Diamond Price Prediction Report

Introduction

Predicting diamond prices serves practical purposes for both consumers and industry stakeholders. Accurate predictions ensure fair pricing, transparency, and consumer confidence. For retailers and wholesalers, it aids in inventory selection, pricing strategies, and sales optimization. In a broader context, the diamond industry is a multi-billion-dollar global market, impacting market stability, investor confidence, and the economies of major diamond-producing countries.

Exploratory Data Analysis

Our Diamond dataset comprises 53,940 records and 10 features, focusing on understanding the relationships between diamond attributes and prices. Features include carat, cut, color, clarity, depth, and table, providing valuable insights into factors influencing diamond prices.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53940 entries, 0 to 53939
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  53940 non-null  int64
 1   carat       53940 non-null  float64
 2   cut         53940 non-null  object
 3   color       53940 non-null  object
 4   clarity     53940 non-null  object
 5   depth       53940 non-null  float64
 6   table       53940 non-null  float64
 7   price       53940 non-null  int64
 8   x           53940 non-null  float64
 9   y           53940 non-null  float64
 10  z           53940 non-null  float64
dtypes: float64(6), int64(2), object(3)
memory usage: 4.5+ MB
```

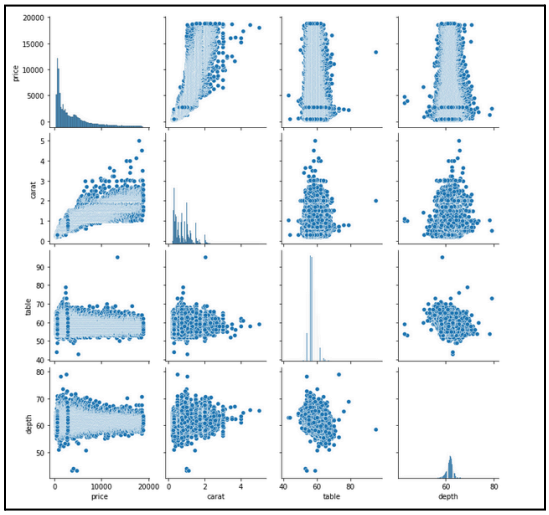
	Carat	depth	table	price	x	y	z
Mean	0.8	61.7	57.5	3932.8	5.7	5.7	3.5
std	0.5	1.4	2.2	3989.4	1.1	1.1	0.7
min	0.2	43.0	43.0	326.0	0.0	0.0	0.0
25%	0.4	61.0	56.0	950.0	4.7	4.7	2.9
50%	0.7	61.8	57.0	2401.0	5.7	5.7	3.5
75%	1.0	62.5	59.0	5324.3	6.5	6.5	4.0
max	5.0	79.0	95.0	18823.0	10.7	58.9	31.8

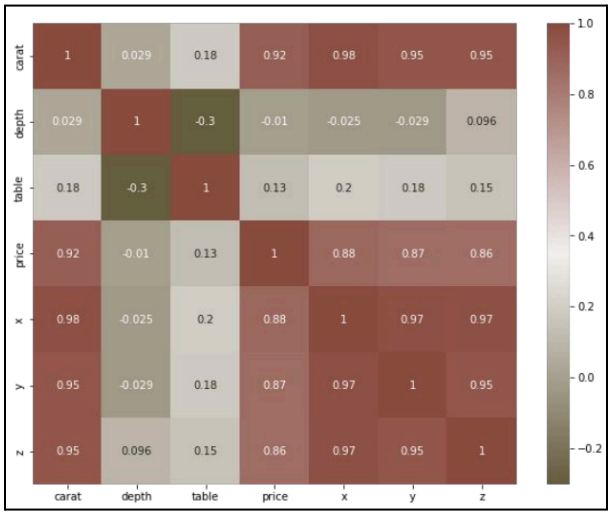
The average carat weight of the diamonds is 0.78 ct, with a standard deviation of 0.47 ct, which indicates a moderate variation in diamond weight. The carats range widely from a petite 0.20 ct to a luxurious 5.01 ct, showcasing the dataset's inclusivity of both lighter and heavier diamonds. The depth(which is a percentage of total height by average of length and width) sits at a mean of 61.75%, with most diamonds clustering close to this average, as the standard deviation is not large. The table size, an essential factor in a diamond's brilliance, averages at 57.46 mm across the dataset. This wide range suggests the dataset includes a diverse array of diamond cuts. On the financial side of things, the average price tags for these gems is at $3,9323, but the prices span a vast range from a modest $326 to a staggering $18,823. This significant spread in prices reflects the diamonds' varied characteristics and the market's valuation of them. Examining the physical dimensions—length (x), width (y), and depth

(z)—we see that they average around 5.73 mm, 5.73 mm, and 3.53 mm, respectively, with the standard deviation pointing to a reasonable consistency in diamond proportions.

Data Cleaning and Preprocessing

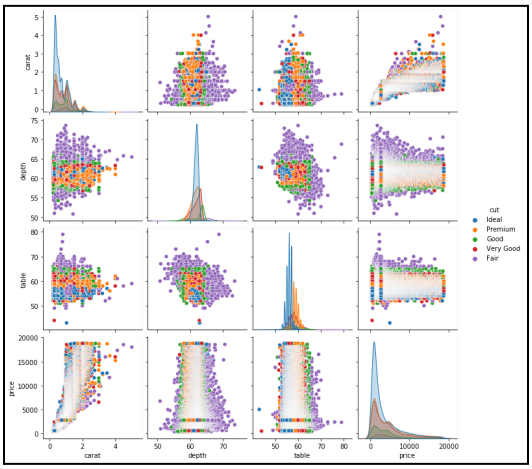
The dataset is explored to understand its structure, identify missing values, and detect outliers. We verified the absence of missing values, ensuring data completeness. The maximum values for the 'depth', 'table' dimensions suggest potential outliers, given their disproportionate scale compared to the rest of the data. We used scatterplots to further verify outliers, capping Depth from 45% to 75% and Table from 40 mm to 80 mm. 7 outliers were removed in total.

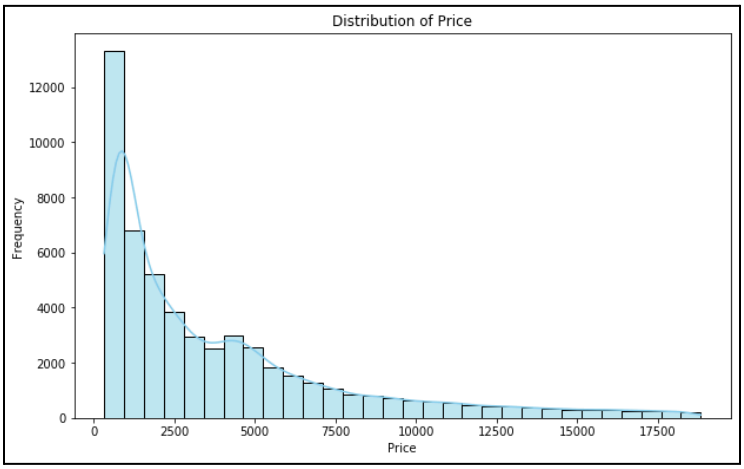




Since we also found high correlations among the 'x', 'y', and 'z' dimensions and other features we excluded these variables to prevent multicollinearity issues that may affect our model accuracy. Categorical variables such as 'cut', 'color', and 'clarity' are converted to ordered variables to prepare our data for exploratory data analysis and predictive modeling.

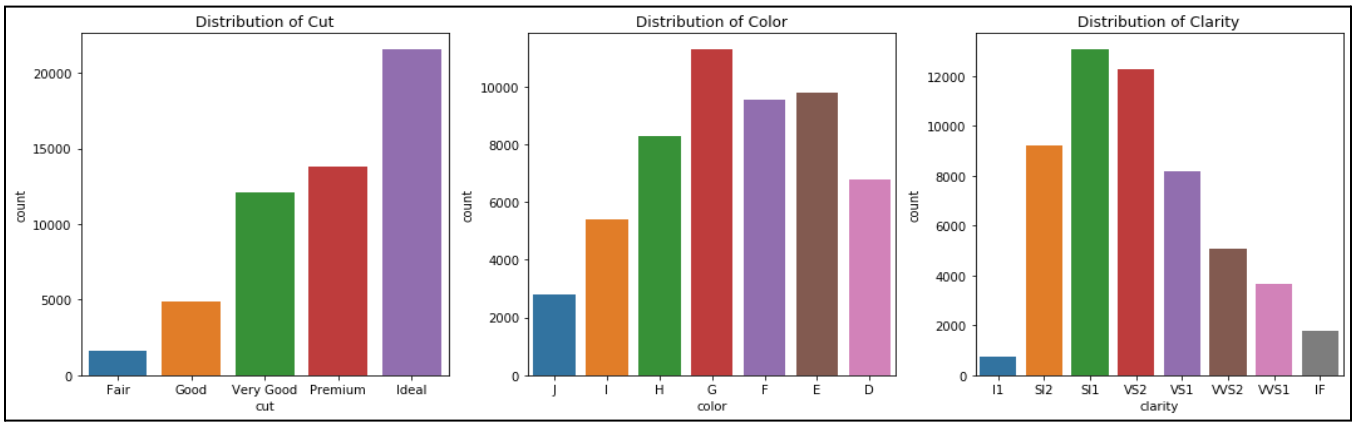
Univariate/Bivariate Analysis



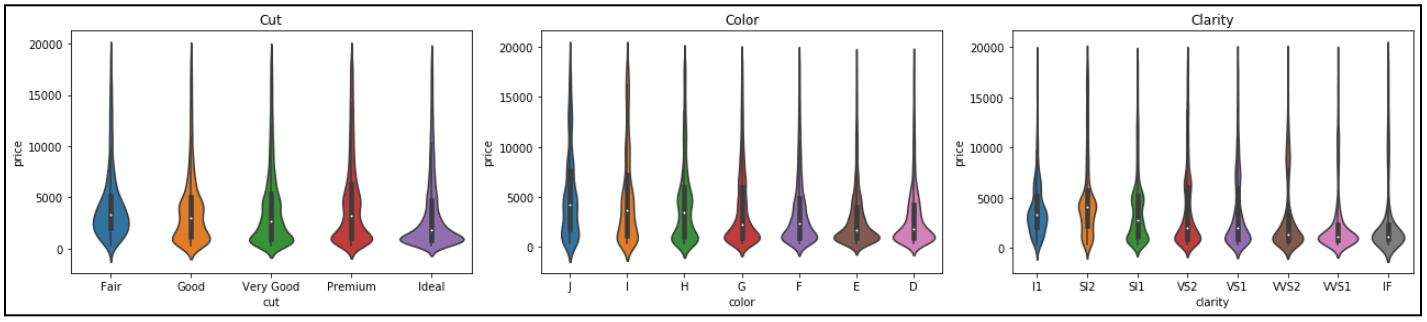


Right after our initial exploratory analysis we are able gain the following insights from our dataset:

A positive correlation was observed between the carat of a diamond and its price. Price distribution is right-skewed, the majority falling below 2500.



The cut, color, and clarity of diamonds have varying distributions across the dataset, with 'Ideal' cut 40%, 'G' color 21%, and 'SI1' clarity 21% being the most common respectively.



Violin plots revealed that higher-quality cuts, colors, and clarities tend to have less mean prices, although the relationship is not strictly linear or uniform

Models Used and Performance Metrics

Model	MAE	MSE	R Square
Randon Forest Regression	199	138965	0.9912
XGBoost Regressor	234	184926	0.9883
Lightgbm	254	225736	0.9858
Gradient Boosting	257	224442	0.9858
Decision Tre Regression	258	233988	0.9852
Catboost	266	235796	0.9851
Lasso Regression	335	366907	0.9769
Linear Regression	341	370080	0.9767
Ridge Regression	341	372471	0.9765
MLP Regressor	337	379798	0.9760
K-Neighbours	644	1433463	0.9096
SVM	723	2151025	0.8643

Model	MAE	MSE	R Square
XGBoost Regressor	273	271981	0.9831
Lightgbm	277	277102	0.9827
Gradient Boosting	278	277972	0.9827
Catboost	285	280573	0.9825
Random Forest Regression	282	294568	0.9817
Decision Tre Regression	323	379900	0.9763
MLP Regressor	342	381104	0.9763
Lasso Regression	340	381945	0.9762
Linear Regression	348	399835	0.9751
Ridge Regression	348	402079	0.9750
SVM	732	2210437	0.8623
K-Neighbours	977	3108979	0.8064

Model	Parameters
Random Forest	{'n_estimators': [50, 100, 150], 'max_depth':range(1, 25)}
Gradient Boosting	{'n_estimators': [50, 100, 150], 'learning_rate': [0.01, 0.1, 0.2], 'max_depth': [3, 5, 7]}
XGBoost	{'learning_rate': np.logspace(-2, 0, 3), 'n_estimators': [50, 100, 200], 'max_depth': [3, 5, 7]}
LightGBM	{'n_estimators': [50, 100, 150], 'learning_rate': [0.01, 0.1, 10,100], 'max_depth': [6, 8, 10, 12], 'num_leaves': [31, 63, 127]}
Decision Tree	{'max_depth':range(1, 25)}
Catboost	{'iterations': [50, 100, 150], 'learning_rate': [0.01, 0.1, 0.2], 'depth': [3, 5, 7]}
MLP	{'hidden_layer_sizes': [(50,), (100,), (50, 50), (100, 50)], 'max_iter': [200, 300, 500], 'alpha': [0.0001, 0.001, 0.01]}
Ridge	{'polynomialfeatures__degree': range(1, 10)}
Lasso	{'polynomialfeatures__degree': range(1, 10)}
Linear Regression	{'polynomialfeatures__degree': range(1, 10)}
K- Neighbours	{'n_neighbors': [3, 5, 7, 9, 11], 'p': [1, 2] }
SVM	{'C': [0.1, 1, 10, 100], 'gamma': [0.01, 0.1, 1, 10]}

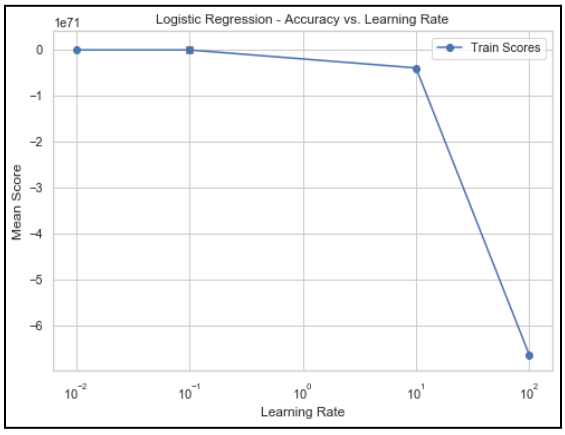
The data has been split into 70% training and 30% testing. The above table showcases performance metrics for various regression models, evaluated on both training and testing datasets. There are three key metrics presented: MAE (Mean Absolute Error), MSE (Mean Squared Error) & R score (R-squared)

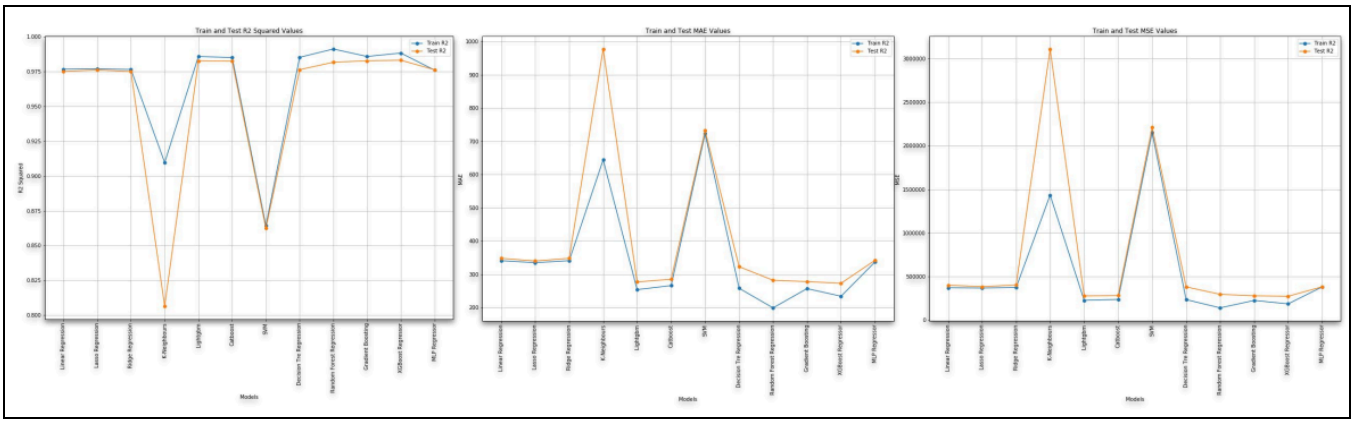
Interpretation of Results:

- Random Forest Regression has the best performance on the training data and moderate on testing data across all three metrics suggesting a sign of overfitting.
- Gradient Boosting, XGBoost Regressor, and LightGBM follow closely, indicating that ensemble methods are performing well on training and testing data.
- SVM and K-Neighbors have the worst performance on the training data as well as testing data suggesting that these models are not well suited to this particular problem.

Best Model (LightGBM) and Parameters

```
# Define the hyperparameters to tune
param_grid = {
    'n_estimators': [50, 100, 150],
    'learning_rate': [0.01, 0.1, 10,100],
    'max_depth': [6, 8, 10, 12],
    'num_leaves': [31, 63, 127]
}

# Use GridSearchCV to tune the hyperparameters
grid_search_lgbm = GridSearchCV(lgbm, param_grid, cv=5, scoring='r2',return_train_score=True)
grid_search_lgbm.fit(X_train, y_train)

# Get the best hyperparameters and best score
best_params_lgbm = grid_search_lgbm.best_params_
best_score_lgbm = grid_search_lgbm.best_score_

print('--------------------------------')
print("Best parameters: ", best_params_lgbm)
print(f"Best R Squared score: {best_score_lgbm*100:.2f}%")

#print(grid_search.cv_results_.keys())
best_lgbm = grid_search_lgbm.best_estimator_

# Calculate the r2 for the training set
print('--------------------------------')
print(f"Train R Squared Value with best parameters: {r2_score(y_train, best_lgbm.predict(X_train))*100:.2f}%")

# Calculate the r2 for the Test set
print(f"Test R Squared Value with best parameters: {r2_score(y_test, best_lgbm.predict(X_test))*100:.2f}%");
```

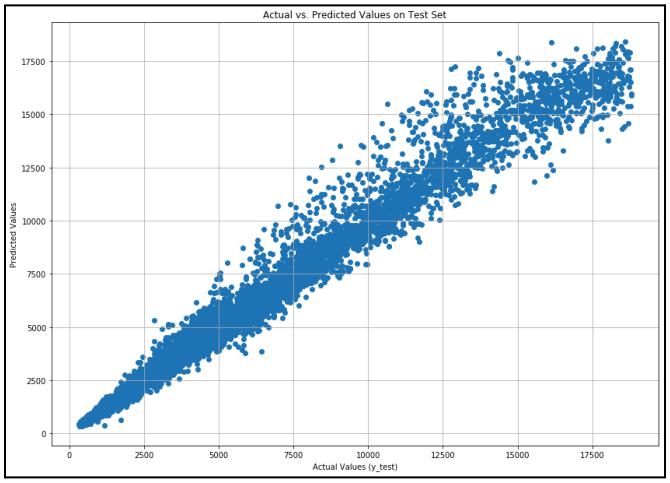


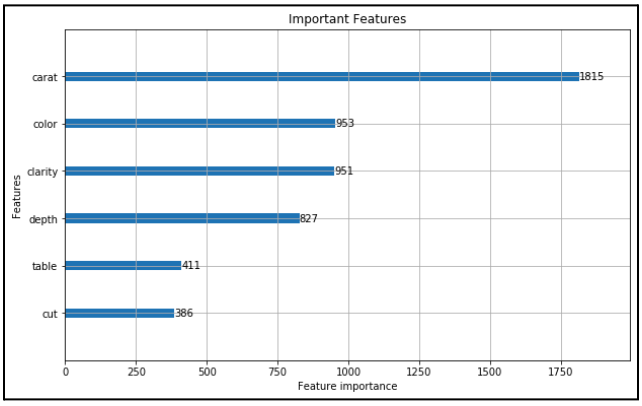


After careful consideration of trade-offs between training and testing data as seen in the line chart for all models, we have identified LightGBM as the optimal model for our final selection. This decision was based on its superior performance, exhibiting a high R-squared score in both training (0.9858) and testing (0.9827) datasets compared to other models.

To train the LightGBM model, we explored various combinations of hyperparameters. After thorough evaluation, the best parameters were determined to be **{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 150, 'num_leaves': 63}.**

Visualizing the model's performance on the test data through a graph of actual versus predicted diamond prices revealed a close alignment, indicating the model's efficacy in price prediction. Furthermore, to gain insights into the relative importance of different features in predicting diamond prices, we examined the feature importance plot generated for LightGBM. Notably, the feature importance analysis highlighted 'Carat' as the most influential variable, suggesting its significance in determining the target variable (price). This underscores the pivotal role of carat weight in diamond pricing and reinforces its prominence in our predictive model.





Sample testing

Diamond	Carat	Cut	Clarity	Color	Depth	Table	Actual Price ($)	Predicted Price ($)
1	0.29	Premium	VS2	Colorless [E]	62.4	58	334	434
2	0.24	Very Good	VVS2	Colorless [D]	62.8	57	336	318
3	0.23	Ideal	VS1	Colorless [D]	62.8	56	340	304
4	0.30	Good	SI1	Colorless [D]	63.8	56	351	381
5	0.22	Fair	VS2	Colorless [E]	65.1	61	337	374

We can see a range of deviations in predicted prices across the dataset, with very close results compared to actual values. Notably, deviations were not consistent, with some predictions overestimating while others relatively much more accurate, indicating the model's flexibility and adaptability to different scenarios.

Business Insights

Predicting diamond prices offers numerous practical advantages across sectors. By identifying distinct segments within the market based on attributes like carat weight, cut quality, color grade, and clarity grade, businesses can tailor their strategies for each segment. Understanding price sensitivity allows for targeted pricing adjustments to maximize revenue and market share, while product differentiation ensures offerings align with diverse customer preferences and budgets. By aligning offerings with customer value perceptions, businesses can enhance satisfaction and loyalty, while assessing competitive strength through predicted prices aids in strategic positioning. Finally, optimizing the supply chain based on factors driving diamond prices ensures cost efficiency and operational streamlining.

Limitations

While our diamond dataset provides valuable insights into predicting diamond prices, it also has several limitations to consider. Firstly, the dataset lacks information regarding the distinction between natural and synthetic diamonds, which can significantly impact market value. Additionally, the absence of origin information, such as country and mining location, limits our ability to account for variations in diamond quality and value influenced by geographical factors, mining regulations, and labor costs. Furthermore, the dataset does not include details on diamond certification (GIA, IGI etc), which are essential for verifying authenticity and quality, thus affecting pricing accuracy. These limitations underscore the need for caution when interpreting predictions and highlight areas for future data enhancement to improve price prediction accuracy.