Principal Componenent Analysis and Multi-Class Classification on Quality of Diamonds Manufactured

Suryaprakash Rajkumar (**40236272**), A. Ben Hamza
Github Link: https://github.com/suryaprakashrajkumar/INSE6220

Abstract

*Diamonds are the most precious and strongest material known to mankind. Its widespread use and importance in many cultures is in-evident that it's manufactured in large quantities. This study is an exploratory data analysis and multiclass classification study on the quality price of diamonds and other attributes associated to it. We deal in principle with analysing the principal components (**PCA**) and classifying the multiclass class through the PYCARET library. Detailed results and analysis of this data set are discussed. Learning algorithms are compared with the PCA transormed dataset and the original dataset.*

1. Introduction

Data science and statistical analysis is the current predominant field of interest for research for researchers in the field of engineering and mathematics. The availability of big data and open source culture among users and corporations has enabled researchers and companies to innovate and redesign the way the data are processed and use for various use case purposes. This study mainly focuses on analysing the multi-class dataset on the quality and cost of diamonds produced. We intend to perform principal component analysis [1] and also use the PYCARET library (an open-source python based library for machine learning applications) to perform machine learning algorithms for classification and perform comparative studies on multiple classification algorithms and compare the performance of the learning algorithms before and after PCA.

2. Problem Statement

The problem statement is to perform a PCA analysis and classification study on a dataset ,i.e. Diamond quality dataset. Performing this study should provide sufficient evidence on the major contributing features and predominant characteristics of the data. Using the classification study, users can classify the diamonds based on the parameters of its quality. Performing ML on PCA transformed data and the original data

3. About the Dataset

The Dataset used for the anlaysis provides a detailed analysis on the Quality of Diamonds and its associated costs [2]. The Quality of the diamond is directly associated to the design parameters. Inevidently greater the quality of the diamond, greater is the price of the diamond correspondingly. The Dataset consists of 53940 observations and 10 corresponding attributes with the quality of cut as a class to comment on the quality of the dataset.

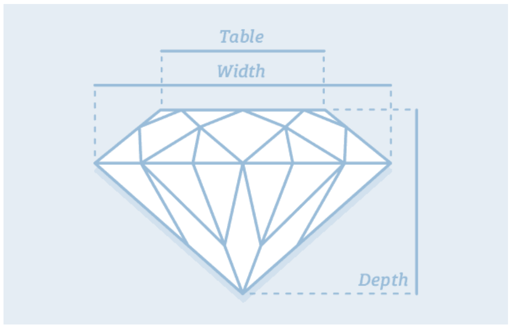


Figure 1. Diamond Design Construction

Every diamond manufactured has parameters such as table, width, depth, colour, size, and carat (weight) (Fig.1). Using these design attributes, a manufactured diamond can be assigned to a class of quality i.e. cut or based on clarity. Cut is designed with the following attributes Fair, Good, Very Good, Premium, Ideal and Clarity to I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best) in the order of worst to best in quality. along with these attributes, there is an associated cost for which the diamond was sold to a buyer. In this study we assign the class to the cut and the classification is done to predict the quality of the diamonds i.e. cut based on the design attributes.

The distribution of data sets classes is visualized in the figure. 2, with the majority of the classes being Ideal, premium, and very good almost contributing to 84% and fair being the lowest in number in the dataset.

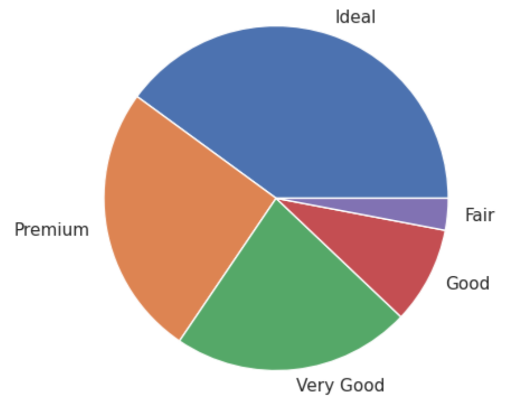


Figure 2. Pie chart for the dataset classes

The dataset has a total of 53940 observations, 13791 premium observations, 12082 very good observations, 4906 Good observations, 21551 ideal observations, with 1610 fair observations. Contributing to 25.5%, 22.3%, 0.09%, 39.9%, 0.02% respecively.

The below figure. 3 represents the distribution of the data with the representation of central values and variability in features. it is observed that all the features has outliers. Feature Carat has outliers only on the positive direction while others have on both positive and negative.

Figure. 4 represents the correlation dependencies of features with each other. We can see X, Y, Z and Carat have high correlation values with each other, which concludes these features have high interdependencies. Figure. 5 support the above observation as we can see highly correlated features being distributed on a monotonic increasing fashion while we can also see tge features depth and table being highly un-correlated with the other features.

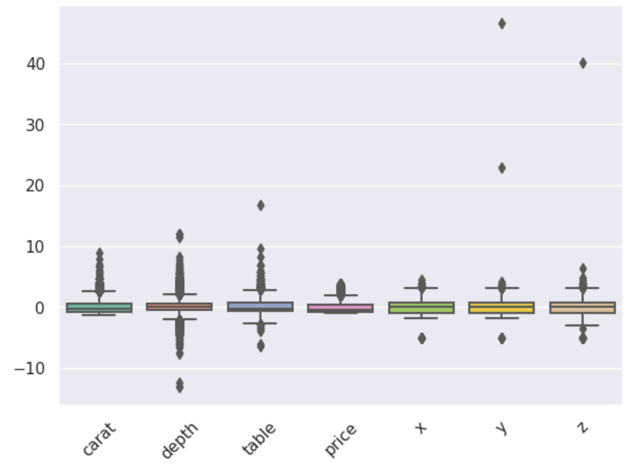
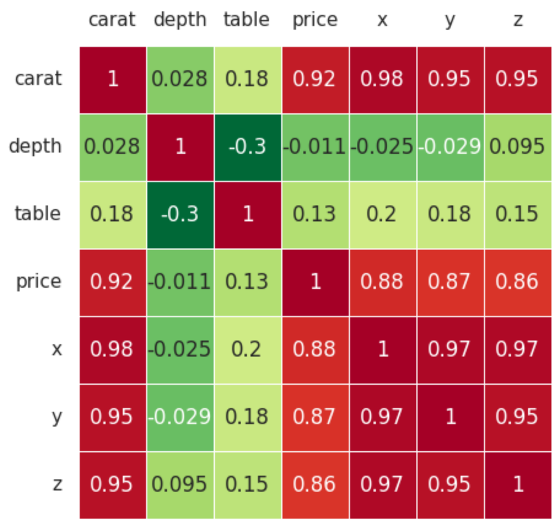


Figure 3. Box plot of the Dataset



	carat	depth	table	price	x	y	z
carat	1	0.028	0.18	0.92	0.98	0.95	0.95
depth	0.028	1	-0.3	-0.011	-0.025	-0.029	0.095
table	0.18	-0.3	1	0.13	0.2	0.18	0.15
price	0.92	-0.011	0.13	1	0.88	0.87	0.86
x	0.98	-0.025	0.2	0.88	1	0.97	0.97
y	0.95	-0.029	0.18	0.87	0.97	1	0.95
z	0.95	0.095	0.15	0.86	0.97	0.95	1

Figure 4. Correlation Matrix

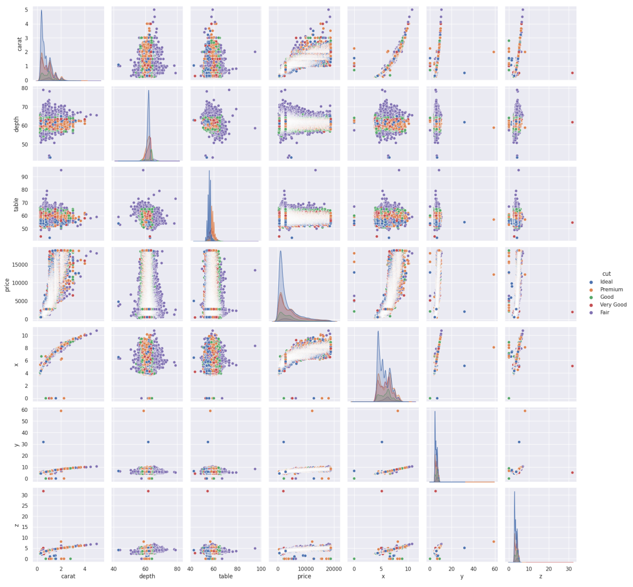


Figure 5. Pair-Plot

4. Preliminary - Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method widely used for data analysis and dimensionality reduction. It is a linear transformation technique that identifies the most significant patterns in a dataset by projecting it onto a lower-dimensional space while retaining as much of the original variation as possible. PCA achieves this by finding the principal components of the data, which are the orthogonal directions that capture the largest variance in the data. PCA has numerous applications in fields such as machine learning, image processing, finance, and biology, among others.

PCA involves computing the eigenvectors and eigenvalues of the covariance matrix of the dataset. The eigenvectors represent the principal components of the data, while the corresponding eigenvalues represent the variance explained by each principal component. The principal components are ordered in descending order of the corresponding eigenvalues, so that the first principal component captures the most variation in the data.

Mathematically, given a data matrix X with n rows and p columns, the covariance matrix C can be computed as follows:

$$C = \frac{1}{n-1}\left(X - \bar{X}\right)^T\left(X - \bar{X}\right) \tag{1}$$

where $\bar{X}$ is the mean of each column of X.

The eigenvectors and eigenvalues of C can be computed as follows:

$$C\mathbf{v} = \lambda\mathbf{v} \tag{2}$$

where $\mathbf{v}$ is the eigenvector and λ is the corresponding eigenvalue.

The principal components of the data can be obtained by projecting the data onto the eigenvectors:

$$\mathbf{A} = X\mathbf{V} \tag{3}$$

where $\mathbf{A}$ is the transformed data matrix, $\mathbf{V}$ is the matrix of eigenvectors, and the columns of $\mathbf{A}$ represent the principal components [3].

5. Preliminary - ML Algorithms

In this study we mainly tune and understand the following three ML algorithms, Logistic regression, Decision tree, K-nearest Neighbour algorithms and their preliminaries are explained in the following subsections.

5.1. Logistic Regression (LR)

Logistic Regression [4] is a variant of logistic regression that is used for classification problems with more than two classes. It models the probability of each class as a function of the input variables, and predicts the class with the highest probability. logistic regression can be formulated using the softmax function, which generalizes the sigmoid function used in binary logistic regression.

Mathematically, given a set of input variables X and a categorical outcome variable Y with K classes, logistic regression models the probability of each class k=1,2,...,K as:

$$P(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\cdots+\beta_{pk}X_p}}{\sum_{j=1}^{K} e^{\beta_{0j}+\beta_{1j}X_1+\cdots+\beta_{pj}X_p}} \tag{4}$$

where $\beta_{0k}, \beta_{1k}, \ldots, \beta_{pk}$ are the model coefficients or parameters for class k, and $X_1, X_2, \ldots, X_p$ are the input variables.

The softmax function ensures that the probabilities of all classes sum up to 1. The predicted class is the one with the highest probability:

$$\hat{Y} = \arg\max_k P(Y = k|X) \tag{5}$$

logistic regression can also be trained using maximum likelihood estimation, where the likelihood is maximized over the observed data. Regularization techniques such as L1 and L2 regularization can also be applied to logistic regression to prevent overfitting and improve generalization.

logistic regression is a widely used and powerful tool for multiclass classification tasks. Its simplicity and interpretability make it a popular choice in many fields, including image classification, natural language processing, and bioinformatics.

5.2. K-nearest Neighbour

K-Nearest Neighbors (KNN) [5]is a simple yet effective machine learning algorithm used for both regression and classification problems. The basic idea behind KNN is to find the K closest training examples (neighbors) to a given test example, and use their labels to predict the label of the test example.

Mathematically, given a set of training examples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and a test example x, KNN predicts the label $\hat{y}$ of x as follows:

The predicted label $\hat{y}$ is the majority label among the K closest training examples to x, where closeness is measured using a distance metric such as Euclidean distance or Manhattan distance:

$$\hat{y} = \arg\max_{y_i} \sum_{i=1}^{K} [y_i = y] \tag{6}$$

where y_i is the label of the i-th closest training example to x, and $[y_i = y]$ is the indicator function that is 1 if $y_i = y$ and 0 otherwise.

5.3. Decision Tree

Decision Trees [6] are a popular machine learning algorithm used for both classification and regression tasks. They build a hierarchical model of decisions and their possible consequences based on the input variables. The algorithm starts at the root node and splits the data into subsets based on the value of one input variable at a time. Each subset is then recursively split until a stopping criterion is met, such as a maximum depth of the tree or a minimum number of instances in each leaf node.

Mathematically, a decision tree can be represented as a binary tree, where each internal node represents a decision based on the value of an input variable, and each leaf node represents a class label or a numerical value. Let X be a vector of input variables and Y be the output variable, and let T(X) be the decision tree. The decision tree can be defined recursively as:

$$T(X) = \begin{cases} y_i & \text{if leaf node} \\ T_1(X) & \text{if } X_j < t_j \text{ and internal node} \\ T_2(X) & \text{if } X_j \geq t_j \text{ and internal node} \end{cases} \tag{7}$$

where y_i is the class label or numerical value at the leaf node, $T_1(X)$ and $T_2(X)$ are the left and right sub-trees, respectively, and j is the index of the input variable used for the decision at the internal node, and t_j is the threshold value for that variable.

6. Results

The following two sections discuss in detail the evaluated PCA algorithm and the learning algorithms implemented on the dataset.

6.1. PCA Results

Upon applying PCA on the given standardized dataset we get the following A matrix and λ

$$\lambda = \begin{bmatrix} 4.76400313 \\ 1.28589192 \\ 0.69082407 \\ 0.17375655 \\ 0.04030797 \\ 0.0329472 \\ 0.01239894 \end{bmatrix} \tag{8}$$

$$\text{A} = \begin{bmatrix} 0.45 & -0.03 & -0.01 & -0.06 & -0.133 & -0.768 & 0.42 \\ -0.001 & -0.73 & 0.67 & -0.04 & 0.08 & -0.014 & -0.05 \\ 0.09 & 0.67 & 0.72 & -0.05 & 0.01 & 0.02 & -0.01 \\ 0.42 & -0.03 & -0.10 & -0.84 & 0.05 & 0.27 & -0.08 \\ 0.45 & 0.02 & -0.03 & 0.249 & -0.08 & -0.19 & -0.82 \\ 0.44 & 0.01 & -0.05 & 0.32 & 0.77 & 0.21 & 0.20 \\ 0.44 & -0.08 & 0.03 & 0.31 & -0.60 & 0.49 & 0.27 \end{bmatrix} \tag{9}$$

As a result of PCA, the first 3 components contributed to about 98.77% of the variance in the entire dataset. With the first principal component majorly conteibuting with 68% variance. and its represented in the equation below.

$$\text{Z1} = 0.45\text{X}_1 - 0.001X_2 + 0.09X_3 + 0.42X_4 + 0.45X_5 + 0.44X_6 + 0.45X_6$$

The second component($Z2$ of the PCA analysis is represented in the equation below

$$\text{Z2} = 0.01\text{X}_1 + 0.67X_2 + 0.72X_3 - 0.1X_4 - 0.03X_5 - 0.05X_6 + 0.03X_6$$

and the third component $Z3$ is computed as follows,

$$\text{Z2} = -0.03\text{X}_1 - 0.73X_2 + 0.67X_3 - 0.03X_4 + 0.02X_5 + 0.01X_6 - 0.08X_6$$

The percentage of variance explained by each principle component is shown on both the scree plot and the Pareto plot. The following calculation through Eq. 10 can be used to calculate the proportion of variance that the j-th PC experienced.

$$j_j = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100, j = 1, 2, \ldots, p \tag{10}$$

using the explained variance the pareto chart figure. 7 and Scree plot figure. 6 has been plotted these values explain the variance contributed by each component for the entire dataset.

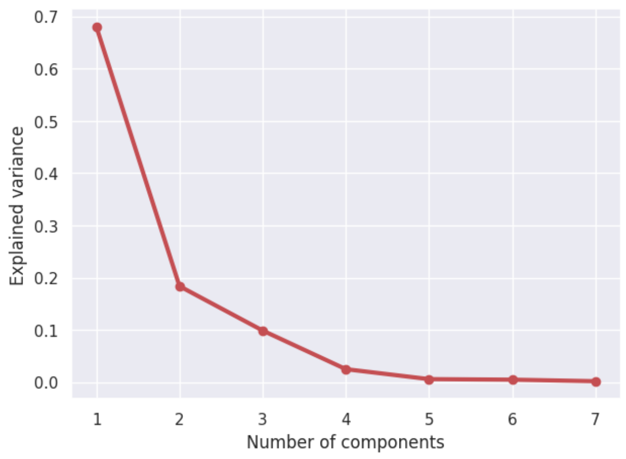


Figure 6. Scree Plot

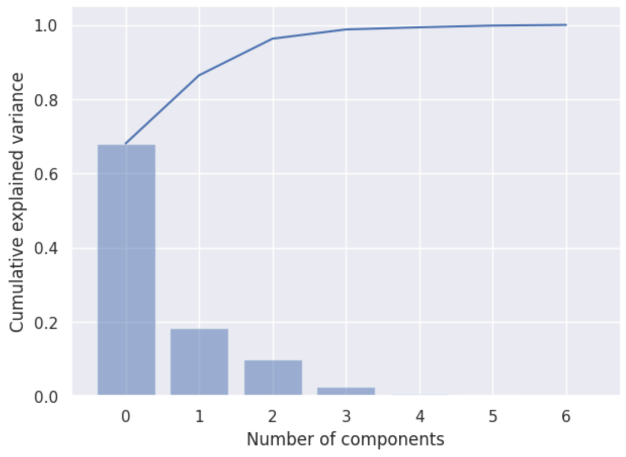


Figure 7. Pareto chart

Figure. 8 represents represents the PC coefficient plot. It visually represents the amount of contribution each feature has on the first two PCs. The figure supports the previous calculation of PCs with table, color, depth, price bwing the higherst contributor to the first principal componenent.

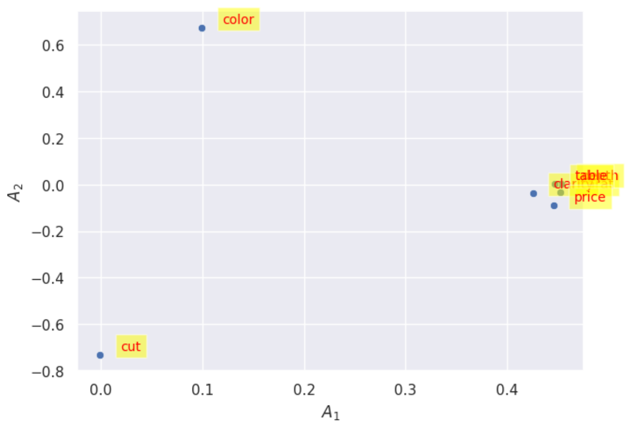


Figure 8. PCA Coeeficient plot

The figure below 9, the first two PCs are shown in a separate graphic form. The first two PCs are represented by the biplot's axes. The eigenvector matrix's rows are displayed as a vector. On the plot, each observation from the dataset is represented by a dot.

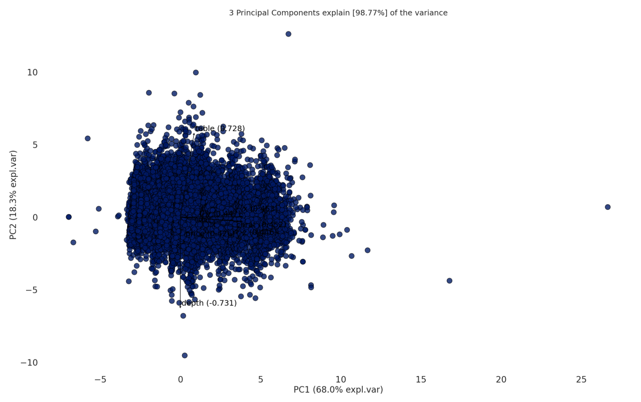


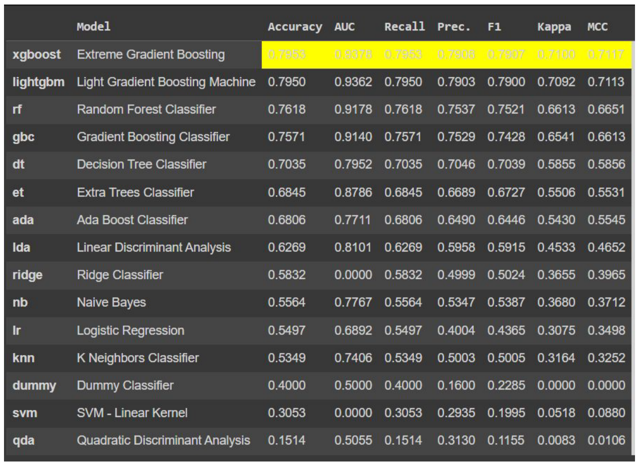
Figure 9. Biplot

6.2. Classification Results

In this section, we discuss the results of applying the PYCARET library to the given and PCA-transformed datasets. Initially we try to classify the data based on the labels of quality of the diamond produced. As seen in Figure. 10, PYCARET algorithm performed classification on the most available and efficient ML algorithms. As a result of this classification Xtreme Gradient Descending Method achieved a 0.7953% with the area under the curve (AUC) of 0.9378. AUC is a metric used to evaluate the performance of a binary classification model. It represents the area under the receiver operating characteristic (ROC) curve, which is a plot of true positive rate (TPR) against false positive rate (FPR) at different classification thresholds. The AUC value ranges from 0 to 1, where a perfect classifier has an AUC of 1. The formula for calculating AUC is:

$$AUC = \int_{-\infty}^{\infty} TPR(FPR^{-1}(u))du \qquad (11)$$

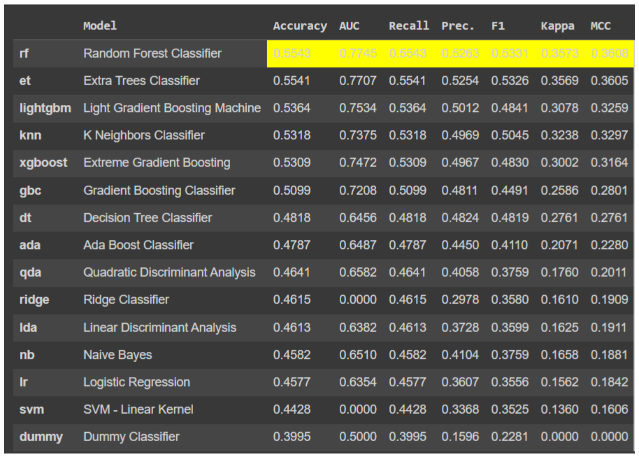
where TPR is the true positive rate and FPR is the false positive rate. $FPR^{-1}(u)$ is the inverse of the FPR, evaluated at u.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
xgboost	Extreme Gradient Boosting	0.7953	0.9378	0.7953	0.7908	0.7907	0.7100	0.7117
lightgbm	Light Gradient Boosting Machine	0.7950	0.9362	0.7950	0.7903	0.7900	0.7092	0.7113
rf	Random Forest Classifier	0.7618	0.9178	0.7618	0.7537	0.7521	0.6613	0.6651
gbc	Gradient Boosting Classifier	0.7571	0.9140	0.7571	0.7529	0.7428	0.6541	0.6613
dt	Decision Tree Classifier	0.7035	0.7952	0.7035	0.7046	0.7039	0.5855	0.5856
et	Extra Trees Classifier	0.6845	0.8786	0.6845	0.6689	0.6727	0.5506	0.5531
ada	Ada Boost Classifier	0.6806	0.7711	0.6806	0.6490	0.6446	0.5430	0.5545
lda	Linear Discriminant Analysis	0.6269	0.8101	0.6269	0.5958	0.5915	0.4533	0.4652
ridge	Ridge Classifier	0.5832	0.0000	0.5832	0.4999	0.5024	0.3655	0.3965
nb	Naive Bayes	0.5564	0.7767	0.5564	0.5347	0.5387	0.3680	0.3712
lr	Logistic Regression	0.5497	0.6892	0.5497	0.4004	0.4365	0.3075	0.3498
knn	K Neighbors Classifier	0.5349	0.7406	0.5349	0.5003	0.5005	0.3164	0.3252
dummy	Dummy Classifier	0.4000	0.5000	0.4000	0.1600	0.2285	0.0000	0.0000
svm	SVM - Linear Kernel	0.3053	0.0000	0.3053	0.2935	0.1995	0.0518	0.0880
qda	Quadratic Discriminant Analysis	0.1514	0.5055	0.1514	0.3130	0.1155	0.0083	0.0106

Figure 10. Classification on original dataset

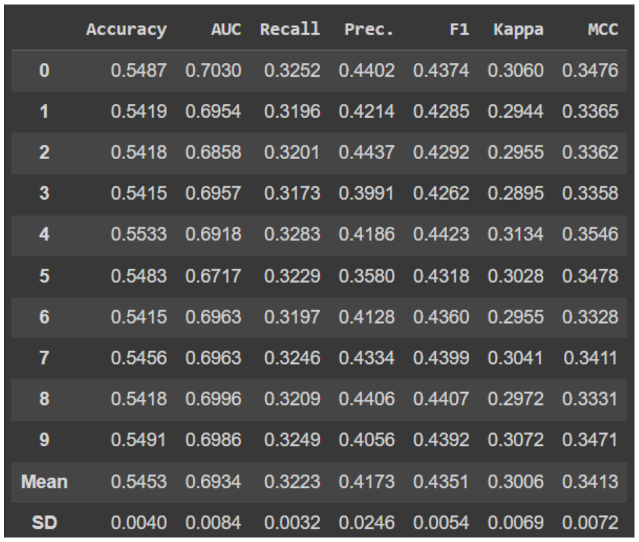
Then the same algorithm was applied to the PCA-transformed dataset and the results are shown in figure. 11 and Random forest tree performs the best for classification with an accuracy of 55.43% and AUC of 0.7745. The difference in the accuracy is due to the reduction in dimension and availability of less features compared to the classification on the original dataset.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
rf	Random Forest Classifier	0.5543	0.7745	0.5543	0.5263	0.5331	0.3573	0.3608
et	Extra Trees Classifier	0.5541	0.7707	0.5541	0.5254	0.5326	0.3569	0.3605
lightgbm	Light Gradient Boosting Machine	0.5364	0.7534	0.5364	0.5012	0.4841	0.3078	0.3259
knn	K Neighbors Classifier	0.5318	0.7375	0.5318	0.4969	0.5045	0.3238	0.3297
xgboost	Extreme Gradient Boosting	0.5309	0.7472	0.5309	0.4967	0.4830	0.3002	0.3164
gbc	Gradient Boosting Classifier	0.5099	0.7208	0.5099	0.4811	0.4491	0.2586	0.2801
dt	Decision Tree Classifier	0.4818	0.6456	0.4818	0.4824	0.4819	0.2761	0.2761
ada	Ada Boost Classifier	0.4787	0.6487	0.4787	0.4450	0.4110	0.2071	0.2280
qda	Quadratic Discriminant Analysis	0.4641	0.6582	0.4641	0.4058	0.3759	0.1760	0.2011
ridge	Ridge Classifier	0.4615	0.0000	0.4615	0.2978	0.3580	0.1610	0.1909
lda	Linear Discriminant Analysis	0.4613	0.6382	0.4613	0.3728	0.3599	0.1625	0.1911
nb	Naive Bayes	0.4582	0.6510	0.4582	0.4104	0.3759	0.1658	0.1881
lr	Logistic Regression	0.4577	0.6354	0.4577	0.3607	0.3556	0.1562	0.1842
svm	SVM - Linear Kernel	0.4428	0.0000	0.4428	0.3368	0.3525	0.1360	0.1606
dummy	Dummy Classifier	0.3995	0.5000	0.3995	0.1596	0.2281	0.0000	0.0000

Figure 11. PCA transformed dataset classification results

The adjustment of hyperparameters is crucial for enhancing a model's performance. The following steps are involved in hyperparameter tuning using PyCaret: build the model, tune it, and assess performance. A classification model for each method is initially created. The model is then tuned using the tune_model() method with optimum hyperparameters. Using stratified K-fold cross-validation, this method automatically optimises the model with useful hyperparameters on a predefined search space. By default, PyCaret validates the models using a 10 fold stratified K-fold method as represented in the figure 12.



	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.5487	0.7030	0.3252	0.4402	0.4374	0.3060	0.3476
1	0.5419	0.6954	0.3196	0.4214	0.4285	0.2944	0.3365
2	0.5418	0.6858	0.3201	0.4437	0.4292	0.2955	0.3362
3	0.5415	0.6957	0.3173	0.3991	0.4262	0.2895	0.3358
4	0.5533	0.6918	0.3283	0.4186	0.4423	0.3134	0.3546
5	0.5483	0.6717	0.3229	0.3580	0.4318	0.3028	0.3478
6	0.5415	0.6963	0.3197	0.4128	0.4360	0.2955	0.3328
7	0.5456	0.6963	0.3246	0.4334	0.4399	0.3041	0.3411
8	0.5418	0.6996	0.3209	0.4406	0.4407	0.2972	0.3331
9	0.5491	0.6986	0.3249	0.4056	0.4392	0.3072	0.3471
Mean	0.5453	0.6934	0.3223	0.4173	0.4351	0.3006	0.3413
SD	0.0040	0.0084	0.0032	0.0246	0.0054	0.0069	0.0072

Figure 12. Example of 10 K-fold stratification applied to LR tuning

We perform further tuning on the decision tree, KNN and Logistic regression analysis for comparison purposes and in-depth study. we also perform the same analysis on the best-performing classification algorithm for the PCA transformed dataset i.e. random forest tree. The following four figures (figure. 13 ,14, 15, 16 represent the learning accuracy curves, AUC curve and Confusion matrix on the test dataset for the decision tree, K-Nearest Neighbours, Logistic regression and PCa best-performing algorithm i.e. random forest tree.

A multiclass confusion matrix is a table that summarizes the performance of a multiclass classification model. It compares the predicted values of the model against the actual values for multiple classes to measure the accuracy of the model. The matrix consists of various metrics like True Positive, False Positive, True Negative, and False Negative for each class, which helps to evaluate the effectiveness of a model in terms of identifying multiple classes. With the results analyzed through the confusion matrix we can also substantiate that the Decision tree on the original dataset performs the best in the prediction of th e quality of diamonds.

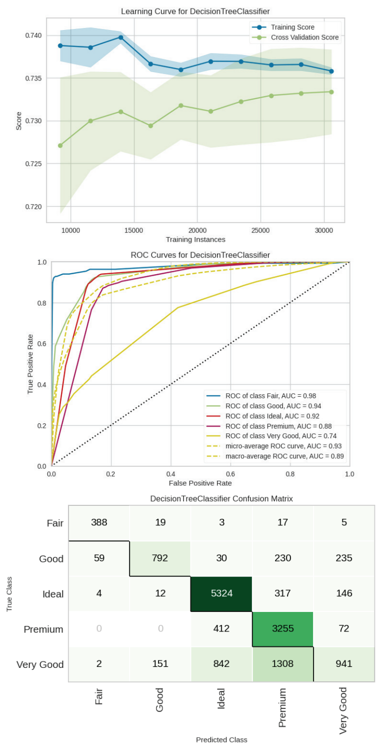


Figure 13. Decision Tree tuned results

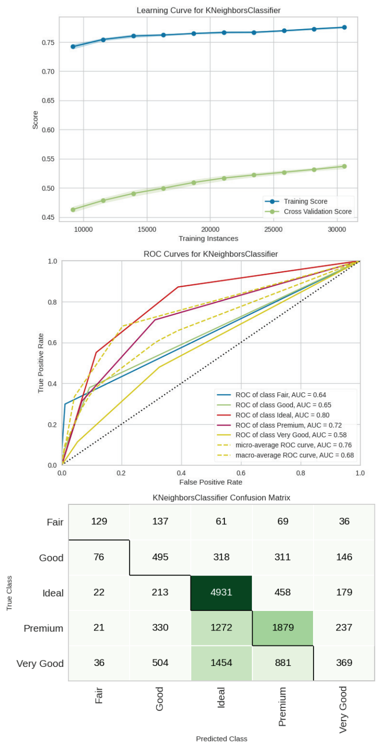


Figure 14. KNN tuned results

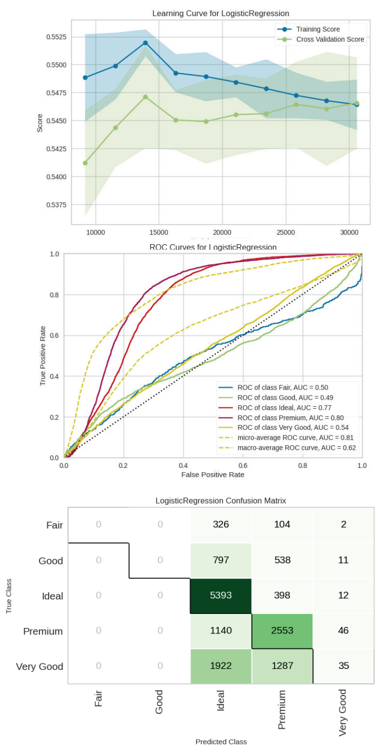


Figure 15. Logistic Regression tuned results

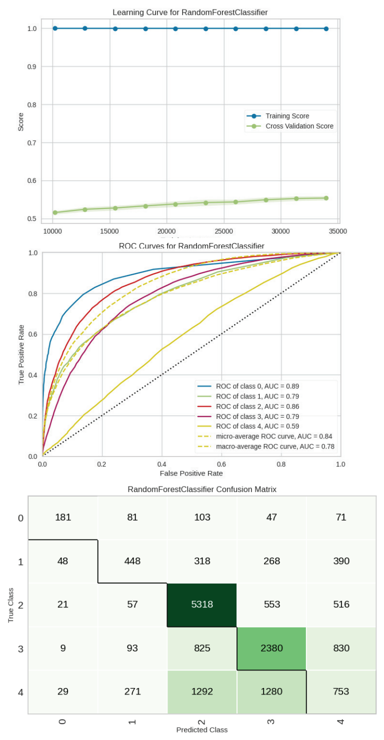


Figure 16. PCA tuned results

Both the ROC curve and the confusion matrix can be seen as two visual representations of the same measurement because of their close relationship. The above results on the accuracy and ROC/AUC curves show the results that are being produced on the classification. With the aforementioned algorithms, we can indeed predict the class and use it for realtime prediction purposes and can be of great use in the industry

7. Conclusion

In conclusion,Principal Component Analysis (PCA) is a potent method for maintaining the most crucial data while lowering the dimensionality of high-dimensional datasets. It can be used to spot trends and connections between variables and aid in the more insightful visualisation of data.

Principal component analysis (PCA) can be used to extract new features from a dataset that can then be used in further analyses like clustering or classification. PCA is a commonly used method that can be a vital tool for data analysis and interpretation in a variety of sectors, including biology, finance, and image processing.

The multiclass classification model for the quality of diamonds can effectively predict the quality of diamonds based on various The model's performance can be evaluated using a multiclass confusion matrix, which provides insights into the model's accuracy, precision, recall, and F1-score for each class. By analyzing the confusion matrix, we can identify areas where the model may need improvement and make necessary adjustments to improve the model's performance. Overall, the multiclass classification model for diamond quality is a useful tool for diamond industry professionals to make informed decisions when evaluating diamonds based on quality.
Note: All the results, and graphs on the learning methods are uploaded on the GitHub link.

References

[1] I. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, p. 20150202, 04 2016. 1

[2] S. Agrawal, "Diamonds quality and price dataset," May 2017. 1

[3] A. B. Hamza, "Advanced statistical approaches to quality," Dec 2022. 3

[4] P. Karsmakers, K. Pelckmans, and J. A. Suykens, "Multi-class kernel logistic regression: a fixed-size implementation," in *2007 International Joint Conference on Neural Networks*, pp. 1756–1761, IEEE, 2007. 3

[5] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 2126–2136, IEEE, 2006. 3

[6] G. Madzarov and D. Gjorgjevikj, “Multi-class classification using support vector machines in decision tree architecture,” in *IEEE EUROCON 2009*, pp. 288–295, IEEE, 2009. 3