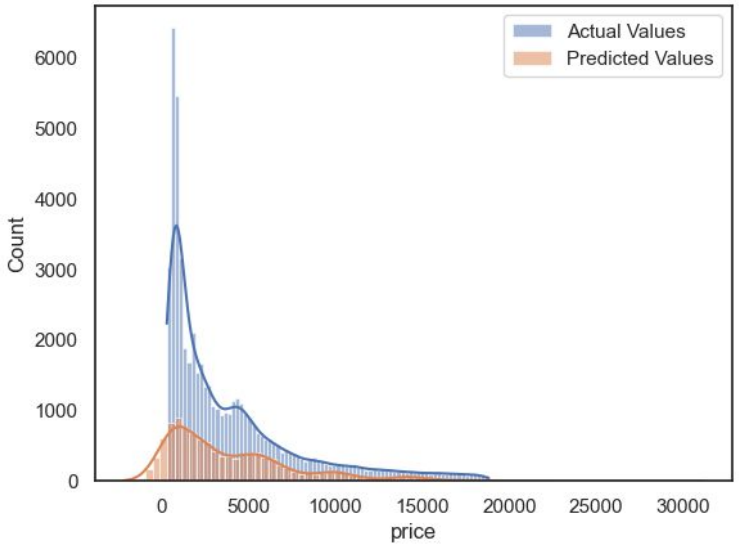
Modelling Cont’d

Model Testing

- Used the model to predict prices using the test data

Visualised the model

- See Histplot on right



Model Evaluation

Calculated R-squared scores:

- This provides a measure of how well the model explains the variability of the target variable
  - **R^2 for Training Data = 0.8853**: This means that approximately 88.5% of the variance in the training data's target variable can be explained by the features used in the model. This is a good sign as it indicates that the model fits the training data well.
  - **R^2 for Test Data = 0.8862**: This means that approximately 88.6% of the variance in the test data's target variable can be explained by the features used in the model. This score is very close to the training R^2 score, suggesting that the model generalizes well to unseen data.

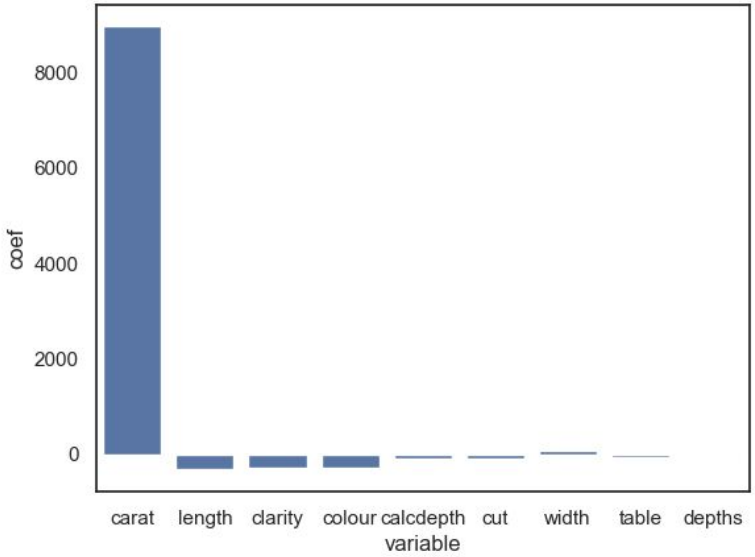
Calculated Errors:

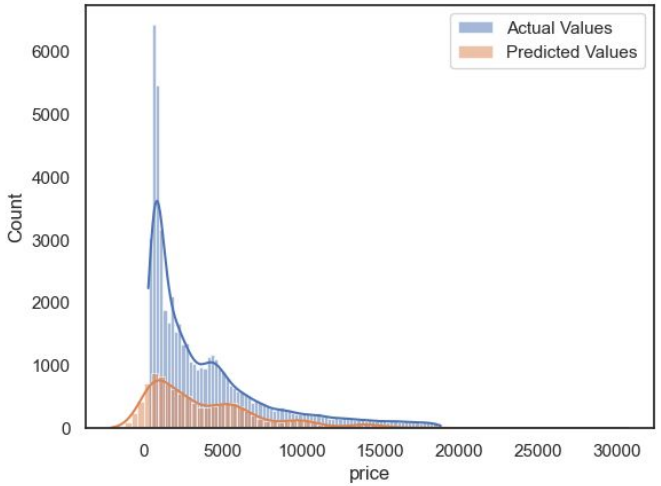
- For regression tasks, metrics like RMSE (Root Mean Squared Error) or MAE (Mean Absolute Error) can provide additional insights into the model's prediction errors
  - **Mean Absolute Error (MAE) = 807.45** (model's predictions are off by approx. 807.45)
  - **Mean Squared Error (MSE) = 1798231.81** (penalizes larger errors more than smaller ones, making it sensitive to outliers. Large value of MSE indicates that there are some significant deviations in predictions.)
  - **Root Mean Squared Error (RMSE) = 1340.98** (typical prediction error is around 1,340.98 units)

Other Points to Consider:

- **Consistency**: The closeness of the training and test R^2 scores suggests that the model is not overfitting. Overfitting occurs when a model performs very well on the training data but poorly on the test data.
- **Domain Context**: The significance of an R^2 score also depends on the domain and the nature of the problem. In some fields, an R^2 score above 0.8 is excellent, while in others, even higher scores might be expected.

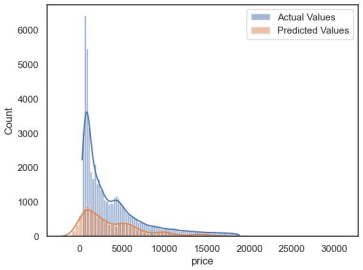
Regularisation using Ridge





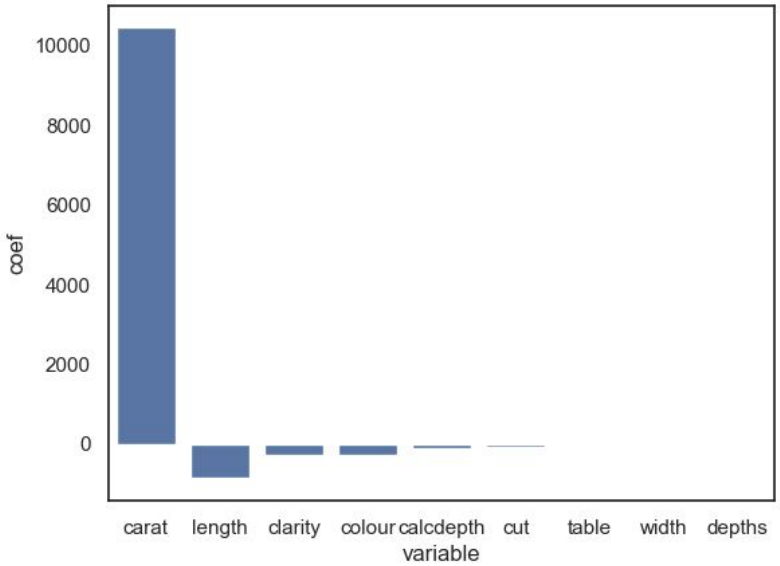
Ridge Coefficients: [8960.96527853
-88.78261283 -275.28575818 -293.63189062
-102.27282875 -55.48393476 -326.63782279
78.5792258 -23.99969211]

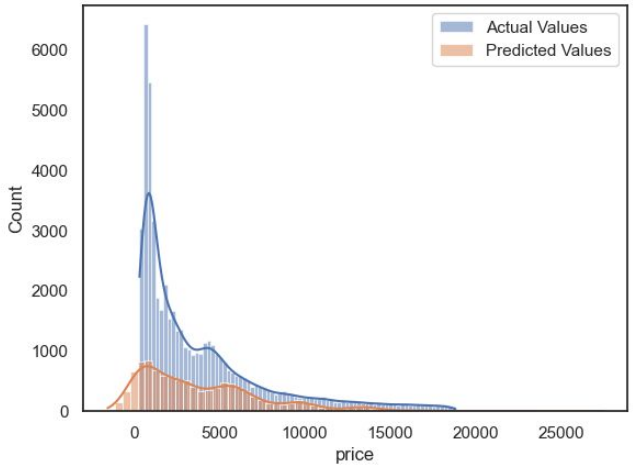
R2_ridge: 0.883
MAE: 853.93
MSE: 1842114.55
RMSE: 1357.25



Original plot

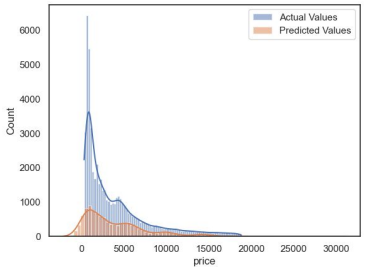
Regularisation using Lasso





Lasso Coefficients: [10434.74555177
-87.02046912 -284.29098989 -284.96997424
-128.91171583 -58.11976031 -870.97605655
0. -0.]

R2_lasso: 0.886
MAE: 818.19
MSE: 1804445.77
RMSE: 1343.3



Original plot

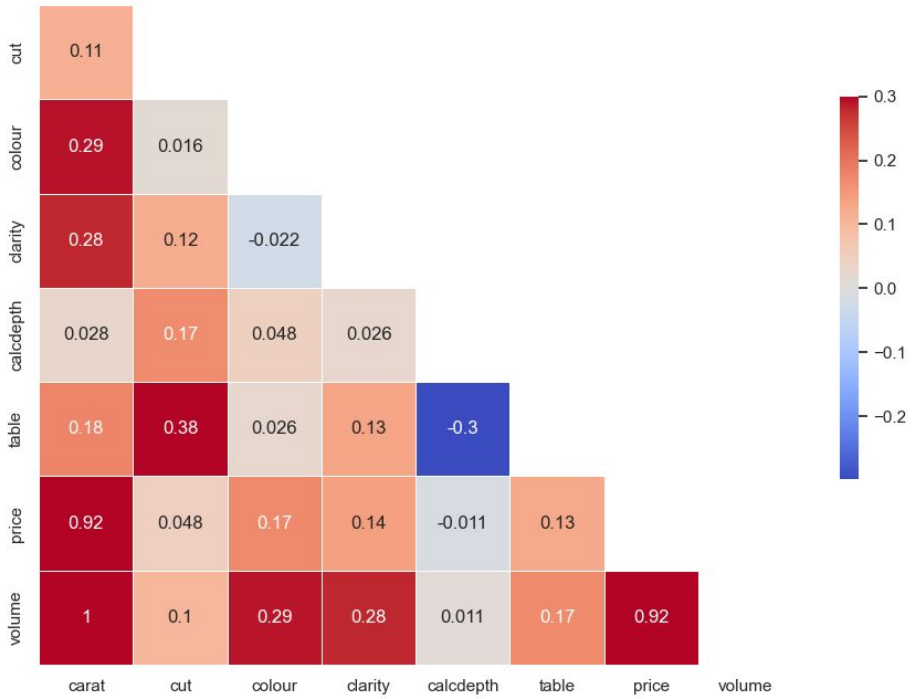
(Additional)

I then omitted the potential outliers

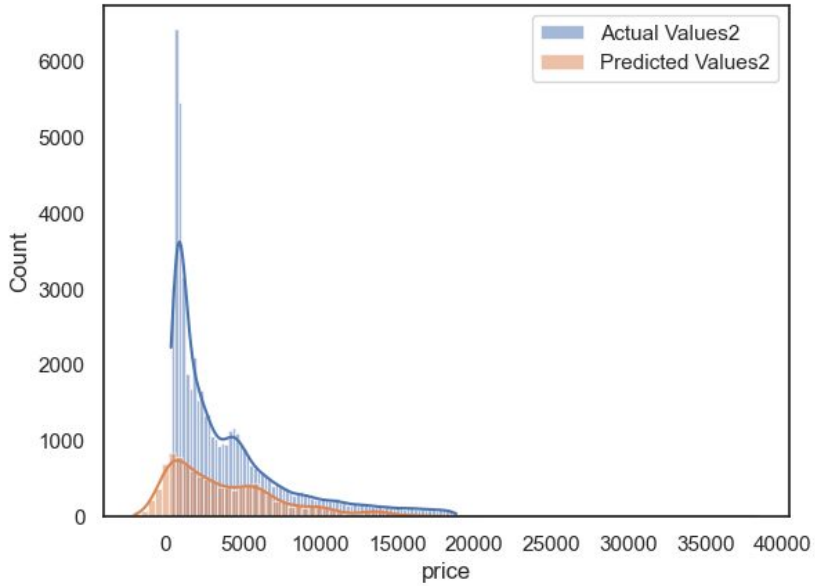
Converted L,W,D to Volume

Re-modelled

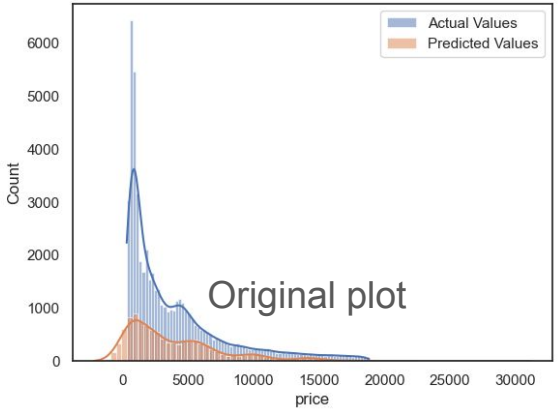
Strong correlators: Carat and Volume



Re-Modelled Results



R2: 0.8819
MAE: 873.22
MSE: 1839871.59
RMSE: 1356.42



Comparisons

Not a lot between them!

Original method performs marginally better

Method	R2	MAE	MSE	RMSE
Original	**0.8862**	**807.45**	**1798231.81**	**1340.98**
Original + Ridge	0.8834	853.928	1842114.55	1357.245
Original + Lasso	0.8858	818.194	1804445.77	1343.296
Remodelled dataset	0.8818	873.216	1839871.59	1356.418

Conclusions

Headlines:

- Approximately 88% of the variance in the test data's target variable could be explained by the features used in the model.
- The model generalised well to unseen data.
- Large value of MSE indicates that there are some significant deviations in predictions
- Typical prediction error is around US$1,341
- Model was not overfitting

Other Points to Consider:

- Get an expert with domain knowledge to advise on feature selection and possible outliers
- Consider a BoxCox transformation to help eliminate potential problems due to heavily skewed data
- Use a scaler (removes the mean and scales to unit variance)?
- Use a different method!

Finally:

- I would **NOT** recommend going into the diamond business (under charging) or diamond shopping (bill shock) relying on this model!

Questions?