A Regression Analysis for Outcome Predictions on the MLF Coursework Dataset

Lupupa Chansa

February 19, 2026

Abstract

This report presents a data science and machine learning pipeline for predicting a continuous target variable. From a given dataset, this approach begins with preprocessing, then feature engineering and model comparison to produce a final prediction file for evaluation.

https://github.com/lupnpa/MLF-Courswork-1

1 Exploratory Data Analysis

1.1 Dataset Overview and Preprocessing

The initial check was to remove any missing values from the given dataset, as well as creating a set of all values in each categorical column to identify potential misspelled words and determine how many different values each feature could take. No missing values were detected, therefore no rows were removed.

The categorical features had ordinal properties, so I considered that encoding this information was potentially useful.

Scatter plots were used to visualise outliers present in each feature. Most features were fairly symmetrical apart from 'price', 'carat' and 'table', which were positively skewed, indicating that large and expensive values are rare.

1.2 Correlation Analysis and Feature Selection

Strong correlations were observed between several features, as illustrated in the correlation heatmap (Figure 2). In particular, x' and y' had a correlation coefficient of 0.91, and carat' and z' showed a correlation of 0.97. High multicollinearity can inflate the variance of coefficient estimates in linear models and potentially increase model variance.

A function was implemented to create clusters of features with pairwise correlations above 0.9. Within each cluster, the feature most strongly correlated with the outcome was retained. This reduced the feature space from 31 to 29 variables while preserving predictive information.

2 Model Selection

2.1 Algorithm Comparison

Three regression algorithms were compared:

- Linear Regression (L.R.) as a baseline model,
- Random Forest (R.F.),
- Multi-layer Perceptron (MLP).

Outlier capping improved the performance of the linear regression model; however, it did not improve the performance of the Random Forest or MLP. Since the Random Forest remained the best performing model throughout, outlier capping was not applied in the final model.

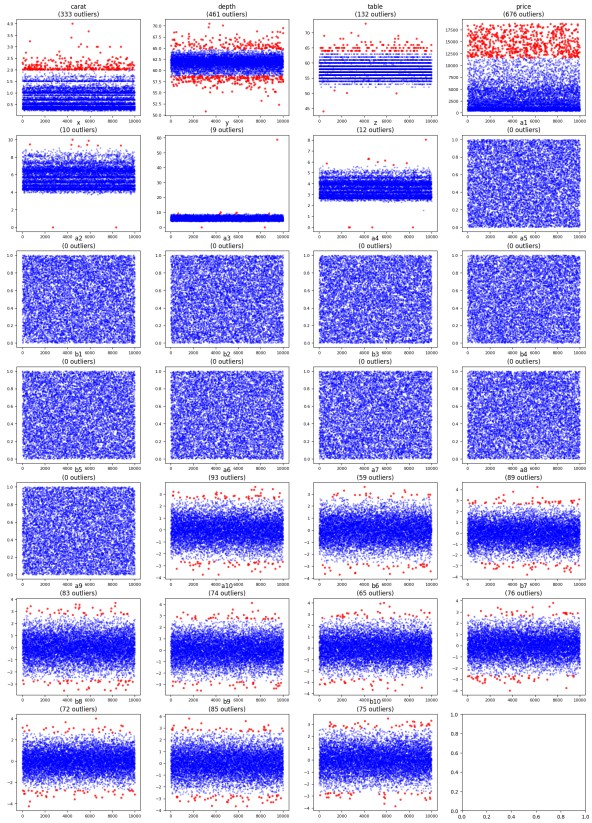

Figure 1: Scatter plot visualization of feature outliers.

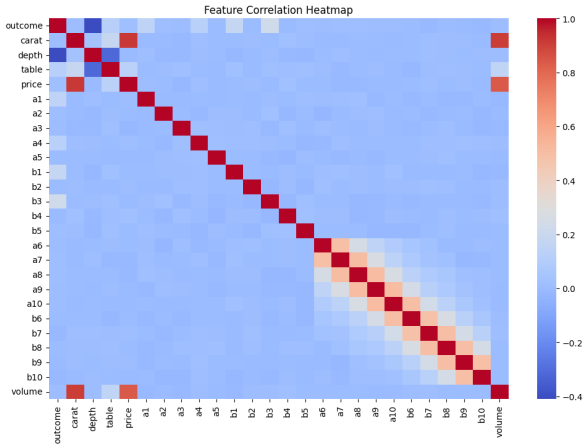

Figure 2: Correlation heatmap showing pairwise correlations between features.

3 Model Training and Evaluation

3.1 Hyperparameter Tuning

A grid search with 5-fold cross-validation was used to determine optimal hyperparameters and reduce variance during training.

The Linear Regression model performed worst overall, achieving an R^2 score of 0.268, likely because it fails to capture non-linear relationships in the data.

The MLP initially failed to converge. After adjusting the learning rate and increasing the maximum iterations, stable convergence was achieved. The best configuration was:

- $\alpha = 0.01$
- hidden layers = (64, 32)
- learning_rate_init = 0.0005

The two-layer architecture provided sufficient capacity to model non-linear relationships without excessive variance. Increased regularisation ($\alpha = 0.01$) improved generalisation by penalising large weights, while a reduced learning rate stabilised training. The best cross-validated R^2 was 0.366.

The Random Forest was the best model, achieving a test R^2 of 0.437. The selected hyperparameters were:

- max_depth = 10
- min_samples_split = 5
- n_estimators = 200

Increasing the number of trees reduces variance by averaging over more decision trees. Performance stabilised around 200 estimators, beyond which improvements were negligible. The model performed better with a maximum depth of 10, likely because this stops unnecessary complexity arising. Increasing the minimum samples required for a split reduces the likelihood of modelling noise within small partitions, improving generalisation.

When the hyperparameter grid was expanded further, cross-validated performance improved slightly, but the test R^2 decreased marginally (0.435 vs 0.437), suggesting some overfitting to the cross-validation sets. Therefore, the simpler configuration was selected.

3.2 Final Model Performance

Model	R^2
Linear Regression	0.268
Neural Network	0.366
Random Forest	0.437

Table 1: Comparison of model performance.

4 Conclusion

The Random Forest outperformed both the Linear Regression and Neural Network models. It was able to model non-linear feature interactions while controlling variance through ensemble averaging, making it the most suitable choice for this dataset.