Diamond Price Prediction

פרויקט גמר - מבוא למדעי הנתונים

שירן אהל - **207332867**

מבוא

יהלום הוא מינרל מוצק העשוי מפחמן נקי. זהו המינרל המוצק והבהיר ביותר שקיים בעולם.
יהלומים משמשים בעיקר לתכשיטים.
כמו הרבה דברים בחיים (למשל מכוניות), ליהלומים יש מחירון ידוע ונפוץ, אשר קובע את המחיר של כל יהלום בהתבסס על הפרמטרים הבאים:

1. Carat - מידת משקל עבור יהלומים.
2. Color - צבע היהלום.
3. Clarity - מידת הניקיון של היהלום.
4. Cut - איכות החיתוך של היהלום (הגורם היחיד שאינו מושפע מהטבע, אלא מהלוטש).

עם זאת, יש נתונים נוספים על היהלומים, שאינם מקבלים התייחסות במחירון:

1. פלורסנציה - זריקת גוון תחת תנאי תאורה שונים.
2. סימטריה - עד כמה טובים ומדוייקים הצדדים של היהלום.
3. צחצוח - מידת החלקות של כל פאות היהלום.
4. מידת עומק היהלום (באחוזים).
5. מידת לוח היהלום (באחוזים).

שאלת המחקר

לאחר שחקרתי את הנושא, ונוכחתי לדעת שיש נתונים נוספים רבים אשר לא נלקחים בחשבון בקביעת מחיר היהלום, השאלה עליה רציתי לענות היא:

האם ניתן לחזות את מחיר היהלום בהתבסס על כל הנתונים שלו?

שאלות נוספות שניתן לענות עליהן (באופן מדגמי כמובן) מניתוח מאגר הנתונים:

1. **איזה צבע יהלום הכי נפוץ?**
2. **איזה ניקיון יהלום הכי נפוץ?**
3. **איזה איכות חיתוך יהלום הכי נפוצה?**

שלבי הפרויקט

1. Data Acquisition
2. Data Cleaning
3. EDA (Visualizations & Research)
4. Model train and prediction
5. Summary and conclusions

1. Data Acquisition

מצאתי את האתר Brilliant Earth אשר מוכר יהלומים אונליין, וכתבתי סקריפט בשביל לבצע Scraping על האתר, ובכך אספתי נתונים על כ - 5000 יהלומים עגולים בלבד, בטווחים של 0.90 עד 1.20 קראט.

לינק לסקריפט:

https://github.com/ShiranOhel/hit_data_science_project/blob/main/scripts/fetch_diamonds_data.py

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4938 entries, 0 to 4937
Data columns (total 18 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   stock_number  4938 non-null   object
 1   gemstone      4938 non-null   object
 2   origin        4938 non-null   object
 3   price         4938 non-null   object
 4   carat         4938 non-null   float64
 5   shape         4938 non-null   object
 6   cut           4938 non-null   object
 7   color         4938 non-null   object
 8   clarity       4938 non-null   object
 9   measurements  4938 non-null   object
 10  table         4938 non-null   object
 11  depth         4938 non-null   object
 12  symmetry      4938 non-null   object
 13  polish        4938 non-null   object
 14  girdle        4938 non-null   object
 15  culet         4933 non-null   object
 16  fluorescence  4889 non-null   object
 17  diamond_id    4874 non-null   float64
dtypes: float64(2), object(16)
memory usage: 694.5+ KB
```

	stock_number	gemstone	origin	price	carat	shape	cut	color	clarity	measurements	table	depth	symmetry	polish	girdle	culet	fluorescence
0	5763178A	Natural, untreated diamond	Botswana Sort	$2,470	0.9	Round	Good	F	SI2	5.96mm x 5.91mm x 3.89mm	63.0%	65.5%	Good	Very Good	6.5	Pointed	None
1	6450036Y	Natural, untreated diamond	Botswana Sort	$2,470	0.9	Round	Very Good	J	VS2	6.04mm x 5.99mm x 3.88mm	59.0%	64.5%	Excellent	Very Good	Slightly Thick - Thick	None	Faint
2	6057071A	Natural, untreated diamond	Botswana Sort	$2,470	0.9	Round	Very Good	I	SI2	6.00mm x 5.93mm x 3.85mm	57.0%	64.6%	Very Good	Very Good	Thick	None	None
3	5073535A	Natural, untreated diamond	Botswana Sort	$2,470	0.9	Round	Very Good	H	SI2	6.14mm x 6.09mm x 3.85mm	57.0%	63.0%	Excellent	Excellent	4.5	Pointed	None
4	5985294A	Natural, untreated diamond	Botswana Sort	$2,480	0.9	Round	Good	H	SI1	5.84mm x 5.79mm x 3.91mm	59.0%	67.3%	Very Good	Excellent	Very Thick	None	None

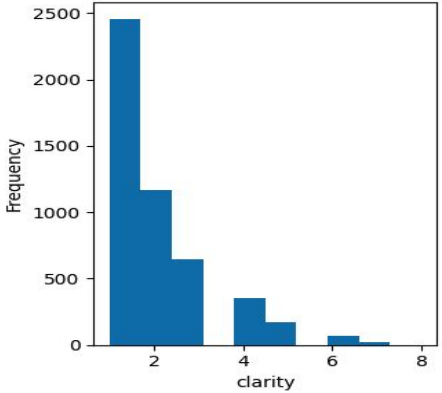
2. Data Cleaning

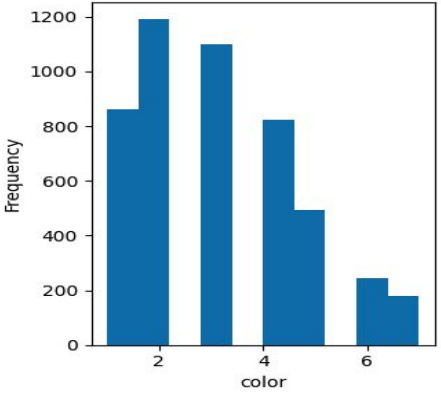
היו מספר עמודות לא רלוונטיות למודל לכן הורדתי אותן, וכמו כן הפכתי עמודות קטגוריאליות למספריות
וגם עמודות הכוללות טקסט ($, %) למספריות.

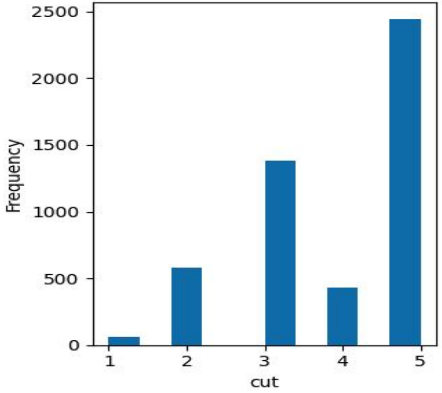
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4899 entries, 0 to 4898
Data columns (total 12 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   price      4899 non-null   float64
 1   carat      4899 non-null   float64
 2   cut        4899 non-null   int64
 3   color      4899 non-null   int64
 4   clarity    4899 non-null   int64
 5   table_pct  4899 non-null   float64
 6   depth_pct  4899 non-null   float64
 7   symmetry   4899 non-null   int64
 8   polish     4899 non-null   int64
 9   x          4899 non-null   float64
 10  y          4899 non-null   float64
 11  z          4899 non-null   float64
dtypes: float64(7), int64(5)
memory usage: 459.4 KB
```

	price	carat	cut	color	clarity	table_pct	depth_pct	symmetry	polish	x	y	z
0	2470.0	0.9	2	5	1	63.0	65.5	2	2	5.96	5.91	3.89
1	2470.0	0.9	3	1	3	59.0	64.5	4	2	6.04	5.99	3.88
2	2470.0	0.9	3	2	1	57.0	64.6	3	2	6.00	5.93	3.85
3	2470.0	0.9	3	3	1	57.0	63.0	4	3	6.14	6.09	3.85
4	2480.0	0.9	2	3	2	59.0	67.3	3	3	5.84	5.79	3.91

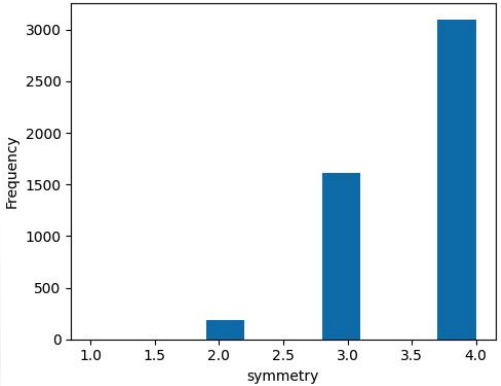
3. EDA

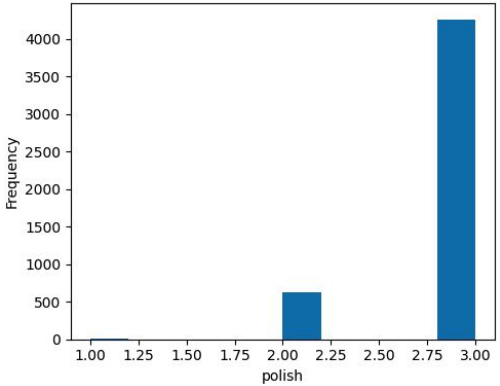




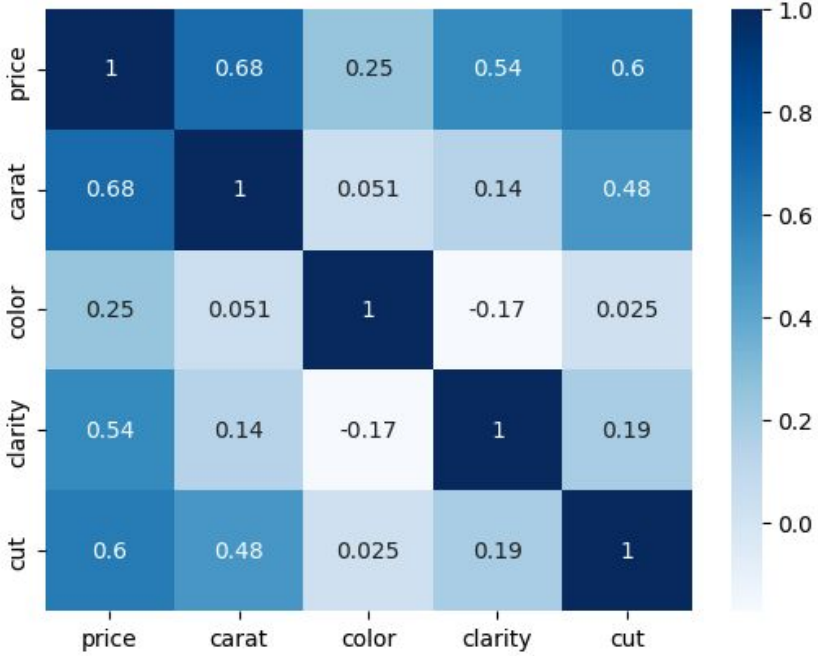


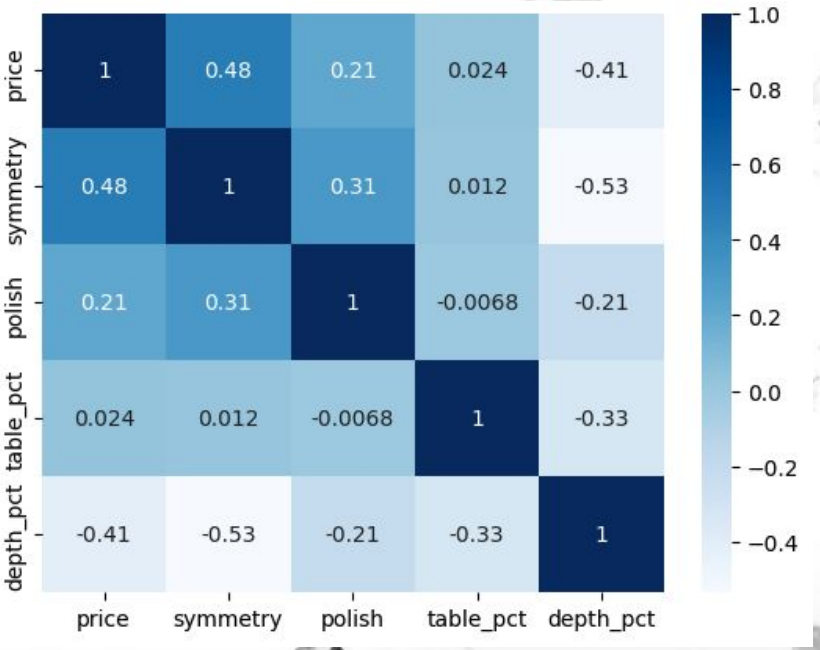
בדיקת תדירות עבור תכונות מרכזיות





בדיקת מתאם בין מחיר היהלום לתכונות השונות שלו





4. Model train and prediction

המודל הנבחר

רגרסיה לינארית

Linear Regression

סוג המודל

מודל חיזוי מבוסס למידה מונחית

Supervised learning predictive model

הסבר המודל

כיוון שהחיזוי הוא עבור ערך מספרי ממשי אינסופי (מחיר), רגרסיה לינארית הוא המודל המתאים ביותר

5. Summary and conclusions

לאחר ביצוע מספר הרצות של המודל, כאשר בכל ריצה השתמשתי ב features שונים מתוך ה dataset,
הגעתי למסקנה שהחיזוי המדויק ביותר מתקבל כאשר משתמשים רק בארבעת התכונות של היהלום
(the diamond 4Cs) אשר המחירון מתבסס עליהן מלכתחילה.

```
model_train_and_predict(df[['price', 'carat', 'color', 'clarity', 'cut']], 'price', 0.25, 7)
```

```
      y_test  y_predicted
0     4950.0  4531.802993
1     3720.0  3491.837357
2     7330.0  5691.437178
3     2960.0  3887.014687
4     6180.0  6903.198049
...      ...          ...
1220  9960.0  7681.905394
1221  7960.0  7040.933506
1222  4830.0  4635.847615
1223  4160.0  4449.200612
1224  3560.0  3979.251799

[1225 rows x 2 columns]
Model r2 score is 0.8083154889312754
```

שימוש בכל ה features הקיימים ב dataset הניב תוצאה דומה:

```
model_train_and_predict(df, 'price', 0.25, 7)
```

```
      y_test  y_predicted
0     4950.0  4525.611696
1     3720.0  3376.848786
2     7330.0  5686.642339
3     2960.0  3795.031689
4     6180.0  6869.920258
...      ...          ...
1220  9960.0  7671.247093
1221  7960.0  7040.771144
1222  4830.0  4662.611090
1223  4160.0  4520.784106
1224  3560.0  3939.163535

[1225 rows x 2 columns]
Model r2 score is 0.809260893022758
```

אך ניסיונות להשתמש בהרכבים שונים של ה features הביאו תוצאות נמוכות יותר:

```
model_train_and_predict(df[['price', 'carat', 'clarity', 'cut', 'symmetry']], 'price', 0.25, 7)
```

```
      y_test  y_predicted
0     4950.0  4326.882359
1     3720.0  3248.910968
2     7330.0  5664.021295
3     2960.0  3662.030751
4     6180.0  5664.021295
...      ...          ...
1220  9960.0  6789.091187
1221  7960.0  6568.441043
1222  4830.0  4440.925291
1223  4160.0  4779.092232
1224  3560.0  4067.250845

[1225 rows x 2 columns]
Model r2 score is 0.7294075844808778
```

```
model_train_and_predict(df[['price', 'carat', 'clarity', 'cut']], 'price', 0.25, 7)
```

```
      y_test  y_predicted
0     4950.0  4326.543060
1     3720.0  3250.895122
2     7330.0  5663.851164
3     2960.0  3658.868017
4     6180.0  5663.851164
...      ...          ...
1220  9960.0  6789.227140
1221  7960.0  6568.322353
1222  4830.0  4437.974462
1223  4160.0  4778.778655
1224  3560.0  4066.840912

[1225 rows x 2 columns]
Model r2 score is 0.7294099699535548
```

לסיכום, ניתן להבין שאכן ארבעת התכונות המרכזיות של היהלום:
Carat, Color, Clarity, Cut הן אכן המשפיעות ביותר בקביעת מחיר היהלום.

אולם מצד שני, הדבר אינו מדויק במאת האחוזים, כאשר אפילו לשני יהלומים עם אותם נתונים מרכזיים דומים, עלולים להיות מחירים שונים.