Diamond.Hackers

Anmar Abdi, Rhoda Nankabirwa, Amy Boateng, Jose Salerno

# Deliverable 1

# Overview

The wholesale diamond dataset spans a period of ten years and serves as a comprehensive repository of diamond sales data for the company. It comprises 407,280 diamonds sold between 2012 and 2023, each represented by 11 attributes: carat, cut, color, depth, table, length (mm), width (mm), height (mm), cost (dollars), clarity, and year of sale.

The goal of the exploratory data analysis is to leverage this dataset to uncover patterns, trends, and correlations that can enhance our understanding of the diamond market. The insights gained from this exploratory analysis will guide our efforts in building predictive models for future diamond prices. By combining numerical and categorical data, we aim to identify factors influencing diamond prices and provide actionable recommendations for strategic decision-making.

# Data Quality

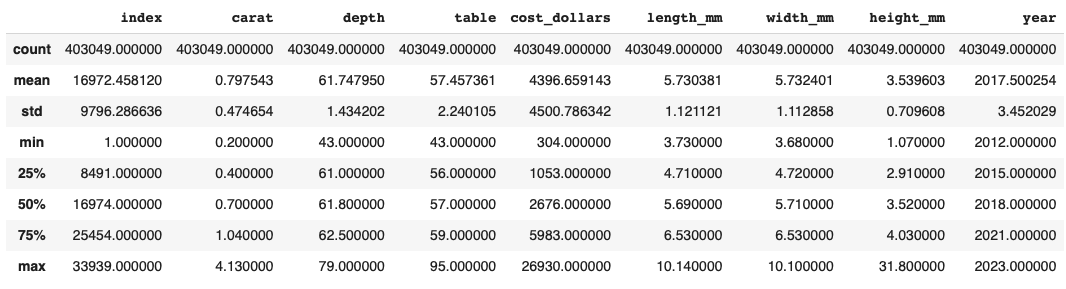
Data quality is a critical aspect that underpins the reliability and effectiveness of any analysis or modeling endeavor. In the context of our wholesale diamond dataset, ensuring data quality involved a thorough examination and cleansing process to address potential issues and inaccuracies, which included:

* 2048 blank entries for carat,
* 2037 negative entries for cost/diamond prices,
* 3 zero cost/diamond price,
* Several zero entries for length, width and height of diamonds,
* Column heading nomenclature, among others.
* Finally, after data cleaning, we ended with 403049 entries

# Findings

### Descriptive statistics

A fundamental aspect of our analysis involves summarizing the distribution of numerical attributes such as carat weight, diamond dimensions (length, width, height), and price. Calculating summary statistics, including measures such as mean, median, standard deviation, and range, will enable the department to gain insights into the central tendencies and variability of these critical variables. Figure 1 below provides the summary statistics of the cleaned diamond dataset.

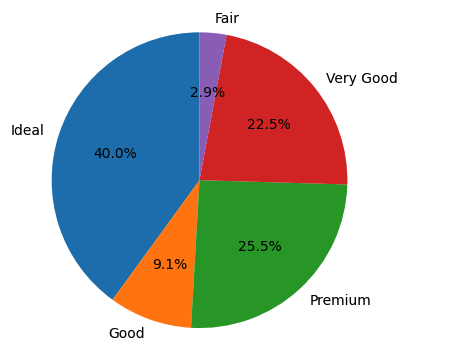


### Diamonds description.

The diamonds in the dataset can be classified according to three categorical attributes; cut type, color and clarity.

**Cut type**

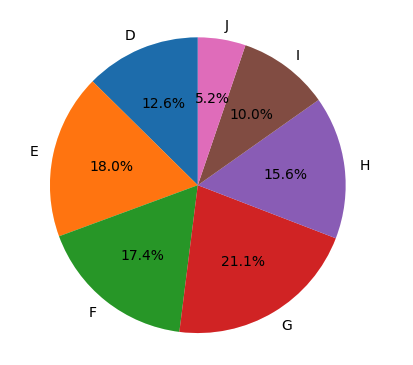
The diamonds were classified into five cut types as indicated in Figure 1 below. Most of the diamonds (40%) fall within the ‘Ideal’ cut type while 2.9% of the diamonds can be described as fair.



**Figure 1:** Distribution of diamond according to cut types.

**Color**

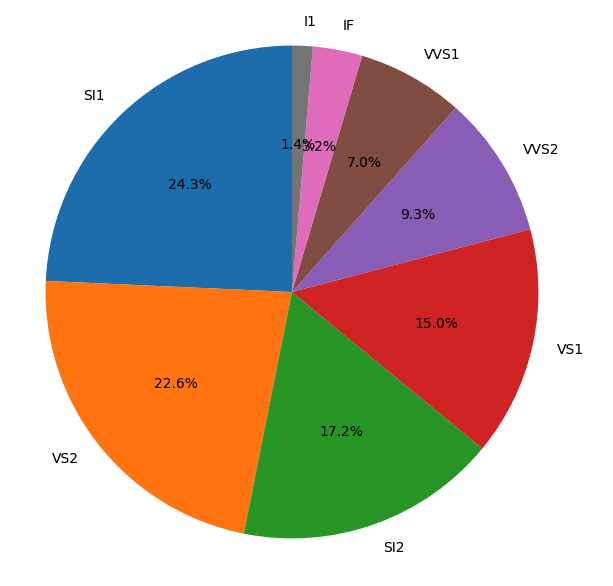
The diamonds were classified into seven colors, see Figure 2 below. Majority (21.1%) of the diamonds belonged to color ‘G’ while only 5.2% belonged to color ‘J’.



**Figure 2:** Distribution of diamonds according to color

**Clarity**

There were eight clarity categories with VS2 being the most common (22.6%) and I1 the least common (1.4%), Figure 3.

**Figure 3:** Distribution of diamonds according to clarity

### Correlation Analysis

A Pearson correlation matrix was calculated, and the results presented in a heatmap (Figure 4).

A screenshot of a graph

Description automatically generated

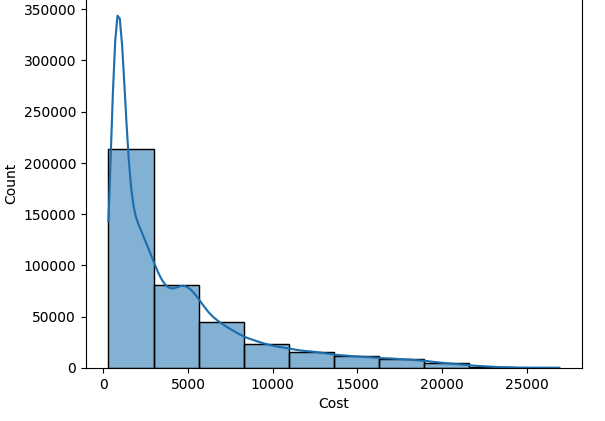
**Figure 4:** Pearson correlation matrix for numeric variables in the diamond dataset

One notable trend is the positive correlation between carat weight and diamond price “cost(dollars)” indicating that heavier diamonds tend to command higher prices. This observation aligns with the traditional understanding of the diamond market, where larger stones are often associated with increased rarity and value. A strong positive relationship was observed between carat and length, width and height of a diamond. Similarly, there was a strong positive correlation between cost (dollars) and length, width and height of a diamond sold. On the other hand, a negative correlation was observed between the table and depth attributes of a diamond.

### Key Trends

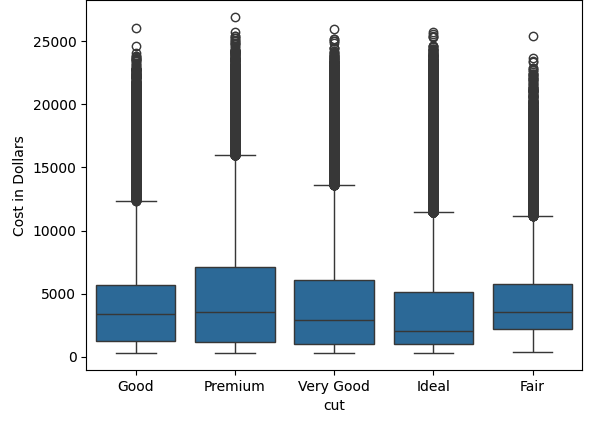
**Price distribution**

Majority of the diamonds sold for less than 5000 dollars. Only a handful sold for above 250,000 dollars, Figure 5, with a price variance of $20,257,077.69 over the ten years. Factors affecting the price of the diamonds included, cut type, color, clarity and year of sale.



**Figure 5:** Diamond price distribution over the ten year sale period

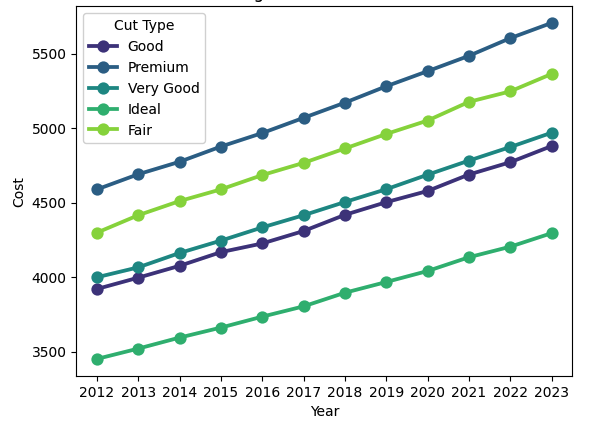
The prices varied across cut types, Generally, the ‘Premium’ cut type cost higher than any other cut type, Figure 6.



**Figure 6:** Average price distribution across cut types.

**Pricing over the years**

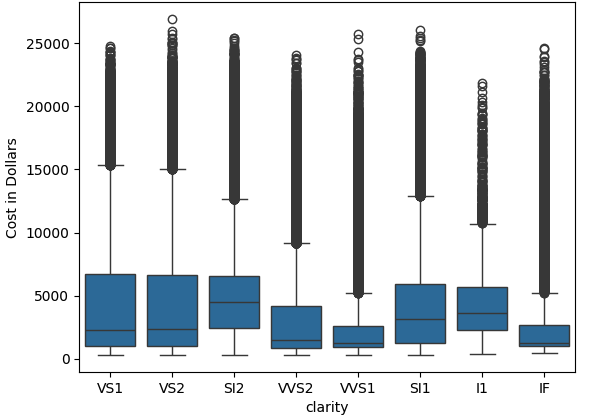
Over the years, diamond prices generally increased for each of the cut types, Figure 7.



**Figure 7:** Trend for diamond prices over the ten-year period.

**Diamond Clarity**

Diamond clarity also affected the price of the diamond with the SI2 category fetching a higher price compared to the rest of the categories, Figure 8.



**Figure 8:** Price variation of diamonds versus clarity types

**Carat distribution**

Most of the diamonds sold in the ten-year period weighed less than 1 carat, Figure 9.

A graph of a number of blue lines

Description automatically generated

**Figure 9:** Carat distribution for diamonds sold between 2012 and 2023.

**Pairwise Clustering of Diamonds**

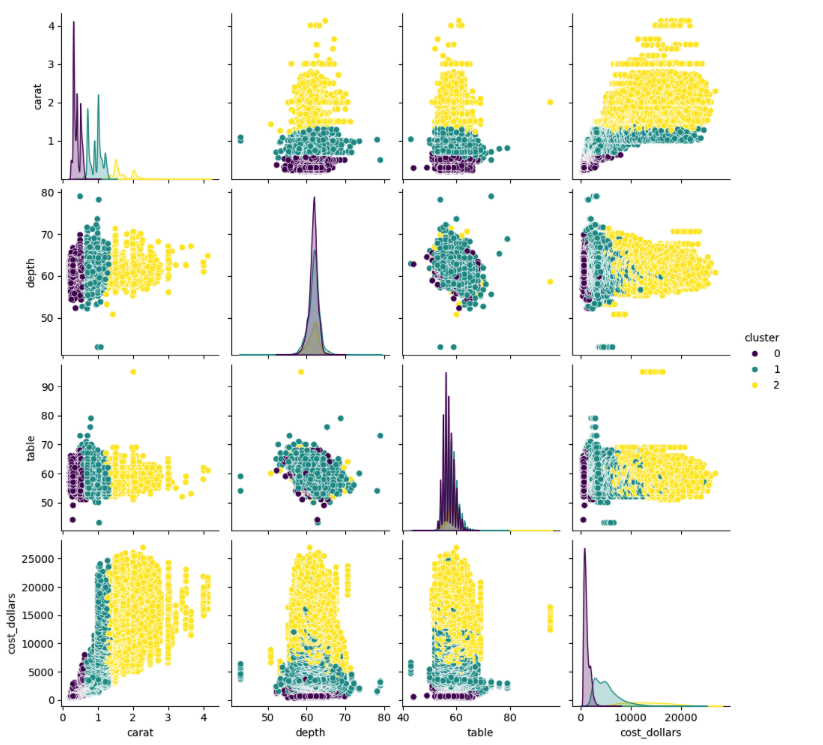
We conducted pairwise clustering to find natural clusters of diamonds considering attributes perceived to have some type of correlation with price as seen below, Figure 10.

Carat vs Cost: There is a positive correlation between the carat of the diamonds and their cost, confirming the expectation that larger diamonds tend to be more expensive.

Depth vs Cost: This relationship is more complex but there seems to be separation based on depth, but this suggests that this cannot be a strong predictor of cost.

Table vs Cost: This relationship is similar to Depth, indicating it is not a good attribute alone to predict cost.

Cluster 2 seems to have a tighter spread within all 3 attributes compared to cost, which could suggest that this group is more similar in physical characteristics.



**Figure 10:** Pairwise plots for Multiple Attributes with natural clusters

# Deliverable 2

# Regression Model Creation and Execution

This report provides an overview and evaluation of five Regressor models developed for predicting the price of diamonds (target variable) in 2024. The model training and validation process involved several key steps aimed at developing accurate and reliable predictive models.

# Data Cleaning

The dataset had been cleaned previously during exploratory data analysis. Consulting with our Alumni lead, Urenna, we realized we missed a few things so we went back and removed the index and year column, and only removed null and 0 columns in the carat column rather than in the dimensions columns.

# Encoding categorical variables

Most machine learning algorithms require numerical input data. The categorical variables were encoded to convert them into numerical format suitable for modeling.

# Train-test split

The choice of train-validation split depends on several factors, including the size of the dataset, the complexity of the model, and the goal of analysis, for example complex models tend to have a higher risk of overfitting, and in such cases, it's important to set aside a larger portion of the data for testing to ensure that the model generalizes well to new data. Similarly, building a predictive model for deployment in the real world may necessitate simulating real-world performance as closely as possible by allocating a larger portion of the data to the test set.

In practice, common train-test split ratios include 70/30, 80/20, or 90/10, where the first number represents the percentage of data allocated to the training set and the second number represents the percentage allocated to the test set. However, there's no one-size-fits-all answer, and the optimal split ratio may vary depending on the factors mentioned above.

The diamond sales dataset was split into training and validation sets using a 70-30 split ratio. All the models were trained on a dataset containing 10000 diamond samples and ten features, i.e carat, cut, color, depth, table, length, width, height, clarity, year. The training set was used to train the model, the validation set was used to tune hyperparameters and evaluate performance during training, and the test set was used to assess the final model's performance.

# Model building

Various machine learning algorithms (Table 1) were employed to train regression models on the training data, including decision tree regression, linear regression, random forest regression, k-nearest neighbors regression, and support vector regression. The respective classification and regression models were built and fitted on the training data by leveraging the respective python packages.

**Table 1:** Machine learning Algorithms used to train the models

| **Model** | **Algorithm used** |
| --- | --- |
| Decision Tree Regression | Decision Tree |
| Linear Regression | Ordinary Least Squares |
| Random Forest Regression | Random Forest |
| Support Vector Regression | Support Vector Machine |
| K Nearest Neighbor Regression | kNN |
| Extreme Gradient Boosting | XGBRegressor |

# Model Performance, Accuracy, and Limitations

Once the models were trained, they were evaluated on the validation set to assess their performance. Evaluation metrics such as mean absolute error (MAE), mean squared error (MSE), and coefficient of determination (R-squared) were used to quantify the models' accuracy and performance. MAE measures the average absolute errors between the actual and predicted values. MSE quantifies the average squared differences between actual and predicted values. Lower MAE and MSE indicate higher accuracy. Additionally, visualization techniques such as histograms of modeling errors were utilized to gain insights into the models' predictive capabilities and identify any potential limitations or areas for improvement.

**Table 2:** Model Performance, Accuracy, Performance and Limitations

| Model | Regression statistics: training | | | Regression statistics: validation | | | Model Limitations |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MAE | MSE | R-squared | MAE | MSE | R-squared |
| K Nearest Neighbor | 305.399 | 282237.539 | 0.986 | 344.639 | 351275.094 | 0.983 | * Sensitive to outliers in the data * May not perform well with a large number of predictors * May not predict well beyond the range of values input in the training data * Choice of k-value can significantly impact predictions |
| Random Forest | 294.99 | 258058.667 | 0.987 | 336.529 | 331733.186 | 0.984 | * Requires more memory than other algorithms because it stores multiple trees. This can be a problem if the dataset is large, just like was the case for the diamond wholesale data * The training time can be longer than other algorithms, especially if the number of trees and the depth of the trees are high * Random Forest can be less interpretable than a single decision tree because it involves multiple trees. It can be difficult to understand how the algorithm arrived at a particular prediction. * Although Random Forest is less prone to overfitting than a single decision tree, it can still overfit the data if the number of trees in the forest is too high or if the trees are too deep |
| Support Vector | 0.0722 | 0.008 | 0.98 | 0.0886 | 0.0115 | 0.97 |  |
| Linear regression | 891.587 | 1889332 | 0.90 | 871.6296 | 1790658.78 | 0.91 | Model easily affected by multicollinearity  Prone to underfitting  Sensitive to outliers  Non-linearity  Constant error variance  Simplistic in some cases  Linearity constraints |
| XGBRegressor (Gradient Boosting) | 421.89 | 541929.847 | 0.973 | 427.157 | 556557.6362 | 0.972 | * Very accurate and Efficient with large data sets * Sensitive to outliers * Efficient for diff data types * Effective for complex relationships * Good with overfitting compared to other models especially when dealing with high dimensional data |

Histogram of modeling errors: Decision Tree Regression

A blue and black graph

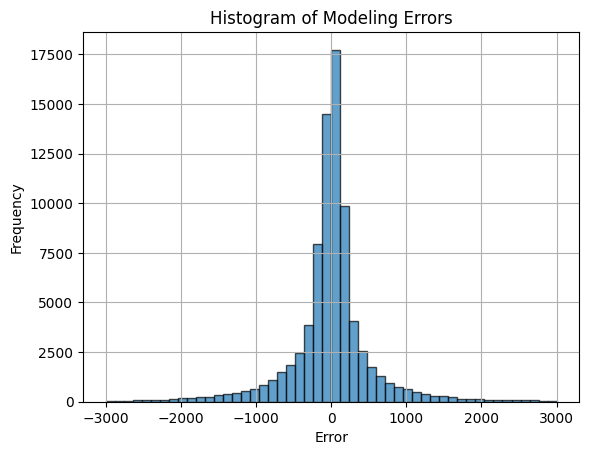
Description automatically generated

Histogram of modeling errors: k Nearest Neighbor

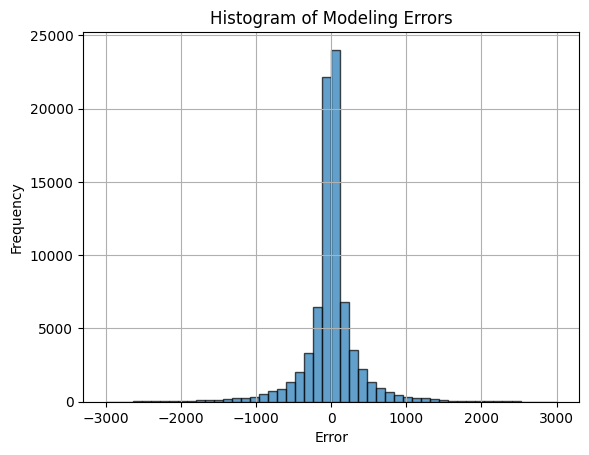
A graph of a normal distribution

Description automatically generated

Histogram of modeling errors: Linear Regression



Histogram for XGB



Histogram for RandomForest

# 

# Deliverable 3

# Introduction

To predict 2024 diamond prices, we developed several predictive models. The first models were built on data that was trained on 70% of the 10 year diamond sales. Because the dataset contained outliers which could cause errors in our models, a meta-model approach was taken. The dataset was clustered into 2 sets, Outliers and Normal Data. The models were trained on 70% on outlier data and 70% on normal data respectively and validated using the remaining 30%. If all the company’s diamonds are sold in 2024, the total sales will be as indicated in Table 1, based on the selected model. In Table 2, we have mapped out the total sales per cut to get an idea on which cut is the most profitable and should be focused on.

### Table 1: Total Predicted Sales and Average Sale Price

| **Model** | **Prediction state** | **Total sales** | **Average diamond price** |
| --- | --- | --- | --- |
| Random Forest | Without Hyperparameter tuning | $139,892,669 | $4,121 |
|  | Based on outlier trained cluster | $390,947,200 | $11,518 |
|  | Based on cluster without outliers | $139,892,669 | $4,121 |
|  | Metamodel (combined prediction based on two clusters) | $265,419,934 | $7,820 |
| LGBM | Without Hyperparameter tuning | $139,063,307 | $4,097 |
|  | Based on outlier trained cluster | $388,653,820 | $11,451 |
|  | Based on cluster without outliers | $139,063,307 | $4,097 |
|  | Metamodel (combined prediction based on two clusters) | $263,858,563 | $7,774 |
| XGB | Without Hyperparameter tuning | $138,996,704 | $4,095 |
|  | Based on outlier trained cluster | $393,830,048 | $11,603 |
|  | Based on cluster without outliers | $138,996,704 | $4,095 |
|  | Metamodel (combined prediction based on two clusters) | $266,413,376 | $7,849 |
| KNN | Without Hyperparameter tuning | $137,478,534 | $4,050 |
|  | Based on outlier trained cluster | $465,417,758 | $13,712 |
|  | Based on cluster without outliers | $137,478,534 | $4,050 |
|  | Metamodel (combined prediction based on two clusters) | $301,448,146 | $8,881 |
| Decision Tree | Without Hyperparameter tuning | $140,015,783 | $4,125 |
|  | Based on outlier trained cluster | $389,640,043 | $11,480 |
|  | Based on cluster without outliers | $140,181,432 | $4,130 |
|  | Metamodel (combined prediction based on two clusters) | $264,910,738 | $7,805 |
| Linear Regression | Without Hyperparameter tuning | $143,733,580 | $4,234 |
|  | Based on outlier trained cluster | $487,811,444 | $14,372 |
|  | Based on cluster without outliers | $143,815,339 | $4,237 |
|  | Metamodel (combined prediction based on two clusters) | $315,813,392 | $9,305 |

### Table 2: Predicted Sales for Each Cut Type

| **Model** | **Prediction state** | **Sales for each Cut Type** | | | | |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Fair** | **Good** | **Ideal** | **Premium** | **Very good** |
| Random Forest | Without Hyperparameter tuning | $4,683,530 | $12,907,915 | $49,281,704 | $40,969,363 | $32,050,155 |
|  | Based on outlier trained cluster | $12,402,131 | $36,537,731 | $148,141,666 | $104,262,180 | $89,603,489 |
|  | Based on cluster without outliers | $4,683,530 | $12,907,915 | $49,281,704 | $40,969,363 | $32,050,155 |
|  | Metamodel (combined prediction based on two clusters) | $8,542,830 | $24,722,823 | $98,711,685 | $72,615,772 | $60,826,822 |
| LGBM | Without Hyperparameter tuning | $4,599,103 | $12,862,761 | $49,028,546 | $40,645,248 | $31,927,647 |
|  | Based on outlier trained cluster | $12,346,799 | $36,852,956 | $146,316,522 | $104,012,108 | $89,125,433 |
|  | Based on cluster without outliers | $4,599,103 | $12,862,761 | $49,028,546 | $40,645,248 | $31,927,647 |
|  | Metamodel (combined prediction based on two clusters) | $8,472,951 | $24,857,858 | $97,672,534 | $72,328,678 | $60,526,540 |
| XGB | Without Hyperparameter tuning | $4,580,829 | $12,857,750 | $48,973,380 | $40,662,768 | $31,921,984 |
|  | Based on outlier trained cluster | $12,412,139 | $37,472,608 | $148,127,024 | $105,277,800 | $90,540,488 |
|  | Based on cluster without outliers | $4,580,829 | $12,857,750 | $48,973,380 | $40,662,768 | $31,921,984 |
|  | Metamodel (combined prediction based on two clusters) | $8,496,484 | $25,165,178 | $98,550,200 | $72,970,280 | $61,231,236 |
| KNN | Without Hyperparameter tuning | $4,478,312 | $12,409,551 | $48,707,819 | $40,319,165 | $31,563,684 |
|  | Based on outlier trained cluster | $9,948,613 | $34,863,842 | $2,068,077,836 | $112,493,276 | $101,304,190 |
|  | Based on cluster without outliers | $4,478,312 | $12,409,551 | $48,707,819 | $40,319,165 | $31,563,684 |
|  | Metamodel (combined prediction based on two clusters) | $7,213,462 | $23,636,697 | $127,757,828 | $76,406,220 | $66,433,937 |
| Decision Tree | Without Hyperparameter tuning | $4,654,058 | $12,919,815 | $49,341,901 | $49,341,901 | $32,104,470 |
|  | Based on outlier trained cluster | $12,505,231 | $36,878,553 | $147,191,947 | $104,267,080 | $88,797,230 |
|  | Based on cluster without outliers | $4,700,033 | $12,919,008 | $49,398,465 | $41,032,096 | $32,131,828 |
|  | Metamodel (combined prediction based on two clusters) | $8,602,632 | $24,898,780 | $98,295,206 | $72,649,588 | $60,464,529 |
| Linear Regression | Without Hyperparameter tuning | $5,137,585 | $13,331,399 | $49,878,808 | $42,602,105 | $32,783,681 |
|  | Based on outlier trained cluster | $14,749,815 | $41,087,048 | $191,284,551 | $130,419,037 | $110,270,992 |
|  | Based on cluster without outliers | $51,370 | $13,341,313 | $49,960,036 | $42,637,279 | $32,739,687 |
|  | Metamodel (combined prediction based on two clusters) | $9,943,418 | $27,214,180 | $120,622,294 | $86,528,158 | $71,505,340 |

# Conclusion

Our predictive analysis for 2024 diamond sales highlights significant insights:  
The metamodel approach projects an optimal blend of accuracy and insight, with Random Forest and XGBoost models indicating a notable increase in average prices and total sales. The analysis suggests a strong market preference for Ideal and Premium cuts, pointing towards higher value transactions. We suggest the company should focus on these cuts in order to capitalize on potential sales growth in 2024.

In the future, we would like to look at price variation based on year to get an idea on if 2024 would be better performing compared to previous years. This could give the company insight on if there would need to be some type of measures implemented to boost sales and stay on track.

# Deliverable 4

# Experience Summaries

Anmar - This Tech.Dive project was a solid chance for me to see what a data role is like. I enjoyed teaming up to tackle our challenges and learned a lot from the experience and my teammates. I had my first introduction to data visualization platforms and feel like I have a grasp of Tableau now. I’m excited to apply these skills at a company and make a meaningful impact with my work.

Jose - I enjoyed the chance to learn about the unique skills and abilities of all the players in our project. Their diverse range of knowledge and expertise came in handy when completing the project. It was evident that each individual brought a unique and valuable perspective to the table.

Amy- I had a great learning experience from my group and I really want to emphasize the learning portion of it. Being that my expertise is more focused in PowerBI, this project taught me to lean on my group and plenty of clarifying questions from them. With all that being said, I had a great time learning more about python and its many uses whilst being able to encourage my group to keep going when things were looking rough.

Rhoda

I enjoyed getting feedback from the Alumni lead, the Tech.Dive leaders and my mentor regarding the various tasks for the assignment.