Diamond cut

A machine learning classification project

Juan Bautista García Traver
University of Santiago de Compostela
Burriana, Castellón, Spain
juanbautista.garcia.traver@rai.usc.es

Alfredo del Río Moldes
University of Santiago de Compostela
Vigo, Pontevedra, Spain
alfredo.del@rai.usc.es

Duarte Salvado Rubio
University of Santiago de Compostela
Santiago de Compostela, A Coruña, Spain
duarte.salvado@rai.usc.es

Alejandro González Santos
University of Santiago de Compostela
Ribeira, A Coruña, Spain
alejandro.gonzalez.santos@rai.usc.es

ABSTRACT

This project proposes an analysis of different classifier models to determine the cut quality of a diamond. A dataset is used and preprocessed to produce training data, performing a descriptive analysis of it through different graphs.

Subsequently, several Machine Learning supervised models, together with an ensemble composed of multiple models, were tested through a cross-validation procedure, including SVM, decision trees, ANN and KNN, comparing their performance values.

Different metrics were used to make this comparison, finally determining the best ML technique and hyperparameters, for which its metrics and a study of its performance are shown. The best models are assembled, and the performance of this ensemble is studied, obtaining also values for it´s metrics.

CCS CONCEPTS

• Computing methodologies ~ Machine learning • Applied computing ~ Physical sciences and engineering ~ Earth and atmospheric sciences

KEYWORDS

Machine learning, Diamonds, Classification, Neural networks, SVM, kNN, Decision tree, Random forest, Ensembles

1. INTRODUCTION

A diamond is a mineral with physical and optical characteristics that make it one of the most valuable minerals in the world. It has extreme hardness, as it has a very rigid crystalline structure that prevents it from being contaminated by impurities. Its transparency is very high, with a high brilliance, which is why it has been used for many years as a gemstone in jewelry. It also has an increasing number of industrial applications, such as the manufacture of machines and tools for drilling, polishing or sawing [1].

In terms of jewelry, the cut of a diamond is important because it is the characteristic that determines its brilliance. Even if a diamond has a high degree of clarity and color, it can look dull if it is not cut correctly. The cut of a diamond is determined by precision, symmetry between facets and polish and can be classified into 5 classes: excellent, very good, good, fair and poor. [2]

The field of Machine Learning gives computers the ability to learn on their own, without being specifically programmed to do so. An ML algorithm starts with data that is prepared for training, allowing programmers to choose a model and let the computer train to find patterns or make predictions [3]. In a classification problem an object is classified depending on the similarity of its characteristics to one of the n classes into which it is to be classified.

A problem of classification of diamonds according to their cut quality arises. Having as starting data certain values of the physical characteristics of each diamond, it should be possible to determine what is its expected cut, being able to be classified and stored correctly at the production site.

The main objective of this work is to define and train different classification machine learning models to predict the cut quality of a diamond.

Secondary objectives are to compare the performance of different models with different hyperparameters for this specific problem.

The Report is divided into 5 main sections in addition to the introduction itself.

In the first section, the state of the art of the problem is analyzed through a brief bibliographic analysis of the main articles found.

The material and methods section provides a description of the dataset and the classification techniques used, as well as some specifications of the implementation and development decisions taken.

The data preparation section details the preprocessing performed on the database in order to make possible the use of certain techniques and to improve their performance.

The results section shows the descriptive analysis of the dataset variables, a correlation analysis of the variables and the performance of the different models for the specified task, as well as other results of interest.

Finally, the last section discusses the performance shown by the different models.

2. RELATED WORK

In order to perform this work, it has been studied in previous publications, related to the application of machine learning to diamonds.

In the article "Machine Learning Algorithms for Diamond Price Prediction" [4], several machine learning models are used to predict the price of diamonds from the same dataset used in this project. The authors conclude that random forest regression provides the best results, with a MAE of 112.93 and a RMSE of 241.97.

In "Diamond quality assessment system using machine learning approach" [5], the quality of diamonds is assessed by analyzing images and extracting features such as color, texture and clarity, which are then passed to a classifier. An accuracy of 92.65% was obtained.

In the article "Comparative Analysis of Supervised Models for Diamond Price Prediction" [6], several models for predicting the price of diamonds, also with the same dataset, are compared and, like in [4], the random forest model provides the best result, with 97.93% accuracy and 581.91 RMSE.

The article "Diamond Price Prediction using Machine Learning" [7] is another comparison of regression models for predicting the price, but this time, the best result, a test accuracy of 98,72% and RMSE of 525.81, is obtained with the CatBoost method.

3. MATERIAL & METHODS

The dataset to be used contains almost 54,000 samples of diamonds describing several features, based on scales created by the most recognized companies of the diamond industry [8].

These features are:

- **carat**: Denotes the weight of the diamond. One carat equals 0.2 grams. In this dataset, it varies between 0.2 to 5.01.
- **cut**: Stylish guide for the polish of a diamond. The values can be Fair, Good, Very Good, Premium, Ideal. This will be the target variable.
- **color**: Grading scale for diamonds, established as the standard of the industry, used by recognized laboratories (GIA & IGI for example). The values vary from J as the worst value to D as the best value.
- **clarity**: Measure of the internal characteristics (inclusions) and the surface defects of a diamond. It is expressed in the GIA scale. The values vary from I1 as the worst value to IF as the best value.
- **x**: Length of a diamond. Measured in millimeters. In this dataset, it varies from 0 to 10.74.
- **y**: Width of a diamond. Measured in millimeters. In this dataset, it varies from 0 to 58.9.
- **z**: Depth of a diamond. Measured in millimeters. In this dataset, it varies from 0 to 31.8.
- **total percentage of depth**: The height of a diamond, from the culet to the table, divided by its girdle diameter. The values vary from 43 to 79.
- **table**: The width of the top of a diamond relative to the widest point. Measured in percentage. The values vary from 43 to 95.
- **price**: The sale value. Measured in dollars.

In the figure 1 below, we can see an explanatory illustration of the last diamond features.

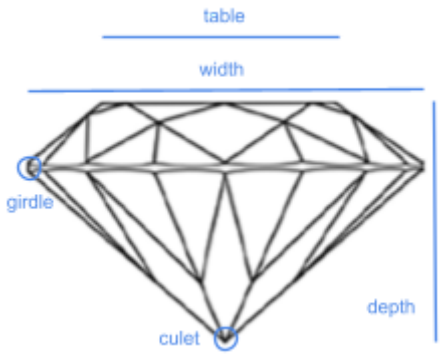

Figure 1: Diamond features [9].

For evaluating the model, we compute several metrics by applying this model to the test dataset. The primary metric to be used is the F-score, since it aggregates recall and precision and it is more appropriate than accuracy for unbalanced datasets. Although this metric is used to select the best models, other metrics from the confusion matrix are obtained, such as positive and negative predictive values, sensitivity and specificity.

As a multi-class classification problem is being solved,treating it as N binary problems, in a one-vs-all manner, generating a confusion matrix for each class and averaging the obtained metrics. As the dataset is unbalanced, this is done with a weighted average.

Different machine learning models are used, which are Support Vector Machines, Artificial Neural Networks, Decision Trees and K-nearest Neighbors. A brief description of each of them is given below.

The SVM uses a maximum margin separator, that is a decision boundary with the largest possible distance to the example points, that is useful to generalize correctly. Then, it creates a linear separating hyperplane, and with the ability to embed the data into a higher-dimensional space, they are able to use the “kernel trick" (data not linearly separable in the original space can be easily separated in the higher dimensional space). The SVM are non

parametric and combine the advantages of non parametric and parametric models, being able to represent complex functions but also being resistant to overfitting [10].

When the dataset is not very large, the SVM is a good method to try, since it performs similarly to logistic regression on separable data and can be better for high-dimensional data [10].

A decision tree is a representation of a function that maps a vector of attribute values to a single output value, performing a sequence of tests to reach a decision. Each internal node in the tree corresponds to a test of the value of one of the input attributes, the branches from the node are labeled with the possible values of the attribute, and the leaf nodes specify what value is to be returned by the function [10]. In general terms, the decision trees can work with discrete or continuous values for the inputs and outputs [10].

A neural network is a computing system made up of a large number of simple elements, elements of highly interconnected processes, which process information through their dynamic state in response to external inputs [11].

In learning, the weighted links between the neurons of the network are adjusted to obtain certain results. Therefore, the network does not need an algorithm to solve a problem, since it can distribute the weights of the links by learning to obtain the desired result. Therefore, the main function of the designer is to obtain an appropriate architecture for the problem to be addressed [11].

K-Nearest neighbors lookup consists in, given a query xq, instead of finding an example that is equal to xq, find the k examples that are nearest to xq. In this case a classification problem is treated. An example using K-Nearest Neighbors would be to find the set of neighbors $NN(k, x_q)$ and take the most common output value. For example with $k = 3$, if the values are $<Yes, No, Yes>$, the resulting classification will be *Yes*. An odd value of k is often used to avoid ties in binary classification problems.[10].

3.1. Explanation of the code structure

The code is organized into three files:

- ml_group_practice.jl:

This is the main script, it loads and preprocesses the data, generates plots for the descriptive analysis of the data, performs the necessary transformations for training the models (one-hot encoding, train-test split and normalization), performs cross-validation for selecting the best hyperparameters and finally trains a model with each approach using the selected hyperparameters.

- practice_functions.jl:

Contains the auxiliary functions developed for performing each of the required tasks in this project.

- assignments_functions.jl:

Contains the functions developed for previous assignments which were reused in this project.

4. DATA PREPARATION

The numerical variables of the dataset will be normalized using a standard zero-mean normalization. To carry out this processing, the normalization parameters appropriate to the training dataset of the model are obtained. These parameters are stored for later use in the normalization of the test dataset, which must be done in a similar way.

The target variable, in this case the diamond cut type, is encoded using the One-Hot method. This is done in order to facilitate its use in certain models, such as ANNs with softmax activation function in the output layer.

There are variables in the dataset of ordinal qualitative type, that is, they are ordered with greater or lesser relevance according to some criterion. For these variables an ordinal numerical encoding is performed, assigning them a real value between 0 and 1 according to the quality represented.

For example, A/B/C/D could become 0/0.33/0.66/1. However, this case is only interesting when in the real world there is an order A < B < C < D. To illustrate this, one can take the case of the color of the diamond, which can be lighter or darker (from D to J), being a diamond of higher (D) or lower (J) quality.

Training a model will be much faster if the inputs provided are on the same scale, i.e. if the model is spared from having to learn the relationship between the scales on which each attribute moves. This type of preprocessing allows a simpler model to solve more complex problems than without it, since it does not need to use part of it to learn the relationship between the scales of the input attributes.

It is necessary to save the parameters used in the normalization for each attribute (mean and standard deviation, training set).

MinMax normalization is appropriate when you are certain that the data is bounded (both top and bottom), since this does not occur for the numerical variables in this problem, it was decided to apply the standard normalization, which is more robust to outliers.

A stratified cross-validation method is used to validate and select the parameters that obtain the best validation metrics for the different models. To perform this, the dataset is previously separated into 90% of samples for cross-validation and 10% for testing, using a hold-out method.

Due to the size of the dataset, it was decided to separate 5 folds from the cross-validation set to perform the cross-validation of the different models. Each fold contains 18% of the total number of samples, which in turn represents 20% of the cross-validation set.

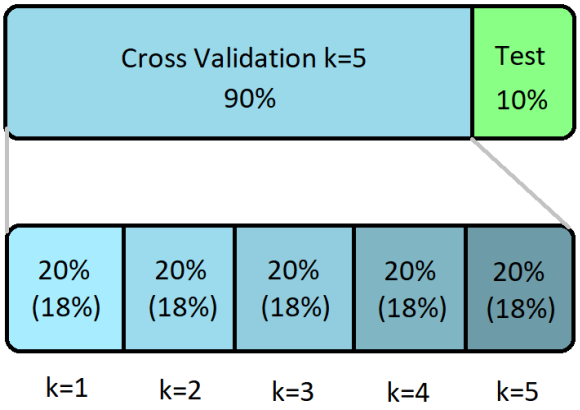

Figure 2: Cross Validation Partition

Because the number of variables is small and in the descriptive analysis, especially in the graphs, there is no clear separation of the classes of the target variable, it was decided to include all candidate variables in the machine learning models.

Nor was any type of dimensionality reduction performed for the same reason, in addition to preserving the original variables and obtaining from models such as the Decision Tree an explanatory and relevant information on the variables used by the model.

5. RESULTS

5.1 Descriptive Analysis

First, a descriptive analysis of all the variables and all the patterns was carried out. The analyzed data set is filtered and without outliers but not normalized. The graphs corresponding to this section can be consulted in section 2.4, while the main statistics obtained are shown and analyzed in this section.

Feature	Mean	S.D.	CV (%)
Carat	0.7088	0.3709	52.33
Length	5.5470	0.9795	17.66
Width	5.5520	0.9737	17.54
Depth	3.4287	0.6057	17.67
Volume	115.5268	59.9088	51.86
Total depth (%)	61.7861	1.1062	1.79
Table	57.2610	2.0214	3.53
Price	3110.96	2749.82	88.39

Table 1: Numeric features statistics.

The means, quasi-standard deviations and coefficients of variation of the numerical variables can be found in Table 1.

The length and width variables have similar means and standard deviations, while the depth variable has a mean of two units less. The coefficient of variation of the three variables is similar, around 17.6%

The volume, carat and especially price variables present the highest CV values, with 51.9%, 52.3% and 88.4% respectively.

The variables total-depth and table have a fairly low variation percentage of 1.79% and 3.53% each.

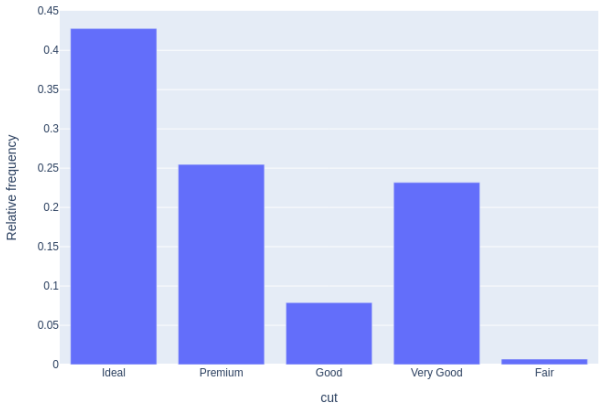

Figure 3: Relative frequency cut

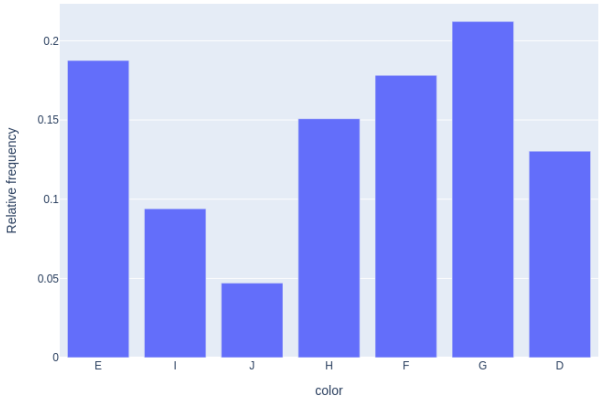

Figure 4: Relative frequency color

As can be seen in Figure 3, the target variable is clearly unbalanced, with many more diamonds having an 'Ideal' quality cut and very few having a 'Fair' quality cut. Specifically, 42.8% of the diamonds in the dataset have an 'Ideal' cut quality and only 0.7% have a 'Fair' cut quality.

The ordinal qualitative variable color is somewhat more balanced than the target variable. The least frequent color class in the sample (4.7%) is J, the worst quality one, while the most frequently repeated color (21.2%) is G, which corresponds to an intermediate color quality.

Regarding the clarity variable, the category that stands out for the low proportion of cases is I1 with 1.0% of diamonds. The categories with a higher presence in the dataset are SI1 and VS2 with 24.3% and 23.0% of diamonds respectively.

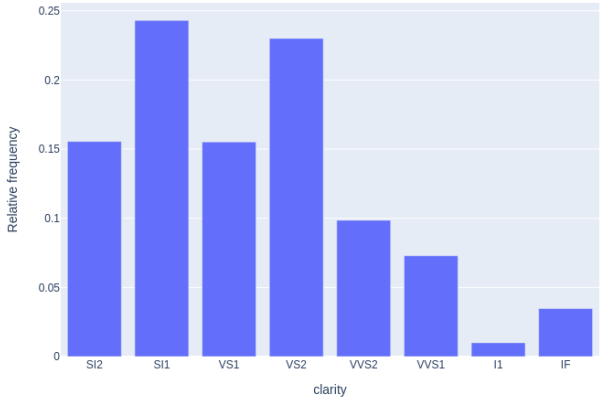

Figure 5: Relative frequency clarity

Some graphs corresponding to the descriptive analysis of the data set.

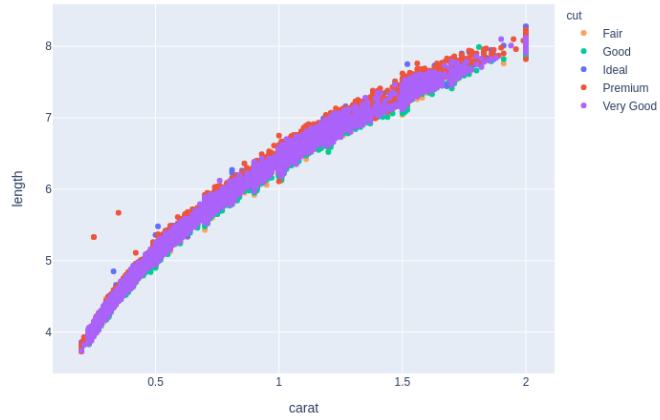

Figure 6: carat VS length

Figure 6 shows the relationship between the variables carat and length. Despite having obtained a high value of linear correlation between this pair of variables, the graph seems to reveal a logarithmic relationship.

Something similar occurs in Figure 7, where the relationship between depth and price is analyzed. The relationship shown by the graph seems to be of the logarithmic type.

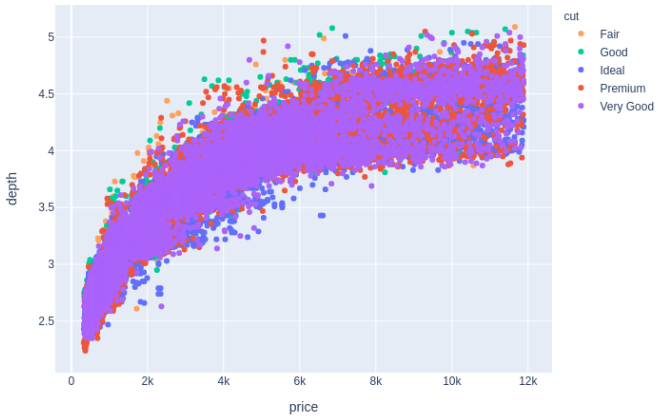

Figure 7: depth VS price

As is beginning to be seen in these plots and will be seen in all point cloud plots, the separation of the target classes is not at all clear. In no case is linear separation of any pair of classes possible, at least with two dimensions.

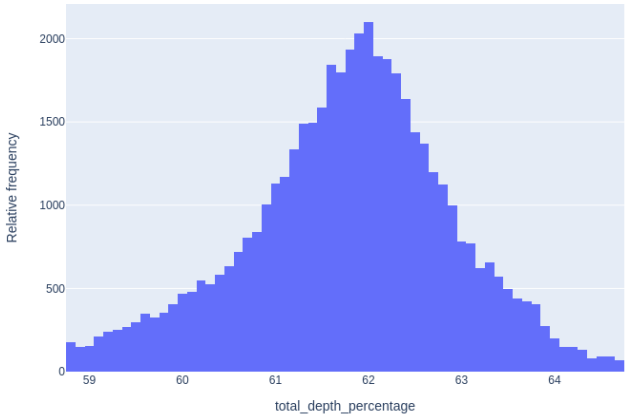

Figure 8: Histogram of total depth (%)

As can be seen in figure 8, the distribution of the values of the variable 'total percentage of depth' seems to follow a normal distribution with a mean of 61.8 and a standard deviation of 1.1. The median of these values is 61.9.

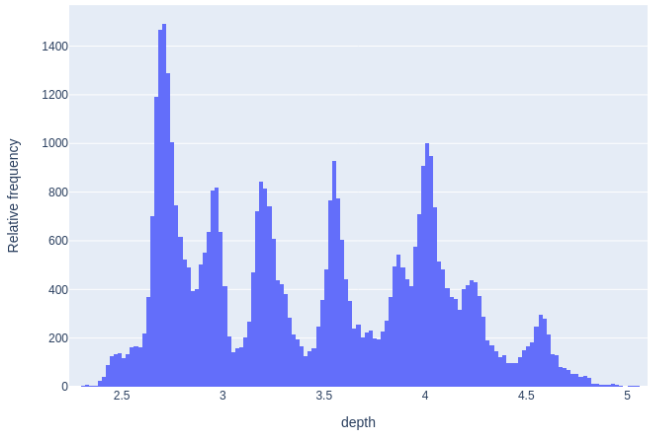

Figure 9: Histogram of depth

Figure 9 shows the distribution of the values of the depth variable. The dispersion of this variable is high and we could say that we are dealing with a variable with a multimodal distribution. 50% of the values are below 3.36.

The length, width, volume, and carat variables have a histogram similar to that of depth in terms of multimodal characteristic and high dispersion.

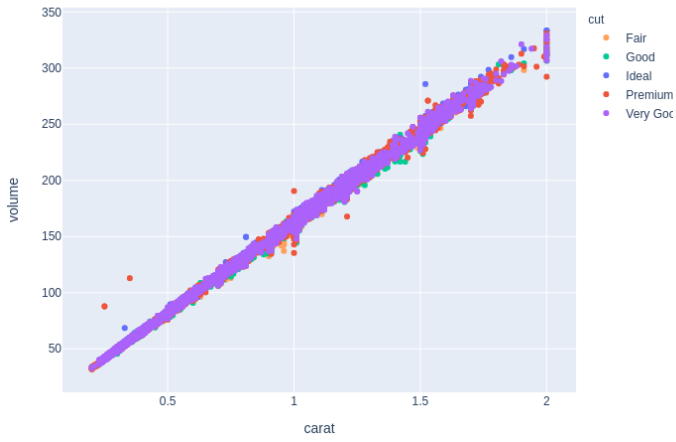

Figure 10: volume VS carat

Figure 10 shows the relationship between the variables volume and carat. The graph shows a very strong positive linear relationship.

The volume variable does appear to have a positive linear relationship with the carat variable, unlike length, width and depth, which show a logarithmic relationship.

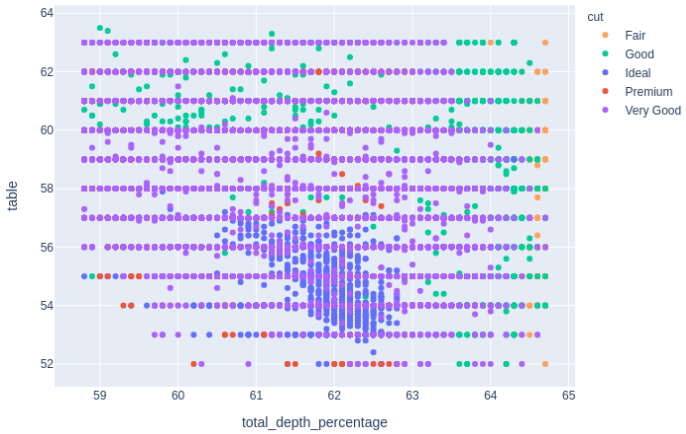

Figure 11: table VS total_depth

In figure 11, in which the table and total depth variables are compared, a certain separation between the categories of the objective variable begins to be seen.

Diamonds with a fair cut appear to have a high total depth percentage, while diamonds with a good cut have high table values and/or high total depth percentage values.

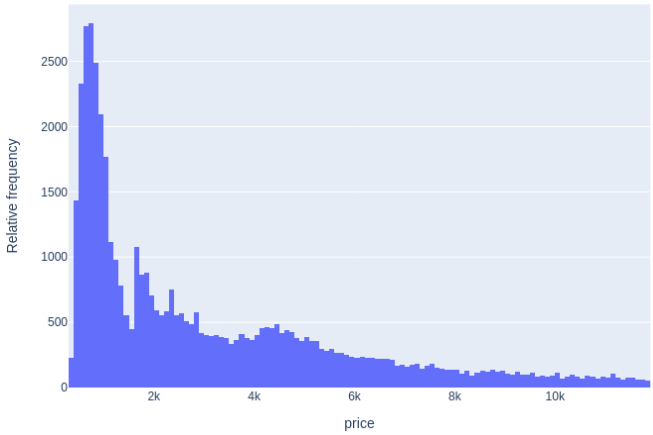

Figure 12: Histogram of price

In figure 12 we can see the distribution of the prices of the diamonds in the dataset. Most of the sampled diamonds have a cheaper value and as the price increases the number of patterns decreases.

50% of the diamonds in the dataset have a value below $2063.

The 3D point cloud graph of the length, width and depth variables shown in Figures 13 and 14 from different perspectives, can be

seen as possible separations of the different categories of the cut variable appear from the interactions of these three variables.

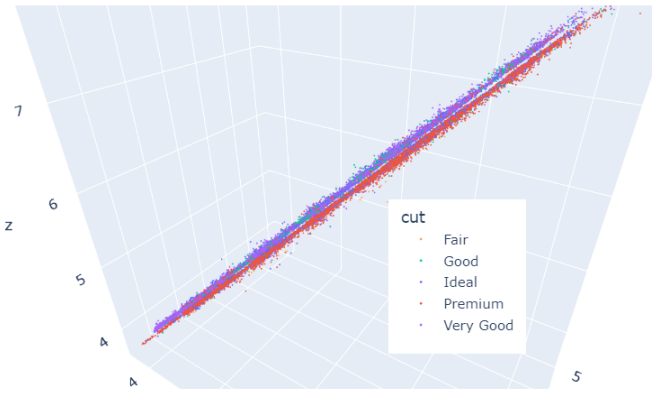

Figure 13: length vs width vs depth (1)

In figure 13 a fairly good separation of the classes 'Premium' and 'Very Good' can be seen, while in figure 14 a possible separation of the classes 'Good' and 'Fair' from the rest is shown.

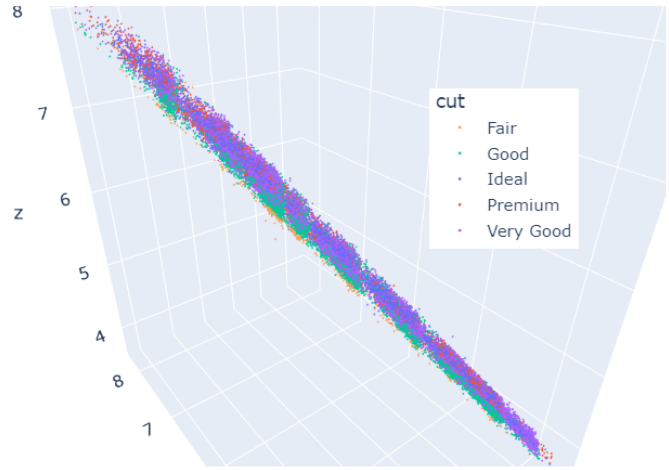

Figure 14: length vs width vs depth (2)

A study of the correlation between the numerical variables and the possible relationship between the target variable and the explanatory variables was made.

The numerical variables taken into account in the linear correlation study were the following:

- carat
- x
- y
- z
- volume
- total percentage of depth
- table
- price

A heatmap elaborated with the values of Pearson's linear correlations can be seen in Figure 15.

The variables length, width, depth, volume, carat, and, to a lesser extent, price, have a very high positive linear correlation. This makes sense, since, in general, diamond dimensions tend to follow certain patterns. The positive correlation of the measurements with the price variable was also predictable, since larger diamonds tend to increase in price.

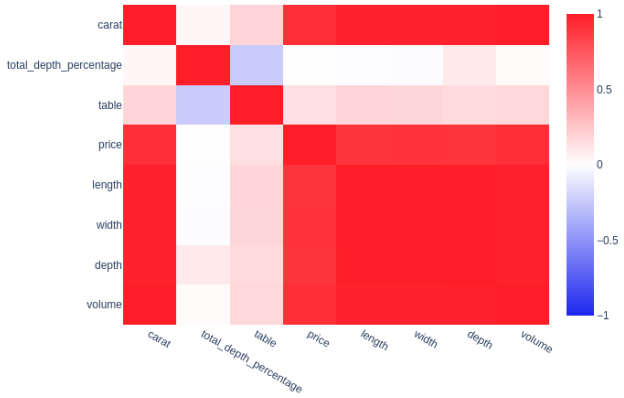

Figure 15: Linear correlations heatmap.

5.2 Models Results

5.2.1 Hyperparameters selection

Parameters	Accuracy	Sensitivity	Specificity	PPV	NPV	F1
lr=0.01 top = [8]	0.713	0.713	0.871	0.697	0.908	0.684
lr=0.01 top = [20]	0.722	0.722	0.874	0.706	0.910	0.698
lr=0.05 top = [8]	0.689	0.689	0.861	0.661	0.902	0.652
lr=0.05 top = [20]	0.665	0.665	0.850	0.634	0.895	0.631

lr=0.01 top = [7, 7]	0.706	0.706	0.869	0.685	0.907	0.672
lr=0.01 top = [8, 4]	0.706	0.706	0.868	0.685	0.908	0.671
lr=0.05 top = [7, 7]	0.671	0.671	0.845	0.629	0.902	0.614
lr=0.05 top = [8, 4]	0.680	0.680	0.850	0.635	0.905	0.619

Table 2: Cross validation metrics for ANN models

The best combination of hyperparameters for the ANN model is a learning rate of 0.01 and a hidden layer of 20 neurons. The worst performing hyperparameters are a learning rate of 0.05 and a topology of two hidden layers of 7 neurons each.

Parameters	Accuracy	Sensitivity	Specificity	PPV	NPV	F1
C=0.1, rbf	0.716	0.716	0.871	0.696	0.909	0.692
C=0.1, poly	0.680	0.680	0.829	0.659	0.906	0.644
C=0.1, sigmoid	0.473	0.473	0.802	0.503	0.794	0.469
C=1, rbf	0.751	0.751	0.882	0.740	0.918	0.737
C=1, poly	0.732	0.732	0.865	0.722	0.913	0.716
C=1, sigmoid	0.463	0.463	0.802	0.508	0.785	0.468
C=10, rbf	0.778	0.778	0.894	0.776	0.923	0.773
C=10, poly	0.757	0.757	0.884	0.750	0.917	0.749
C=10, sigmoid	0.453	0.453	0.816	0.513	0.779	0.468

Table 3: Cross validation metrics for SVM models

The best combination of hyperparameters for the SVM model is a C value of 10 and a rbf kernel . The worst performing hyperparameters based on validation f1-score are the sigmoid kernel, either with C values of 0.1, 1 and 10.

Parameters	Accuary	Sensitivity	Specificity	PPV	NPV	F1
max_depth=5	0.731	0.731	0.875	0.729	0.913	0.706
max_depth=7	0.737	0.737	0.878	0.734	0.913	0.715
max_depth=10	0.746	0.746	0.882	0.739	0.915	0.729
max_depth=12	0.748	0.748	0.885	0.739	0.913	0.734
max_depth=14	0.746	0.746	0.887	0.737	0.909	0.736
max_depth=16	0.741	0.741	0.886	0.731	0.904	0.733

Table 4: Cross validation metrics for Decision Tree models

The best combination of hyperparameters for the Decision Tree model is a max depth value of 14. The worst performing max depth value based on validation f1-score is 5.

Parameters	Accuary	Sensitivity	Specificity	PPV	NPV	F1
k=3	0.661	0.661	0.858	0.641	0.881	0.647

k=5	0.689	0.689	0.865	0.666	0.895	0.669
k=7	0.699	0.699	0.868	0.678	0.899	0.680
k=9	0.704	0.704	0.869	0.683	0.902	0.683
k=11	0.706	0.706	0.869	0.686	0.904	0.685
k=13	0.708	0.708	0.869	0.688	0.905	0.686

Table 5: Cross validation metrics for kNN models

The best combination of hyperparameters for the kNN model is a k value of 13. The worst performing k value based on validation f1-score is 3.

5.2.2 Model selection

Model	Accuracy	Sensitivity	Specificity	PPV	NPV	F1
ANN	0.722	0.722	0.874	0.706	0.910	0.698
SVM	0.778	0.778	0.894	0.776	0.923	0.773
DT	0.746	0.746	0.887	0.737	0.909	0.736
kNN	0.708	0.708	0.869	0.688	0.905	0.686
Ensemble	0.788	0.788	0.898	0.784	0..926	0.783

Table 6: Cross-validation metrics for the best model of each technique

The best validation F-score among all tests was obtained with the ensemble model, therefore we will choose this as the final model.

5.2.3 Final results of models

ANN	Fair	Good	Very Good	Premium	Ideal
Fair	5	23	1	3	0
Good	0	218	109	48	7
Very Good	0	62	353	409	261
Premium	0	0	56	971	145
Ideal	0	2	58	99	1909

Table 7: ANN Confusion Matrix

SVM	Fair	Good	Very Good	Premium	Ideal
Fair	23	5	1	3	0
Good	2	223	149	4	4
Very Good	0	42	695	105	243
Premium	0	0	130	888	154
Ideal	0	1	82	65	1920

Table 8: SVM Confusion Matrix

DT	Fair	Good	Very Good	Premium	Ideal
Fair	20	6	4	2	0
Good	7	251	78	40	6
Very Good	1	52	504	300	228
Premium	2	6	81	935	148
Ideal	0	4	80	103	1881

Table 9: DT Confusion Matrix

kNN	Fair	Good	Very Good	Premium	Ideal
Fair	10	17	3	2	0
Good	2	225	96	50	9
Very Good	0	72	372	376	265
Premium	0	14	101	900	157
Ideal	0	2	58	90	1918

Table 10: kNN Confusion Matrix

Ensemble	Fair	Good	Very Good	Premium	Ideal
Fair	24	4	1	3	0
Good	1	237	135	5	4
Very Good	0	32	715	119	219
Premium	0	0	97	922	153
Ideal	0	1	82	64	1921

Table 11: Ensemble Confusion Matrix

Final Model	Accuracy	Sensitivity	Specificity	PPV	NPV	F1
Final ANN	0.729	0.729	0.878	0.720	0.912	0.707
Final SVM	0.791	0.791	0.898	0.790	0.927	0.787
Final DT	0.758	0.758	0.891	0.753	0.915	0.747
Final kNN	0.722	0.722	0.875	0.709	0.909	0.704
Final Ensemble	0.800	0.800	0.902	0.798	0.929	0.796

Table 12: Test metrics for final models

Table 12 shows the different values of the test metrics for the models selected in cross validation and for the ensemble.

The best model obtained for all the metrics taken into account is the Ensemble model. The model with the worst values for all metrics is kNN.

6. FINAL DISCUSSION

For the prediction of the diamond cut, the Ensemble model performed the best results among the final models.

In general, all the models obtained considerable metric values, so it is considered important to highlight that, although the Ensemble model is the best, another model could be chosen, such as the Decision Tree, since it provides greater explainability for the predictions made and, also, as the time required to run it is considerably less.

If only interested in the correct prediction of the diamond cut, the Ensemble model or the SVM should be chosen.

If we compare the performance of the different final models for the different classes of the target variable we have the following:

- The model that best classifies the Fair class is the Ensemble. The worst model for this class is ANN, which classifies almost all Fair cut diamonds as Good cut.
- The model that best classifies the Good class is the Decision Tree. The worst model for this class is ANN, which classifies many Good cut diamonds as Very Good and Premium cuts.
- The model that best classifies the Very Good class is the Ensemble. The worst model for this class is ANN, which classifies many Very Good cut diamonds as Premium and Ideal cuts.
- The model that best classifies the Premium class is the ANN. The worst model for this class is SVM, which classifies many Premium cut diamonds as Very Good and Ideal cuts.
- The models that better classify the Ideal class are the Ensemble and SVM models. The worst model for this class is Decision Tree, which classifies some Ideal cut diamonds as Very Good and Premium cuts.

As can be seen in the vast majority of models, most of the erroneous predictions deviate from the actual class by one or two levels, indicating that the models are doing a good job even though they are obviously not perfect and make certain errors. The misclassifications in almost all cases do not stray too far from the actual value of the diamond cut.

If the person interested in using these models has a preference for a particular cut class, he/she should consider the option of using the model that best suits the particular class, even though the other classes may not give the best performance.

REFERENCES

[1] "El diamante y sus utilidades." http://geomuseu.upc.edu/index.php/es/el-diamante-y-sus-utilidades/.

[2] "Corte del diamante: Orientación y clasificación de corte de diamante." https://www.tiffany.com.mx/engagement/the-tiffany-guide-to-diamonds/cut/ (accessed Nov. 23, 2022).

[3] S. Brown, "Machine learning, explained," Apr. 21, 2021. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (accessed Nov. 23, 2022).

[4] W. Alsuraihi, E. Al-hazmi, K. Bawazeer, and H. Alghamdi, "Machine Learning Algorithms for Diamond Price Prediction," May 2020, doi: 10.1145/3388818.3393715.

[5] M. Navale, M. Bikchandani, V. Bhosale, P. Bhosale, and S. Ghadge, "Diamond quality assessment system using machine learning approach," vol. 07, no. 04. pp. 874–876, 2020.

[6] G. Sharma, V. Tripathi, M. Mahajan, and A. Kumar Srivastava, "Comparative Analysis of Supervised Models for Diamond Price Prediction," Jan. 2021, pp. 1019–1022, doi: 10.1109/Confluence51648.2021.9377183.

[7] H. Mihir, M. I. Patel, S. Jani, and R. Gajjar, "Diamond Price Prediction using Machine Learning," Dec. 2021, pp. 1–5, doi: 10.1109/C2I454156.2021.9689412.

[8] "Diamonds," 2017. https://www.kaggle.com/datasets/shivam2503/diamonds (accessed Nov. 23, 2022).

[9] Schullin, "All about cut," 2015. https://mydiamondring.com/en/article/756841 (accessed Nov. 23, 2022).

[10] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach. United Kingdom: Pearson Education, 2022.

[11] J. R. Hilera González and V. J. Martínez Hernando, Redes neuronales artificiales: fundamentos, modelos y aplicaciones. Madrid, España: RA-MA, 1995.