Group 5: 40447010S 楊喻文、60847017S 朱苓語、80747003S 林于立

Data Mining Final Group Project:
Diamond Price Range Prediction

Source: https://www.kaggle.com/shivam2503/diamonds

Database Description

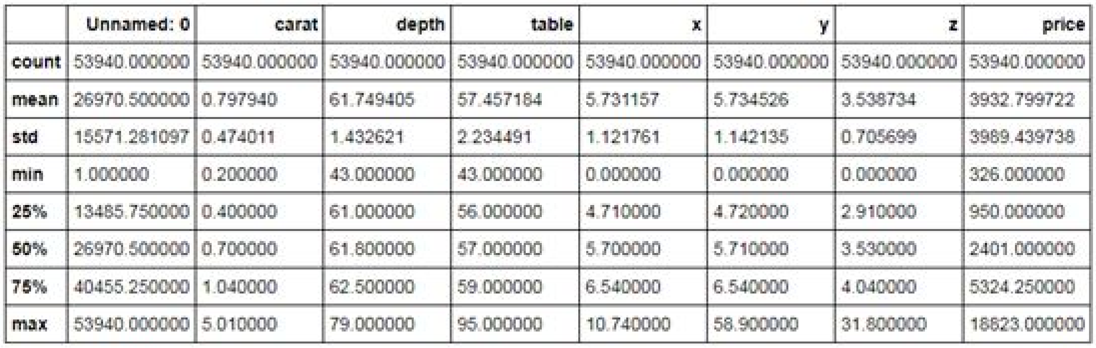
name	type			description
id	nominal	continuous	numerical	index
carat	ratio	continuous	numerical	weight of the diamond
cut	ordinal	discrete	categorical	quality of the cut {fair, good, very good, premium, ideal}
color	ordinal	discrete	categorical	diamond color, {j (worst) - d (best)}
clarity	ordinal	discrete	categorical	a measurement of how clear the diamond is from worst to best: {i1 , si2, si1, vs2, vs1, vvs2, vvs1, if }
depth	ratio	continuous	numerical	total depth percentage depth = z / mean(x, y) = 2 * z / (x + y)
table	ratio	continuous	numerical	width of the top of the diamond relative to the widest point
price	ratio	continuous	numerical	price in us dollars
x	ratio	continuous	numerical	length in mm
y	ratio	continuous	numerical	width in mm
z	ratio	continuous	numerical	depth in mm

Purpose of the Project

The market values of high-end jewelry usually cover an enormous range of prices, but the consumers know little about how the prices are set. Since diamonds are one of the most popular high-end jewelry, our project aims to develop a model to better predict the price range of a diamond by its most common features, which include its weight in carat, color, clarity, and the quality of the cut.

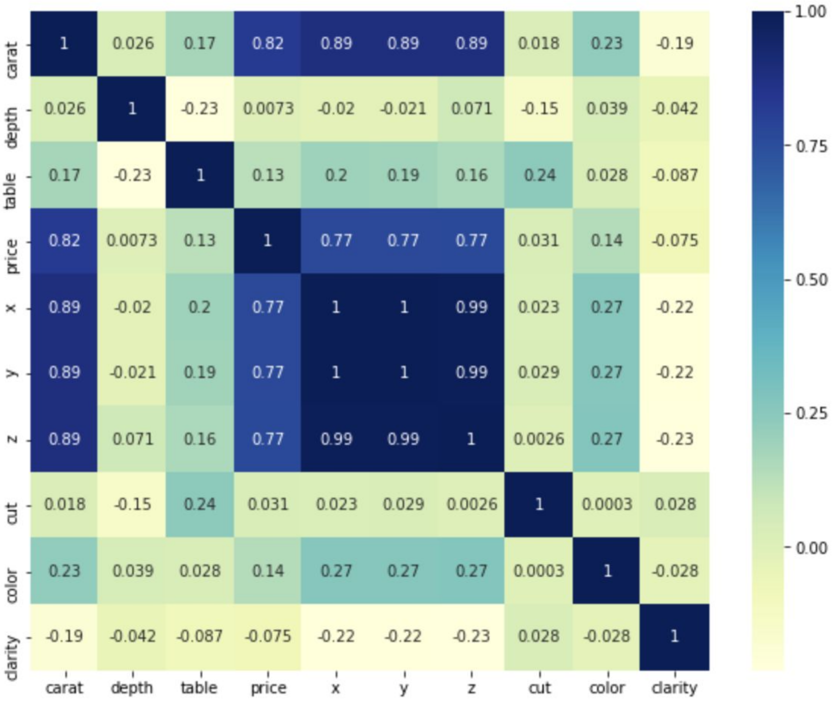
Data Preprocessing

Before we build a model based on the data set, we need to make sure the data is clean and ready for processing. The full details of our raw data set are shown in the table below. We first checked if there were any missing values in the dataset and there were none. However, when we took a closer look at each feature, we found that the minimum of the features that showed a diamond's dimensions ('x', 'y', and 'z') are zero. Since diamonds are three dimensional solid objects, these the value of these three features should not be zero. Therefore, we set aside 20 data that have values of zero in their 'x', 'y', or 'z' feature.



	Unnamed: 0	carat	depth	table	x	y	z	price
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	26970.500000	0.797940	61.749405	57.457184	5.731157	5.734526	3.538734	3932.799722
std	15571.281097	0.474011	1.432621	2.234491	1.121761	1.142135	0.705699	3989.439738
min	1.000000	0.200000	43.000000	43.000000	0.000000	0.000000	0.000000	326.000000
25%	13485.750000	0.400000	61.000000	56.000000	4.710000	4.720000	2.910000	950.000000
50%	26970.500000	0.700000	61.800000	57.000000	5.700000	5.710000	3.530000	2401.000000
75%	40455.250000	1.040000	62.500000	59.000000	6.540000	6.540000	4.040000	5324.250000
max	53940.000000	5.010000	79.000000	95.000000	10.740000	58.900000	31.800000	18823.000000

The target of our prediction is a diamond's price. The feature 'id' is used as an index and has no relation with the price of a diamond; therefore, it was omitted during data preprocessing. While observing the heatmap (shown below) of the remaining features, we noticed that some features, such as 'table' and 'depth', have a low correlation with 'price' feature; therefore, we deleted those features. Furthermore, upon researching online resources, we found that most diamond seller websites do not include the information on a diamond's length, width, or depth, which are named 'x', 'y', and 'z' in our data set. Since we hope that our model can help general consumers, we believe that it is better to develop a model that predicts the price range of a diamond based solely on its weight in carat, color, clarity and the quality of the cut. Because the features 'x', 'y', and 'z' would not be used when building our models, those previously set-aside data that have a value of 0s in these three features were added back in.



'Carat', 'cut', 'color', and 'clarity', commonly known as '4Cs', are the most well-known features of a diamond, which would be used as predictors in our model. For the feature 'carat', we kept the original feature and value while creating new feature 'carat_discrete', in which we discretize the continuous value of 'carat' into 3 categories. As for the remaining prediction features, we first binarized them into new features, and then we converted the values into numbers in order to better show its ordinal characteristic, with the value ‘1’ being the best quality. The quality of these three features decreases while the value increases.

The only remaining feature to be pre-processed was ‘price’, which would be the target of our classification model. We know that the heavier a diamond is, the higher its price will be. Because we want our model to apply to different sizes of diamonds, we calculated each value of the new feature ‘priceperpoint’, which indicates a diamond’s unit price per point (1 carat equals 100 points) in US dollars. Lastly, since we were about to develop a classification model, we discretized the ‘priceperpoint’ feature into 4 categories: below 21, between 21 and 42, between 42 and 63, and above 63, which represents a low-price range, medium-low-price range, medium-high-price range, and high-price range respectively. The data preprocessing of every feature is shown on the table below:

<table>
<tr><th><i>Feature</i></th><th><i>Data Preprocessing</i></th></tr>
<tr><td>id</td><td>● irrelevant: deleted</td></tr>
<tr><td>carat</td><td>● the original feature kept
● new feature ‘carat_discrete’ created: discretized ‘carat’ into 3 categories
<table>
<tr><td>Range(x)</td><td>category</td><td>number</td></tr>
<tr><td>0 ≤ x &lt; 0.5</td><td>1</td><td>17674</td></tr>
<tr><td>0.5 ≤ x &lt; 1</td><td>2</td><td>17206</td></tr>
<tr><td>1 ≤ x</td><td>3</td><td>19060</td></tr>
</table></td></tr>
<tr><td>cut</td><td>● the original value of 5 categories converted from string to numerical
● binarize 5 categories into another 5 features</td></tr>
<tr><td>color</td><td>● the original value of 7 categories converted from string to numerical
● binarize 7 categories into another 7 features</td></tr>
<tr><td>clarity</td><td>● the original value of 8 categories converted from string to numerical
● binarize 8 categories into another 8 features</td></tr>
<tr><td>depth</td><td>● low correlation: deleted</td></tr>
<tr><td>table</td><td>● low correlation: deleted</td></tr>
<tr><td>price</td><td>● target data: <b>priceperoint</b> = price / (carat x 100) , discretized into 4 categories
<table>
<tr><td>Range(x)</td><td>category</td><td>number</td></tr>
<tr><td>0 ≤ x &lt; 21</td><td>1</td><td>7432</td></tr>
<tr><td>21 ≤ x &lt; 42</td><td>2</td><td>26551</td></tr>
<tr><td>42 ≤ x &lt; 63</td><td>3</td><td>12474</td></tr>
<tr><td>63 ≤ x</td><td>4</td><td>7483</td></tr>
</table></td></tr>
</table>

x	● not commonly used to determine diamond price: deleted
y	● not commonly used to determine diamond price: deleted
z	● not commonly used to determine diamond price: deleted

Model Development

We chose 7 different classification models: KNN, ANN, Random Forest, Decision Tree, AdaBoost, Naïve Bayes Classifiers (GaussianNB, MultinomialNB), and SVM(LinearSVC). Each member of our group tried and develop 2 to 3 models, from which each of us would take the best model and compare it with the result of each other.

Name	*Models*	*Model with Better / the Best Result*
喻文	● Naïve Bayes Classifiers (GaussianNB, MultinomialNB) ● SVM (LinearSVC)	SVM (LinearSVC)
苓語	● KNN ● ANN	KNN
于立	● Random Forest ● Decision Tree ● Adaboost	Random Forest

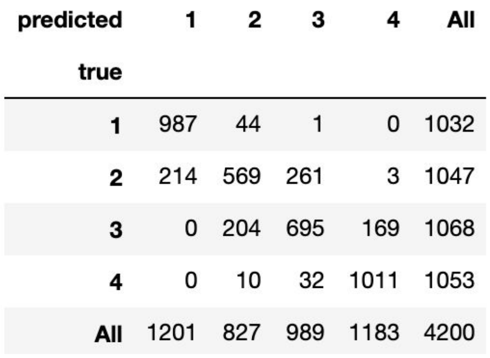
Model Comparision

The result of the three better classification models is shown in the table below. The confusion metrics of each model shows the result of how each model classified the price ranges of diamonds with different features.

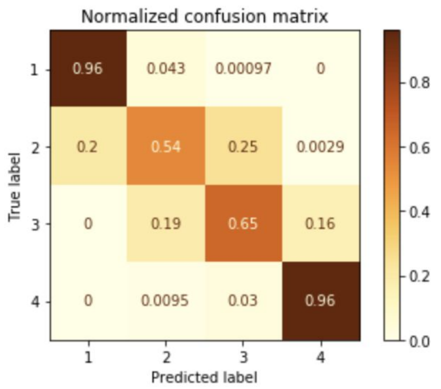
We found that though SVM performed the best in predicting diamonds with the highest and the lowest price-range (recall of category 1 and 4: 0.96), it performed poorly in predicting the price range of diamonds with medium price range; the recall of category 2 was only 0.54 and category 3 with only 0.65. The overall accuracy of SVM model was 0.78.

KNN and Random Forest had similar results. All of the recalls of these two classification models were above 0.8. The overall accuracy of KNN was 0.86, and the overall accuracy was 0.89. Random Forest slightly outperformed KNN.

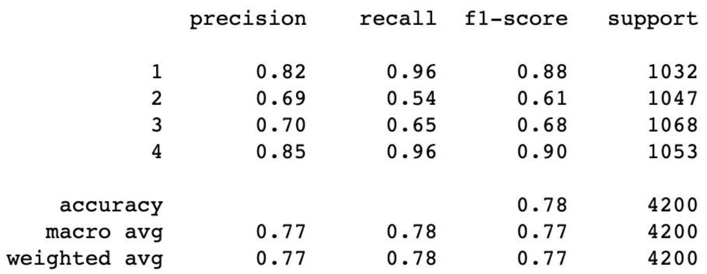
Models	Evaluation Method
SVM (LinearSVC)	Confusion Metrics



predicted	1	2	3	4	All
true					
1	987	44	1	0	1032
2	214	569	261	3	1047
3	0	204	695	169	1068
4	0	10	32	1011	1053
All	1201	827	989	1183	4200

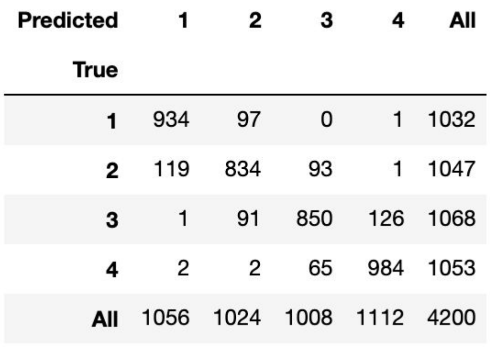


Classification Report

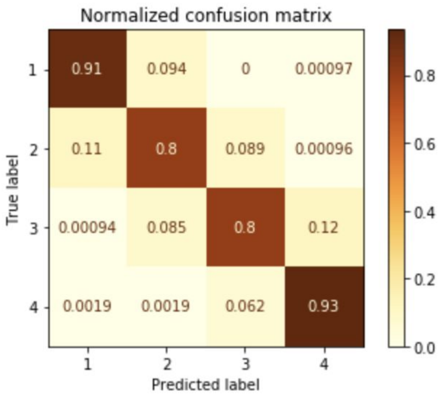


```
              precision    recall  f1-score   support

           1       0.82      0.96      0.88      1032
           2       0.69      0.54      0.61      1047
           3       0.70      0.65      0.68      1068
           4       0.85      0.96      0.90      1053

    accuracy                           0.78      4200
   macro avg       0.77      0.78      0.77      4200
weighted avg       0.77      0.78      0.77      4200
```

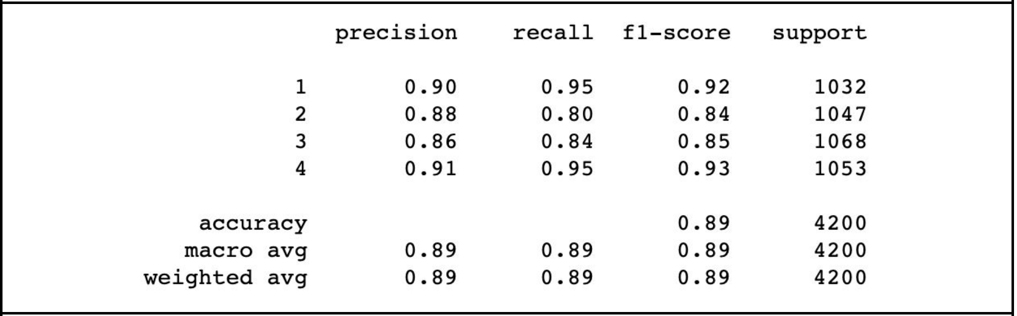
KNN	Confusion Metrics



Predicted	1	2	3	4	All
True					
1	934	97	0	1	1032
2	119	834	93	1	1047
3	1	91	850	126	1068
4	2	2	65	984	1053
All	1056	1024	1008	1112	4200

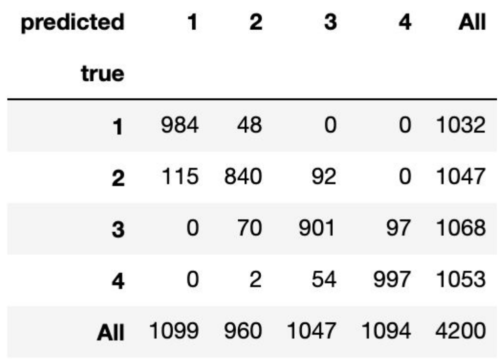


Models	Evaluation Method
KNN	Classification Report

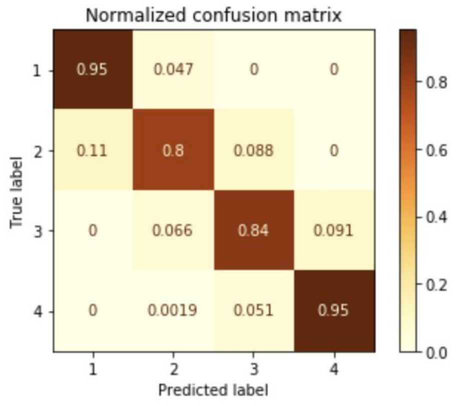


```
              precision    recall  f1-score   support

           1       0.90      0.95      0.92      1032
           2       0.88      0.80      0.84      1047
           3       0.86      0.84      0.85      1068
           4       0.91      0.95      0.93      1053

    accuracy                           0.89      4200
   macro avg       0.89      0.89      0.89      4200
weighted avg       0.89      0.89      0.89      4200
```

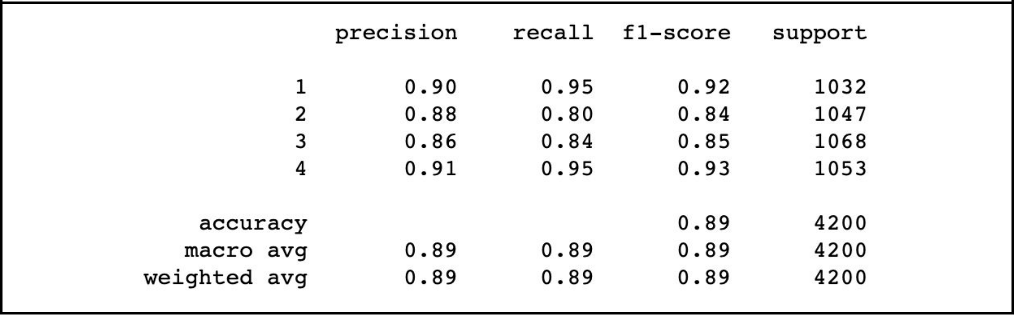
Models	Evaluation Method
Random Forest	Confusion Metrics



predicted	1	2	3	4	All
true					
1	984	48	0	0	1032
2	115	840	92	0	1047
3	0	70	901	97	1068
4	0	2	54	997	1053
All	1099	960	1047	1094	4200



Classification Report

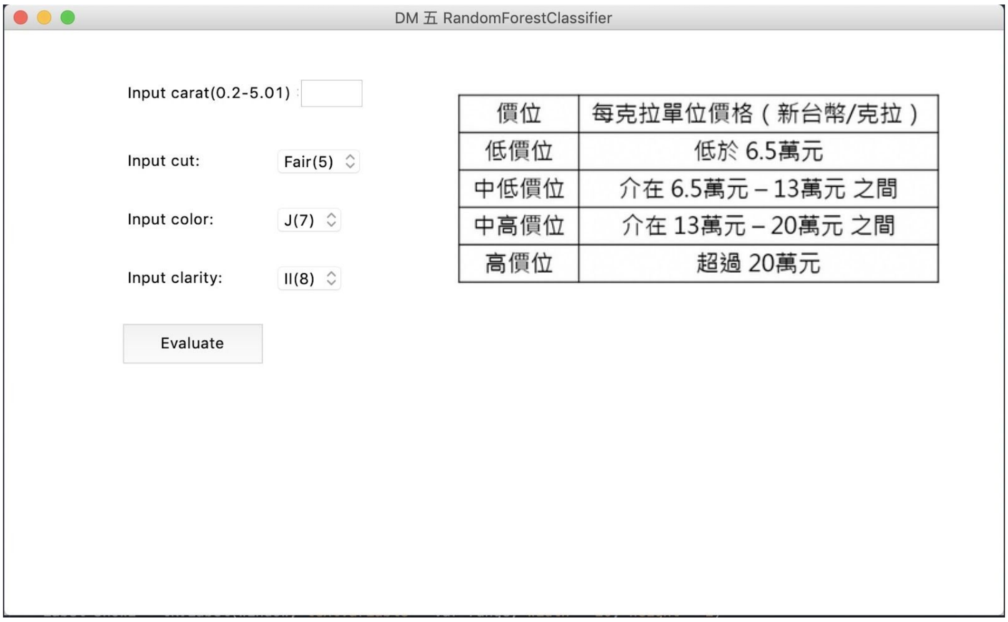


```
              precision    recall  f1-score   support

           1       0.90      0.95      0.92      1032
           2       0.88      0.80      0.84      1047
           3       0.86      0.84      0.85      1068
           4       0.91      0.95      0.93      1053

    accuracy                           0.89      4200
   macro avg       0.89      0.89      0.89      4200
weighted avg       0.89      0.89      0.89      4200
```

Discussion

As shown in the previous section, the SVM model underperformed the other two models. This might be because SVM works better for data sets with features that have a certain level of correlation with each other. However, our data set does not possess this characteristic. The main features in our dataset (carat, cut, color, and clarity) have low correlations with each other, e.g. a diamond with an ideal cut may do poorly in its color or clarity.

Both KNN and Random Forest Classifiers have excellent results, with accuracy over 0.8. Since the overall accuracy of Random Forest Classifier is slightly higher than KNN, we went with Random Forest Classifier while building our 'diamond price range prediction model'. The layout of our classifier is shown below: The users can put in the criteria of each diamond. Our classifier will predict which price range the diamond falls into, and compute the estimated price of that diamond.



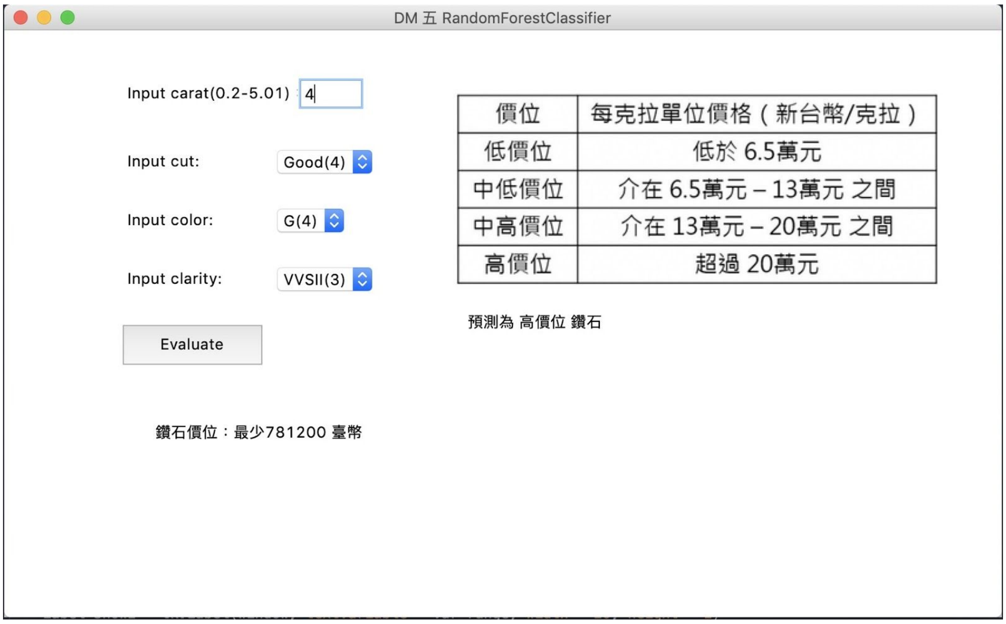
Input carat(0.2-5.01)

Input cut: Fair(5)

Input color: J(7)

Input clarity: II(8)

價位	每克拉單位價格（新台幣/克拉）
低價位	低於 6.5萬元
中低價位	介在 6.5萬元－13萬元 之間
中高價位	介在 13萬元－20萬元 之間
高價位	超過 20萬元



Input carat(0.2-5.01) 4

Input cut: Good(4)

Input color: G(4)

Input clarity: VVSII(3)

價位	每克拉單位價格（新台幣/克拉）
低價位	低於 6.5萬元
中低價位	介在 6.5萬元－13萬元 之間
中高價位	介在 13萬元－20萬元 之間
高價位	超過 20萬元

預測為 高價位 鑽石

鑽石價位：最少781200 臺幣