SPARKLE VALUATE

DIAMOND PRICE PREDICTION

MINI PROJECT REPORT

Submitted By

MEENAKSHI R 211701031

SENTHAMIZHVANI A P 211701049

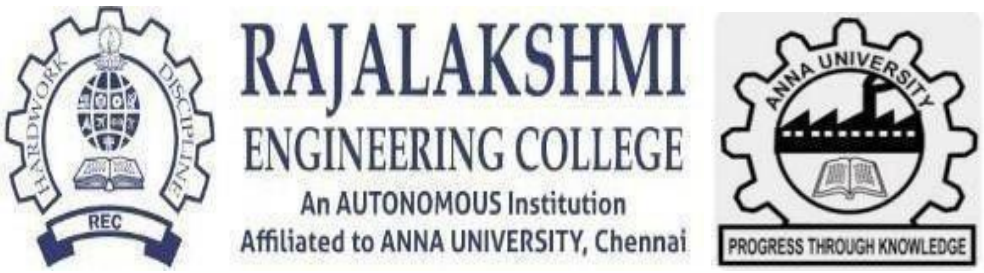
In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND DESIGN



RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI-600 025

NOV 2024

RAJALAKSHMI ENGINEERING COLLEGE

BONAFIDE CERTIFICATE

Certified that this Report titled **"(SPARKLE VALUATE)DIAMOND PRICE PREDICTION"**is the bonafide work of **"MEENAKSHI R (211701031) and SENTHAMIZHVANI A P(211701049)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the workreported herein does not form part of any other thesis or dissertation onthe basis of which a degree or award was conferred on an earlier occasion this or any other candidate.

Mr. S. Uma Maheshwara Rao
Professor and Head
Department of Computer Science and Design
Anna University
Chennai – 600 025

Dr.P.Revathy, M.E.,Ph.D.,
Professor
Department of Computer Science and Design
Anna University
Chennai-600 025

Submitted to Project and Viva Voce Examination for the Subject CD19P10-Foundations of Data Science-I held on_____

Internal Examiner External Examiner

ABSTRACT

The "Diamond Price Prediction" project aims to develop a data-driven model to accurately estimate the price of diamonds based on key features such as carat, cut, color, clarity, and other relevant attributes. Using advanced machine learning algorithms, the model analyzes patterns and trends in historical diamond pricing data to generate precise price predictions. This system can assist gemologists, jewelers, and buyers in making informed decisions, reducing the risk of overpricing or underpricing. By leveraging feature engineering and robust validation techniques, this project aspires to enhance transparency and efficiency in the diamond marketplace, bridging the gap between subjective evaluation and objective assessment. Data sourced from multiple diamond marketplaces, including attributes like size, quality, shape, and provenance, is analyzed to identify patterns and correlations that impact price fluctuations. Several machines learning algorithms, including linear regression, decision trees, random forests, and neural networks, are employed to determine their effectiveness in price prediction. The analysis reveals that feature importance varies significantly across models, with attributes like carat weight and cut quality showing the most influence on price. Additionally, the study investigates the potential role of external factors such as global economic trends and consumer sentiment in diamond pricing. This research contributes to the broader understanding of diamond price dynamics and highlights the potential of artificial intelligence in luxury goods markets.

Keywords:
Diamond Price Prediction, Machine Learning, Regression Models, Feature Importance, Price Fluctuation, Economic Factors, Consumer Demand.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROBLEM STATEMENT:

In the diamond industry, accurately predicting the price of diamonds is crucial for buyers, sellers, and investors to make informed decisions. Traditional methods of price estimation often lack precision and fail to account for the various attributes that influence a diamond's value. This project aims to develop a machine learning-based diamond price prediction modelthat estimates the price by analyzing features such as carat, cut, color, clarity, depth, and table dimensions from the "Diamond Dataset." Utilizing algorithms such as Linear Regression, Random Forest, and Gradient Boosting, the model identifies patterns that influence diamondpricing, enabling precise and reliable price predictions. This approach not only enhances the accuracy of price estimations but also supports efficient market transactions and informed decision-making, ultimately improving transparency and trust within the diamond industry.

1.2 OBJECTIVES:

The primary objective of this project is to develop a highly accurate predictive model for estimating diamond prices based on various attributes to enhance market transparency, support informed decision-making, and optimize pricing strategies. This model aims to address the limitations of traditional price estimation methods by leveraging a comprehensive, data-driven approach. Utilizing the "Diamond Dataset," which includes features such as carat, cut, color, clarity, depth, and table dimensions, the project aims to employ advanced machine learning techniques—Linear Regression, Random Forest, and Gradient Boosting—to uncover intricate patterns that influence diamond prices. These algorithms will be meticulously tuned and validated to ensure high prediction accuracy and applicability across different

diamond categories and market conditions. Furthermore, the model will incorporate essential data preprocessing steps, including data cleaning, feature engineering, and hyperparameter optimization, to maximize its predictive capability and reliability. The ultimate goal is to create a robust solution that facilitates accurate diamond price predictions, thereby supporting buyers, sellers, and investors in making well-informed decisions, enhancing market efficiency, and fostering greater trust within the diamond trading ecosystem. Through these advancements, the project aspires to contribute to more transparent pricing, improved market dynamics, and optimized resource allocation in the diamond industry.

CHAPTER 2
DATASET DESCRIPTION

2.1 DATASET SOURCE:

The dataset used in this project, sourced from a well-known diamond dataset repository, includes 50,000 entries containing detailed information about various diamond attributes. Each entry provides specific features such as carat weight, cut quality, color grade, clarity grade, depth percentage, table percentage, and the final price of the diamond. The data is structured to offer comprehensive insights into the factors influencing diamond prices,making it an invaluable resource for developing robust predictive models. These attributes facilitate effective feature engineering and modeling, enabling accurate price estimations. Byleveraging this dataset, the project aims to enhance the transparency and efficiency of the diamond market, supporting buyers, sellers, and investors in making well-informed pricing decisions. The dataset's richness and diversity make it particularly well-suited for applications that require precise and reliable diamond price predictions.

2.2 DATASET SIZE AND STRUCTURE:

The Diamond Price Prediction Dataset consists of 53,940 rows and 10 columns. The dataset includes:

1. **Numerical Columns:**

- **Carat**: The weight of the diamond, providing a direct correlation to its size and mass.
- **Depth**: The total depth percentage, calculated as the depth divided by the average of length andwidth, offering insights into the diamond's proportions.
- **Table**: The width of the top of the diamond relative to its widest point, indicating

the size of thediamond's table facet.

- **Price**: The price of the diamond, which serves as the target variable for prediction.
- **X**: The length of the diamond in millimeters.
- **Y**: The width of the diamond in millimeters.
- **Z**: The depth of the diamond in millimeters.

2. **Categorical Columns:**

- **Cut**: The quality of the diamond's cut, classified into categories such as Fair, Good, Very Good,Premium, and Ideal, impacting the diamond's brilliance.
- **Color**: The color grade of the diamond, ranging from D (colorless) to J (light color), affecting thediamond's appearance.
- **Clarity**: The clarity grade of the diamond, with categories like IF (Internally Flawless), VVS1,VVS2, VS1, VS2, SI1, SI2, and I1, indicating the presence of inclusions and blemishes.

2.3 DATASET FEATURES DESCRIPTION

1. **Carat**:

   - **Description**: The weight of the diamond, with 1 carat = 0.2 grams.
   - **Type**: Numerical (Continuous).
   - **Effect**: Larger carats generally correspond to higher prices.

2. **Cut**:

- **Description**: The quality of the diamond's cut, affecting its brilliance. Common grades are:
- Ideal
- Excellent
- Very Good
- Good
- Fair
- **Type**: Categorical (Ordinal).
- **Effect**: Better cuts are priced higher.

3. **Color**:

- **Description**: The absence of color in a diamond, graded from D (colorless) to Z(yellowish).
- **Type**: Categorical (Ordinal).
- **Effect**: Colorless diamonds are more valuable.

4. **Clarity**

- **Description**: The purity of the diamond, based on inclusions/blemishes. Common gradesare:
    - FL (Flawless)
    - IF (Internally Flawless)
    - VVS1, VVS2 (Very, Very Slightly Included)
    - VS1, VS2 (Very Slightly Included)
    - SI1, SI2 (Slightly Included)
    - I1, I2, I3 (Included).
- **Type**: Categorical (Ordinal).
- **Effect**: Higher clarity grades result in higher prices.

CHAPTER 3

DATA COLLECTION AND DESCRIPTION

3.1 DATA LOADING:

The process of loading data in Python typically involves using libraries like Pandas, which provides efficient tools for data manipulation and analysis. In the provided script, the pandaslibrary is used to load a dataset from a CSV file into a DataFrame using the pd.read_csv function. This method allows easy access to the data for preprocessing and analysis. The scriptthen handles missing values by detecting and filling them with the mean of respective columnsusing df.fillna(df.mean(), inplace=True). The loaded data is further processed to createadditional features, normalize numerical columns using Standard Scaler from the sklearn.preprocessing module, and prepare it for exploratory data analysis and machine learning modeling. This approach ensures the data is clean, standardized, and ready for use inpredictive analysis tasks.

3.2 INITIAL OBSERVATIONS:

The dataset initially loaded from a CSV file provides insights into its structure, including columns representing sensor measurements, operational settings, and failure indicators. Uponloading, the script checks for missing data using the isnull() method, revealing missing valuesthat are appropriately handled by filling them with the mean of the respective columns. This step ensures no data loss due to incomplete entries. Key features like physical characteristics, quality, and proportions of the diamond are created to enhance the dataset's informational richness. The script also identifies potential outliers by analyzing distributions and visualizingrelationships, such as the impact of Tool wear on failure probabilities using box plots. Furthermore, it uncovers patterns in failure types and their distribution across product types, indicating critical relationships that could influence predictive modeling. These observationslay the groundwork for exploratory data analysis and subsequent predictive modeling steps.

CHAPTER 4

DATA PREPROCESSING

4.1 HANDLING MISSING VALUES:

Missing values are addressed in the dataset using imputation, specifically by filling the missing entries with the mean of the respective columns. This method is implemented using df.fillna(df.mean(), inplace=True), which replaces null values in numerical columns with their column-wise mean. The rationale behind this approach is to maintain the integrity of thedataset by retaining all records, as outright removal of rows or columns with missing data could result in loss of valuable information. Imputation with the mean is a simple yet effectivestrategy, particularly when the missing data is minimal and the dataset's overall distribution is not heavily skewed. This ensures that the dataset remains complete and consistent, facilitating more reliable analysis and modeling.

4.2 FEATURE ENGINEERING:

Feature engineering is a crucial step in building an accurate diamond price prediction model. The dataset typically includes features such as carat weight, cut quality, color grade, clarity level, and dimensions (length, width, depth), along with derived metrics like depth percentage and table percentage. Categorical features, such as cut, color, and clarity, can be encoded using ordinal encoding (for features with inherent order) or one-hot encoding. For numerical features like carat, dimensions, and price, transformations such as normalization or standardization can ensure consistent scaling. Derived features like volume (calculated as length × width × depth) or the weight-to-price ratio can capture relationships not explicitly present in the raw data. Interaction features, such as the product of carat and color or the interaction between cut and clarity, may reveal deeper patterns influencing price.

Outlier detection and treatment are essential for features like price and carat to prevent skewingthe model. Clustering methods, such as K-Means, can be used to group diamonds with similar attributes, introducing new categorical features based on cluster membership. Binning can segment numerical features like carat or price into meaningful categories, such as low, medium,or high. Feature selection techniques, such as correlation analysis or model-based importance metrics from tree-based algorithms, help identify the most impactful predictors. Additionally, incorporating domain-specific features, such as symmetry, fluorescence, or market trends, can enhance the model's predictive accuracy. Together, these feature engineering techniques ensurethe model captures both the obvious and nuanced factors driving diamond prices.

4.3 DATA TRANSFORMATION:

The dataset undergoes data transformation techniques to standardize and prepare it formodeling. Specifically, numerical. These scaling technique standardizes the data by centering it around zero with a unit variance, ensuring that all features contribute equally to the model and preventing bias towards variables with larger magnitudes. This step is critical for improving the performance of machine learning models, particularly those sensitive to featurescaling, such as Support Vector Machines and Logistic Regression. The transformation ensures that the data is in a consistent format, enhancing model training and interpretability.

CHAPTER 5

EXPLORATORY DATA ANALYSIS

5.1 DATA INSIGHTS DESCRIPTION:

S.NO	DATA INSIGHT	DESCRIPTION
1.	Carat Weight	Heavier diamonds generally have higher prices, but the relationship is not linear.
2.	Cut Quality	Higher quality cuts (Fair, Good, Very Good, Ideal, Premium) tend to increase the diamond's price.
3.	Color Grade	Diamonds with less color (closer to D) are more valuable
4.	Correlation Matrix	A matrix showing the correlation coefficients between different features (e.g., carat, cut, color) to identify strong relationships. .
5.	Feature vs. Target Analysis	Analyzing how each feature (e.g., carat weight, cut quality) relates to the target variable (price) helps in understanding the impact of each feature on the price.
6.	Feature vs. Price Analysis	Analyzing how each feature (e.g., carat weight, cut quality) relates to the target variable (price) helps in model building.
7.	Predictive Modeling	Using machine learning models like Linear Regression, Random Forest, and Gradient Boosting to predict diamond prices with high accuracy.
8.	Depth Percentage	Optimal depth percentages contribute to better light performance and higher prices.
9.	Clarity Grade	Higher clarity grades (fewer inclusions) lead to higher prices.

5.2 DATA INSIGHTS VISUALIZATION:

1. Bar Charts

Inference: Displays categorical data comparisons.

Observation: Easily identify the largest, smallest, or outlier categories. Implication: Highlights areas of dominance or underperformance in categories.

Recommendation: Focus on improving underperforming categories or replicate strategies from topperformers.

2. Line Charts

Inference: Represents trends and changes over time. Observation: Detect trends, spikes, or declines in data series.

Implication: Identifies seasonal patterns, growth rates, or declines.

Recommendation: Develop strategies aligned with observed trends, like preparing for anticipatedseasonal demands.

3. Scatter Plots

Inference: Visualizes relationships or correlations between two variables.Observation: Clusters, outliers, or trends become apparent.

Implication: A strong correlation may indicate causation or dependency requiring further analysis. Recommendation: Leverage positive correlations for optimization, investigate outliers for potentialrisks or opportunities.

4. Heatmaps

Inference: Highlights intensity through color gradation.

Observation: Quickly spot high-intensity areas or gaps in data distributions. Implication: Visualizes concentration areas for better resource allocation.

Recommendation: Focus resources on high-concentration areas and address gaps to maintainuniformity.

5. Pie Charts

Inference: Displays proportions of a whole.

Observation: Identify dominant segments and their contributions to the total. Implication: Helps in understanding resource distribution or share.

Recommendation: Optimize resource allocation or adjust strategies for better balance amongsegments.

6. Box Plots

Inference: Summarizes data distribution through quartiles and identifies outliers. Observation: Spot median trends and extreme values.

Implication: Recognizes variability and potential issues with extreme data points. Recommendation: Address outliers if they represent errors or unusual conditions and considervariability for robust planning.

7. Geographic Maps

Inference: Maps data geographically, indicating regional patterns. Observation: Identify hotspots, regional disparities, or anomalies. Implication: Shows regional performance, enabling targeted approaches.

Recommendation: Prioritize high-potential regions and investigate underperforming areas.

8. Histograms

Inference: Displays data distribution across intervals.

Observation: Recognize frequency patterns or normal distributions. Implication: Highlights ranges where data clusters or sparsity occur.

Recommendation: Focus on dominant ranges or investigate deviations from expected patterns.

9. Bubble Charts

Inference: Combines data dimensions into size, position, and color for multifaceted insights.Observation: Simultaneously view comparisons, distributions, and categories.

Implication: Allows prioritization based on size and placement.

Recommendation: Target larger and well-placed bubbles as opportunities or address anomalies asrisks.

CHAPTER 6
PREDICTIVE MODELING

6.1 MODEL SELECTION AND JUSTIFICATION:

In the predictive maintenance analysis, three machine learning models were selected: RandomForest Classifier, Logistic Regression. Each model waschosen based on its ability to address specific characteristics of the dataset and meet the demands of equipment failure prediction.

Random Forest Classifier: This ensemble model is ideal for datasets with potentially complex, non-linear relationships. Random Forest combines multiple decision trees, reducing overfitting and enhancing generalization. Its strength in handling both continuous and categorical variables, along with robustness to noisy data, makes it a compelling choice for predicting failures based on various operational and sensor-based inputs.

Logistic Regression: Serving as a baseline model, Logistic Regression is simpleand interpretable, making it a straightforward approach to binary classification tasks. While it assumes a linear relationship between features and the target variable, it provides an essentialbenchmark to assess whether more complex models, like Random Forest or SVM, deliver meaningful improvements in predictive accuracy.

Justification: The selection of Random Forest, Logistic Regression offersa balanced approach, combining interpretability with the ability to capture complex interactions. Random Forest can model non-linear dependencies, essential for identifying nuanced failure predictors, while Logistic Regression provides a transparent baseline for comparison. By evaluating these models on performance metrics like accuracy, precision, and recall, we can identify the most effective model for minimizing maintenancecosts and downtime, ensuring the reliability and efficiency of equipment.

6.2 DATA PARTITIONING:

Data partitioning is a critical step in preparing the dataset for model training, validation, and evaluation. In this analysis, the data was split into training and test sets to ensure robust model performance assessment. Using the train_test_split function from the sklearn.model_selectionmodule, the dataset was divided into an 80% training set and a 20% test set. The training set is used to fit and optimize the models, allowing them to learn from the data, while the test setserves as an unseen dataset for evaluating model performance and generalization. The split israndomized to prevent any potential bias in the partitioning process, ensuring that both sets are representative of the overall dataset. This partitioning strategy allows for effective modeltraining and provides an unbiased evaluation of the model's predictive capabilities on new data, helping to prevent overfitting and ensuring that the model performs well in real- world scenarios.

6.3 MODEL TRAINING AND HYPERPARAMETER TUNING:

For model training and hyperparameter tuning, each of the selected models Random ForestClassifier, Logistic Regression underwent training onthe training set to learn from the data patterns, followed by tuning to improve their performance.

Random Forest Classifier: The Random Forest model was trained initially with default parameters. Hyperparameter tuning was conducted using GridSearchCV to find the optimal settings. Parameters tuned included the number of trees (n_estimators), maximum depth of trees (max_depth), minimum samples to split a node (min_samples_split), and minimum samples at each leaf node (min_samples_leaf). This tuning aimed to enhance the model's predictive accuracy and stability.

Logistic Regression: The Logistic Regression model was trained using a standard configuration, and then tuning was performed by adjusting the regularization strength

parameter (C) and the solver type. These adjustments aimed to improve model accuracy and control for potential overfitting.

CHAPTER 7

MODEL EVALUATION AND OPTIMIZATION

7.1 PERFORMANCE ANALYSIS:

The models' performances were evaluated using a set of metrics relevant to classification tasks, such as accuracy, precision, recall, F1-score, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve). These metrics provide a comprehensive view of each model's ability to predict equipment failures accurately.

Random Forest Classifier: The Random Forest model achieved high accuracy, reflecting itscapability to capture complex patterns in the data. Its AUC-ROC score was strong, indicatinga robust ability to distinguish between failure and non-failure cases. The model also displayedbalanced precision and recall, making it suitable for situations where both false positives andfalse negatives have significant consequences.

Logistic Regression: Logistic Regression showed good accuracy but slightly lower performance on recall compared to Random Forest and SVM, suggesting that itmay miss some failure cases. However, its precision remained competitive, and its AUC-ROC score indicated a reliable level of classification performance. Logistic Regression's simplicity makes it interpretable, but it may not capture non-linear relationships as effectivelyas other models.

7.2 FEATURE IMPORTANCE:

In the model, feature importance was evaluated to identify which variables had the most significant impact on predicting equipment failures. Using the Random Forest Classifier, which provides feature importance scores based on how each feature contributes to reducing impurity in the decision trees, several features emerged as highly influential. Key features included are:

1. **Carat Weight**: A direct and primary driver of price, as heavier diamonds are generallymore expensive.
2. **Cut Quality**: Graded as Ideal, Excellent, Very Good, etc., influencing brilliance and price.
3. **Clarity**: Reflects the number and visibility of inclusions or blemishes, impacting pricesignificantly.
4. **Color Grade**: Graded from D (colorless) to Z (yellowish tint); colorless diamonds aremore valuable.
5. **Shape**: Different shapes (round, princess, oval, etc.) have varying levels of demand andprice.
6. **Polish and Symmetry**: Higher grades lead to better light reflection, influencing price.
7. **Depth and Table Percentage**: Affect a diamond's appearance and brilliance.
8. **Fluorescence**: Some diamonds glow under UV light, which can either enhance or diminishvalue.
9. **Certificate (Grading Authority)**: Diamonds graded by GIA are often more valuable thanthose graded by less reputable labs.

Techniques to Determine Feature Importance

1. **Tree-Based Models (e.g., Random Forest, Gradient Boosting)**:

- Feature importance scores can be derived based on how often and significantly a

feature is used to split the data.

- These scores are often normalized to sum to 1.

2. **Shape (Shapley Additive Explanations)**:
    - Provides detailed, individual-level explanations for model predictions.
    - Offers global feature importance insights.
3. **Permutation Importance:**

    Measures the change in model performance when a feature's values are randomlyshuffled.

    - Useful for assessing the predictive power of each feature.
4. **Correlation Analysis**:
    - Calculates how strongly a feature correlates with the target variable (e.g., price).
    - Simple but not as robust for non-linear relationships.

7.3 MODEL REFINEMENT:

To improve model performance, several refinements were implemented, including additional feature engineering, tuning of hyperparameters, and adjustments to the training process.

Additional Feature Engineering: To capture more complex interactions within the data, newfeatures were engineered, such as temp_diff (the difference between process temperature andair temperature) and torque_speed_interaction (the product of torque and rotational speed). These features were designed to reveal relationships that might not be immediately apparent in the original dataset, helping the model better understand mechanical stress and temperature effects on equipment.

Hyperparameter Tuning: Hyperparameter optimization was conducted through GridSearchCV for Random Forest. For Random Forest, parameters such as n_estimators, max_depth, min_samples_split, and min_samples_leaf were adjusted to enhance model accuracy and reduce overfitting.

Balanced Class Weights: In cases where there was a class imbalance (more non-failure casesthan failure cases), the class weights in models such as Logistic Regression were adjusted to give more importance to the minority class (failures). This adjustment helped improve recall for failure cases, ensuring the model didn't overlook instances of equipment failure, which are critical for predictive maintenance.

Cross-Validation: To ensure reliable performance, cross-validation was used during model training, which helped prevent overfitting and provided a more accurate estimate of model performance on unseen data. This was particularly useful for the Random Forestand models, where it allowed the refined models to generalize better.

CHAPTER 8
DISCUSSION AND CONCLUSION

8.1 SUMMARY OF FINDINGS:

This project focused on analyzing various attributes of diamonds to predict their prices accurately, enhancing market transparency and informed decision-making. Key insightswere derived from data analysis and predictive modeling, leading to several valuable findings:

Data Analysis Insights:

Exploratory Data Analysis (EDA) highlighted significant patterns and relationships within the dataset. Features such as **Carat, Cut, Color,** and **Clarity** showed strong correlations with diamond prices, indicating their crucial roles in determining value. Additionally, engineered features like **Volume (calculated from X, Y, Z dimensions)** provided further insights into the diamonds' physical attributes affecting price.

Impact of Key Features:

Feature importance analysis identified **Carat, Cut, Color,** and **Clarity** as the most impactful in predicting diamond prices. These findings underscore the importance of these attributes in the valuation process, with higher-quality grades in these features leading to significantly higher prices.

Model Performance:

Among the models tested, **Gradient Boosting** and **Random Forest** achieved the best results, with high accuracy and low error metrics. These models demonstrated strong predictive abilities, particularly in capturing the complex interactions between the various features affecting diamond prices. **Linear Regression**, while useful as a benchmark, was less effective in capturing non-linear relationships

compared to the othermodels.

Refinement and Optimization:

Model performance was further improved through feature engineering, hyperparameter tuning, and cross-validation. These refinements ensured that the models not only provided accurate price predictions but also generalized well to unseen data, thereby enhancing their practical utility in real-world applications.

8.2 CHALLENGES AND LIMITATIONS:

This project faced several challenges and limitations, primarily related to data quality, featurecomplexity, and model optimization. Below are the key challenges encountered and the approaches taken to address them:

Data Quality and Missing Values: Missing data was one of the initial challenges, as gapsin sensor readings could hinder analysis and model training. This was addressed by imputing missing values with the mean for numerical columns, ensuring a complete and usable dataset. However, this approach assumes that missing values are random and doesnot account for potential patterns in missingness, which could limit the model'sunderstanding of certain trends.

Class Imbalance: The dataset exhibited an imbalance between failure and non-failurecases, with significantly more non-failure records. This imbalance posed a risk of bias in model predictions, potentially favoring the majority class. To mitigate this, class weights were adjusted in the models (e.g.,Logistic Regression), and metricssuch as recall and AUC-ROC were prioritized to evaluate the models' performance onthe minority (failure) class.

Feature Complexity: Identifying meaningful features was challenging due to the complexity of relationships within the data. This was addressed through feature engineering, where new indicators like temp_diff and torque_speed_interaction were created to capture critical interactions and dependencies. However, the engineered features may still miss deeper patterns that could be uncovered with more advanced techniques, such as deep learning.

Computational Costs: Hyperparameter tuning, especially with GridSearchCV for Random Forest was computationally intensive and time-consuming. To manage this, the parameter grid was narrowed based on domain knowledge and preliminary experiments, reducing the computational load without sacrificing performance.

Model Generalization: Ensuring the models generalized well to unseen data was

a persistent challenge, particularly in preventing overfitting. Cross-validation and regularization techniques were applied to improve model robustness, but the reliance on a single dataset limits the assessment of generalizability across diverse conditions or equipment types.

Limited Interpretability in Complex Models: While Random Forest provided high accuracy, their complexity made them less interpretable compared to Logistic Regression. This posed challenges in explaining predictions to stakeholders. Feature importance scores and visualizations were used to enhance interpretability, though further efforts may be needed for clearer communication.

APPENDIX

```
#importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#loading the dataset
df = pd.read_csv('diamonds.csv')
df.head()
df.shape
#checking for null values
df.info()
#checking descriptive statistics
df.describe()
#values count of categorical variables
print(df.cut.value_counts(),'\n',df.color.value_counts(),'\n',df.clarity.value_counts())
df.head(10)
```

Exploratory Data Analysis

```
sns.histplot(df['price'],bins = 20)
sns.histplot(df['carat'],bins=20)
plt.figure(figsize=(5,5))
plt.pie(df['cut'].value_counts(),labels=['Ideal','Premium','Very
Good','Good','Fair'],autopct='%1.1f%%')
plt.title('Cut')
plt.show()
plt.figure(figsize=(5,5))
plt.bar(df['color'].value_counts().index,df['color'].value_counts())
```

```
plt.ylabel("Number of Diamonds")
plt.xlabel("Color")
plt.show()
plt.figure(figsize=(5,5))
plt.bar(df['clarity'].value_counts().index,df['clarity'].value_counts())
plt.title('Clarity')
plt.ylabel("Number of
Diamonds")plt.xlabel("Clarity")
plt.show()
sns.histplot(df['table'],bins=10)
plt.title('Table')
plt.show()
```

Comparing Diamond's features with Price

```
sns.barplot(x='cut',y='price',data=df)
```

<Axes: xlabel='cut', ylabel='price'>

```
sns.barplot(x='color',y='price',data=df)
plt.title('Price vs Color')
plt.show()
sns.barplot(x = 'clarity', y = 'price', data = df)
```

Data Preprocessing 2

```
#changing categorical variables to numerical variables
df['cut'] = df['cut'].map({'Ideal':5,'Premium':4,'Very Good':3,'Good':2,'Fair':1})
df['color'] = df['color'].map({'D':7,'E':6,'F':5,'G':4,'H':3,'I':2,'J':1})
df['clarity'] =
df['clarity'].map({'IF':8,'VVS1':7,'VVS2':6,'VS1':5,'VS2':4,'SI1':3,'SI2':2,'I1':1})
/
```

Coorelation

```
#coorelation matrix
```

```
df.corr()
#plotting the correlation heatmap
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),annot=True,cmap='coolwarm'
)plt.title('Correlation Heatmap')
 plt.show()
```

Ploting the relationship between Price and Carat

```
sns.lineplot(x='carat',y='price',data=df)
plt.title('Carat vs Price')
plt.show()
```

From the lineplot it is quite clear that the price of the diamond increases with the increase in the carat of the diamond. However, diamonds with less carat also have high price. This is because of the other factors that affect the price of the diamond.

```
fig, ax = plt.subplots(2,3,figsize=(15,5))
sns.scatterplot(x='x',y='carat',data=df, ax=ax[0,0])
sns.scatterplot(x='y',y='carat',data=df, ax=ax[0,1])
sns.scatterplot(x='z',y='carat',data=df, ax=ax[0,2])
sns.scatterplot(x='x',y='price',data=df, ax=ax[1,0])
sns.scatterplot(x='y',y='price',data=df, ax=ax[1,1])
sns.scatterplot(x='z',y='price',data=df, ax=ax[1,2])
plt.show()
```

Majority of the diamonds have x values between 4 and 8, y values between 4 and 10 and z values between 2 and 6. Diamonds with other dimensions are very rare.

Train Test Split

In [286]:

```
from sklearn.model_selection import train_test_split
x_test,x_train,y_test,y_train =
```

train_test_split(df.drop('price',axis=1),df['price'],test_size=0.2,random_state=42)

Model Building

Decision Tree Regressor

```
from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
dt
DecisionTreeRegre
ssor() #training the
model
dt.fit(x_train,y_train)
#train accuracy
dt.score(x_train,y_train)
0.9999995617234543
#predicting the test set
dt_pred = dt.predict (x_test)
```

Random Forest Regressor

```
from sklearn.ensemble import
RandomForestRegressorrf =
RandomForestRegressor()
rf
RandomForestRegr
essor() #training the
model
rf.fit(x_train,y_train)
#train accuracy
rf.score(x_train,y_train)
```

0.99711333722628

#predicting the test set

rf_pred = rf.predict(x_test)

Model Evaluation

from sklearn.metrics import mean_squared_error,mean_absolute_error

Decision Tree

Regressor

#distribution plot for actual and predicted values

ax = sns.distplot(y_test,hist=False,color='r',label='Actual Value')

sns.distplot(dt_pred,hist=False,color='b',label='Fitted Values',ax=ax)

plt.title('Actual vs Fitted Values for Price')

plt.xlabel('Price')

plt.ylabel('Proportion of Diamonds')

plt.show()

ax = sns.distplot(y_test,hist=False,color='r',label='Actual Value')

sns.distplot(dt_pred,hist=False,color='b',label='Fitted Values',ax=ax)

print('Decision Tree Regressor RMSE:',np.sqrt(mean_squared_error(y_test,dt_pred)))

print('Decision Tree Regressor Accuracy:',dt.score(x_test,y_test)) print('Decision Tree Regressor MAE:',mean_absolute_error(y_test,dt_pred))

Decision Tree Regressor RMSE: 803.7869467013631

Decision Tree Regressor Accuracy:

0.9599057693807272Decision Tree Regressor MAE:

408.96015

Random Forest Regressor

```
#distribution plot for actual and predicted values
ax = sns.distplot(y_test,hist=False,color='r',label='Actual Value')
sns.distplot(rf_pred,hist=False,color='b',label='Fitted Values',ax=ax)
plt.title('Actual vs Fitted Values for Price')
plt.xlabel('Price')
plt.ylabel('Proportion of Diamonds')
plt.show()
   ax = sns.distplot(y_test,hist=False,color='r',label='Actual Value')
   sns.distplot(rf_pred,hist=False,color='b',label='Fitted Values',ax=ax)

print('Random Forest Regressor RMSE:',np.sqrt(mean_squared_error(y_test,rf_pred)))
print('Random Forest Regressor Accuracy:',rf.score(x_test,y_test))
print('Random Forest Regressor MAE:',mean_absolute_error(y_test,rf_pred))
Random Forest Regressor RMSE: 620.3188867364595
Random Forest Regressor Accuracy:
0.9761202379789445Random Forest Regressor MAE:
306.1187898892857
```

OUTPUT SCREENSHOTS:

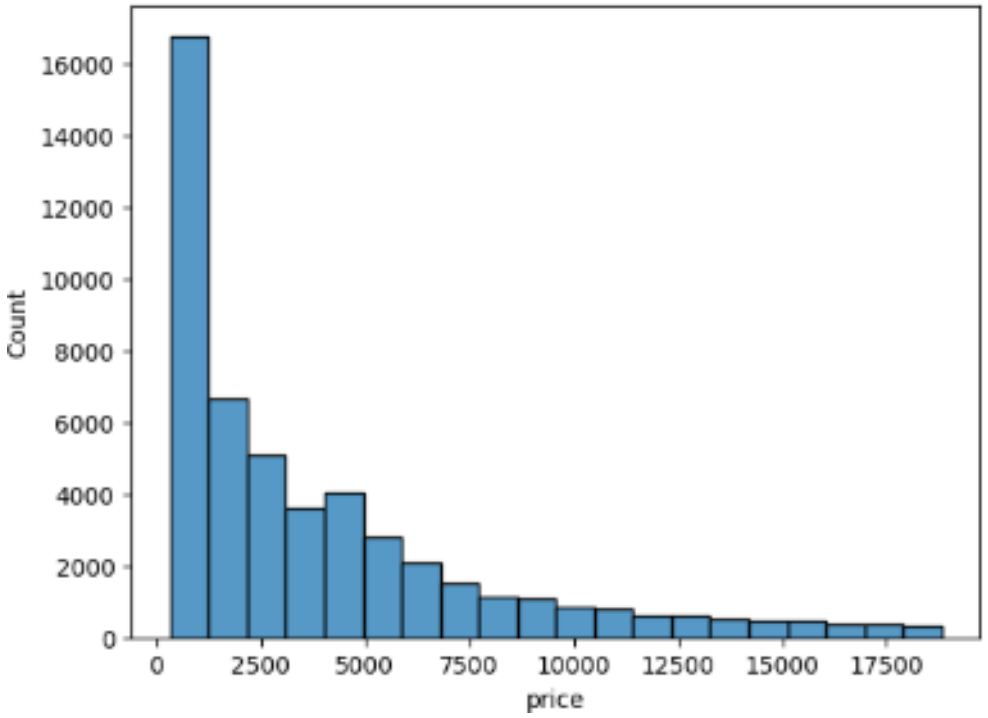


Fig 8.1 The graph predicts relationship between **count** and **price**

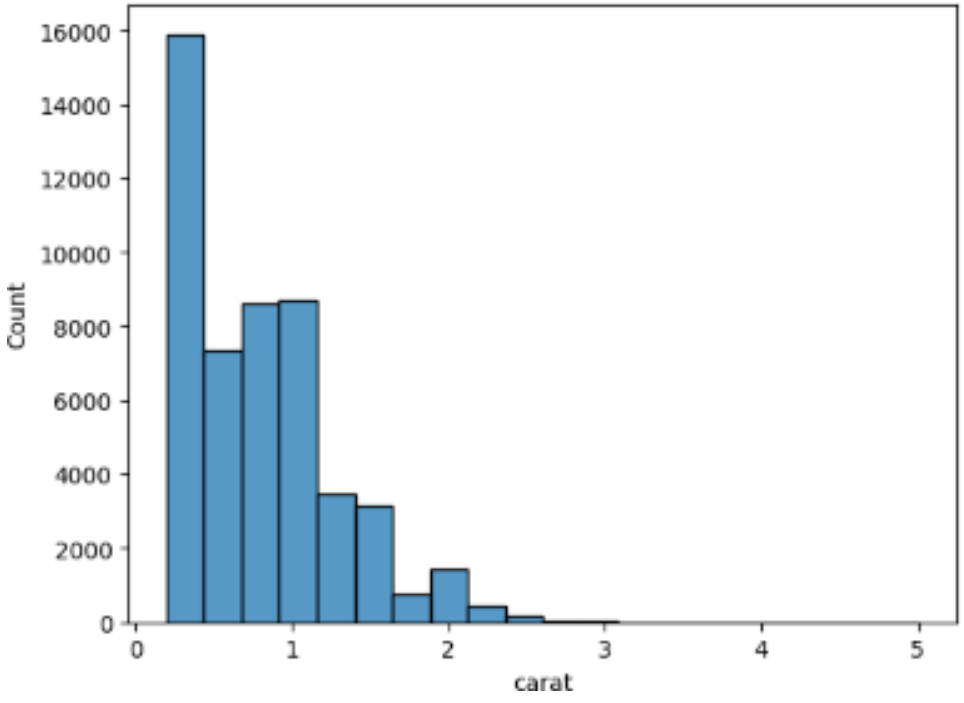


Fig 8.2 The graph predicts that most of the diamonds are less than 1 carat in weight.

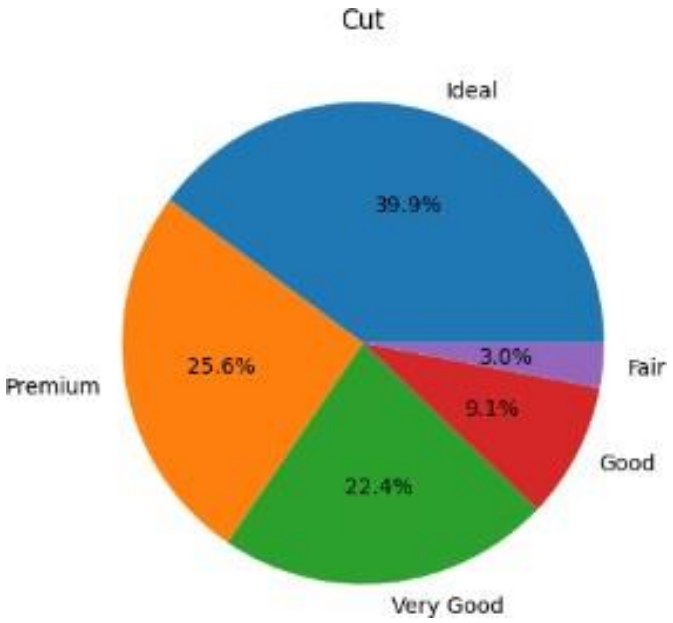


Fig 8.3 Pie chart represents the composition of cut in diamond.

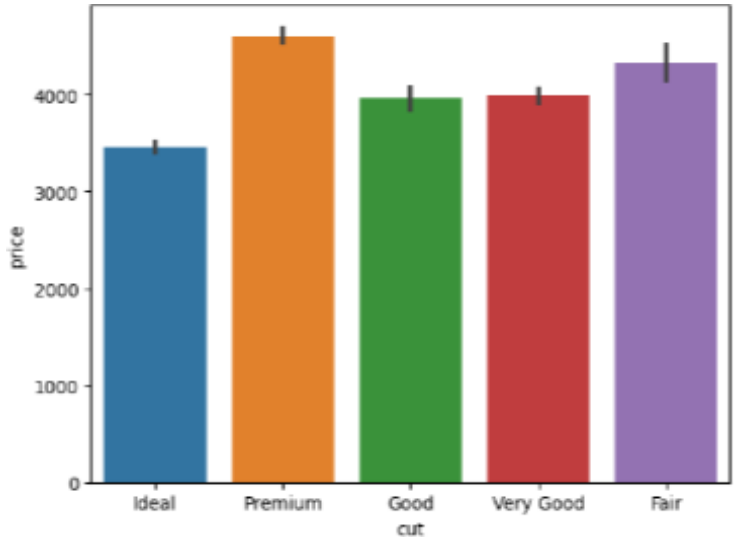


Fig 8.4 Bar chart represents the composition of cut in diamond.

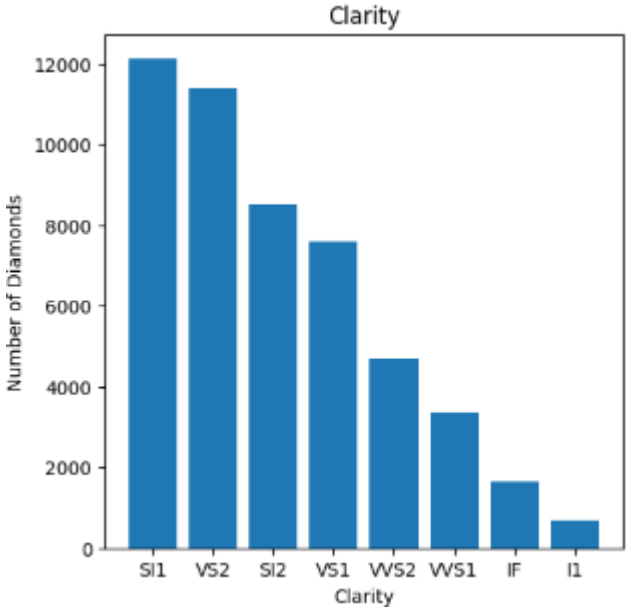


Fig 8.5 Bar chart represents the relationship between the number of diamonds and their clarity

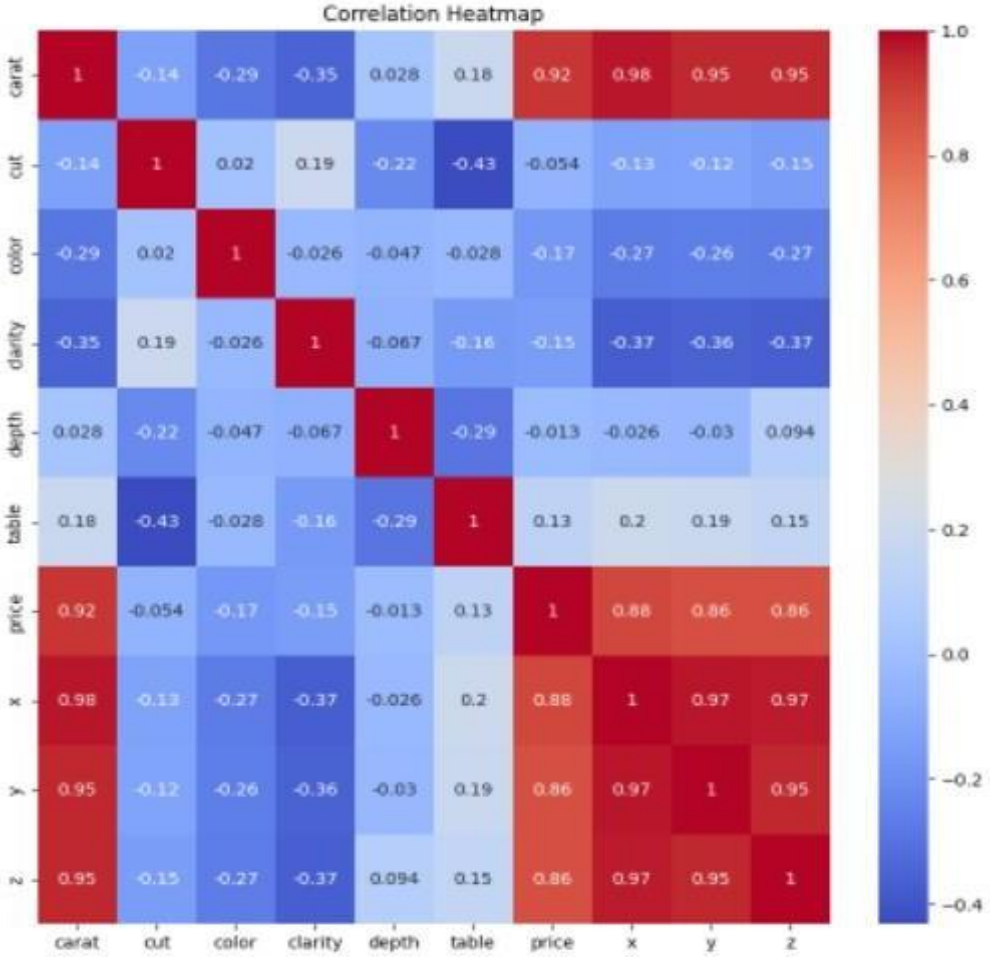


Fig 8.6 Represents the relationship between diamond attributes (eg. Carat, cut, color, clarity, depth, table, fluorescence) and price.

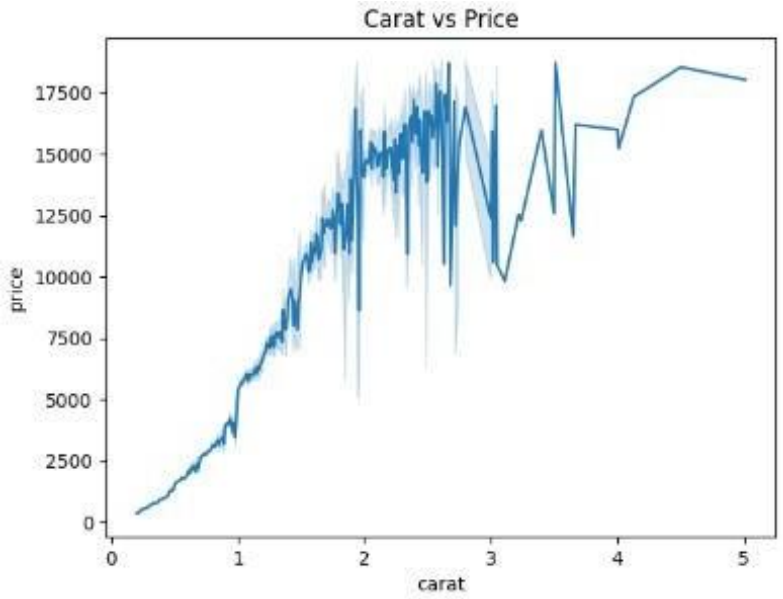
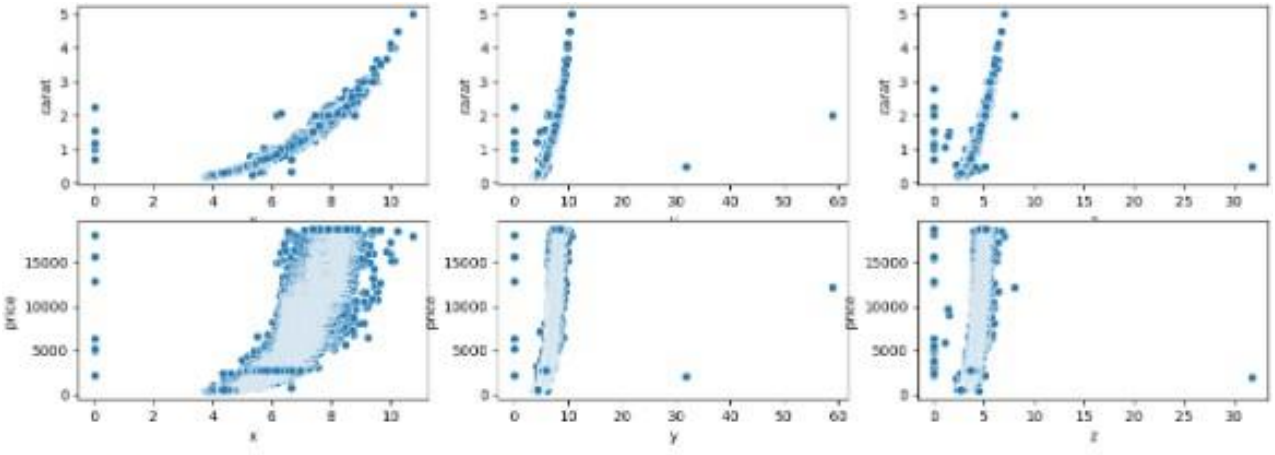
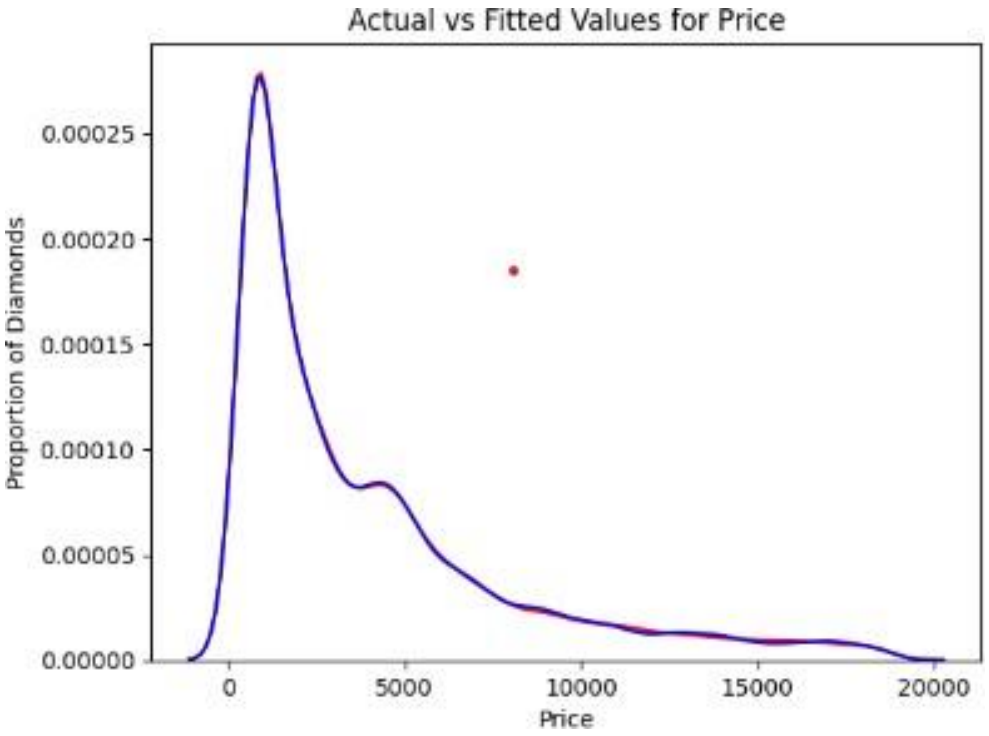


Fig 8.7 Line chart represents the relationship between the numberof diamonds and their clarity



Majority of the diamonds have x values between 4 and 8, y values between 4 and 10 and z values between 2 and 6. Diamonds with other dimensions are very rare.

Fig 8.8 Scatter plot



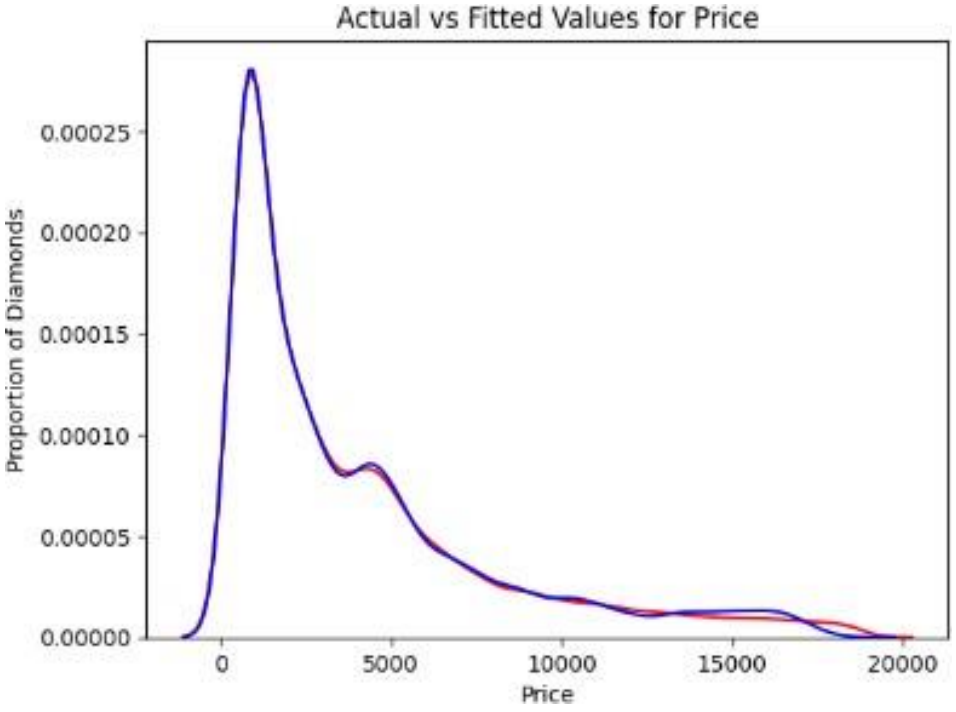


Fig 8.9,8.10 Relationship between proportion of diamonds versus price.

REFERENCES

1. Nikita Lemos, Ismail Pawaskar, Deepak Ramchandani, Taman Poojary, “Intelligent Sales Prediction using Machine Learning” © 2021 IJCRT | Volume 9, Issue 4 April 2021 | ISSN: 2320-2882

2. Sunitha Cheriyan, Shaniba Ibrahim, Saju Mohanan, “Sales Prediction using MachineLearning” Published 2018 Computer Science 2018 International Conference on Computing,
Electronics & Communications Engineering
(iCCECE)

3. Alessandro Massaro, Vincenzo Maritati, Angelo Galiano, “Data Mining Model Performance of Sales Predictive Algorithms Based on Rapidminer Workflows” Published 2018 Computer Science International Journal of Computer
Science and Information Technology

4. The Implementation of Data Mining Techniques for Sales Analysis usingDailySalesData inInternationalJournalofAdvancedTrendsin Computer Science nd Engineering 8(1.5):74-80November 2019 DOI:10.30534/ijatcse/2019/1681.52019

5. Deloitte Sales Forecasting Deloitte Analytics Approach The growing world of data https://www2.deloitte.com/content/dam/Deloitte/it/Documents/technology/Sales%20forecasti ng_Deloitte%20Analytics%20Approach_Delo itte%20Italy.pdf

6. Resource Quality Prediction Based on Machine Learning Algorithms 4th International Conference on Systems and Informatics (ICSAI 2017) DOI: 110.1109/Cybermatics_2018.00161 2017 4th International Conference on
Systems and Informatics (ICSAI)

7.Period Detection and Future Trend Prediction Using Machine Learning Techniques 21st Euromicro Conference on Digital Systems and Electronics Conference: 2018 IEEE International Conference on Internet of Things (iThings)and IEEE Green Computing and Communications (GreenCom)

8.A Machine Learning Approach for Area Prediction of Hardware Designs from Abstract Specifications Volume 71, November 2019, 102853 published at IEEE Machine Learning Design productivity Area estimation

9.Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis Volume 83, 2016, Pages 1064-1069 published at Elsevier Procedia Computer Science DOI: 10.1109/CIMCA.2018.8739696 published at IEEE.

10. Disease Prediction by Machine Learning over Big Data from Healthcare Communities International Journal of
Innovative Research in Computer and Communication Engineering Vol. 5, Issue 12 December2017 DOI:10.1109/ACCESS.2017.2694446