

Prediction Diamond Price

โดย

63010139 นายจิรภัทร แก้วส่งแสง

63010235 นายชินาธิป มีสวนนิล

63010326 นายณัฐพงษ์ นาคสามัคคี

63010354 นายดิษฐพงษ์ จรัสชัยโรจน์

โครงการนี้เป็นส่วนหนึ่งของการศึกษา

วิชา 01076032 ELEMENTARY DIFFERENTIAL EQUATIONS AND

LINEAR ALGEBRA

ปีการศึกษา 2564

บทคัดย่อ

ชื่อโครงการ	Prediction Diamond Price
ชื่อผู้จัดทำ	นายจิรภัทร แก้วส่งแสง นายชินาธิป มีสวนนิล นายณัฐพงษ์ นาคสามัคคี และนายดิษฐ์พงษ์ จรัสชัยโรจน์
สถานศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

โครงการ Prediction Diamond Price มีวัตถุประสงค์เพื่อที่จะสร้างโมเดลที่จะสามารถทำนายราคาของเพชร ด้วยองค์ประกอบและลักษณะต่างๆของเพชรจากข้อมูลที่ได้รับมา ซึ่งนำไปสู่โมเดลที่มีประสิทธิภาพในการทำนายราคาของเพชร และสามารถนำโมเดลไปใช้ในการประเมินราคาเบื้องต้นได้

โดยมีขั้นตอนการดำเนินงานดังนี้คือ 1) Cleaning Data 2) ศึกษาเทคนิคและทฤษฎีที่เกี่ยวข้อง 3) ออกแบบองค์ประกอบและคิอัลกอริทึมที่จะนำมาใช้งาน 4) ทำโมเดลตัวทดสอบตามอัลกอริทึมที่ได้มา 5) แก้ไขข้อผิดพลาดต่างๆของโมเดลที่ได้ 6) ทดสอบประสิทธิภาพของโมเดลแต่ละตัวที่ได้ทำขึ้นมา 7) คัดเลือกโมเดลที่จะนำมาใช้งาน 8) พัฒนาประสิทธิภาพของโมเดลที่ได้เลือกมา ให้ได้ค่า MAPE (Mean Absolute Percentage Error) มีค่าน้อยที่สุด

การทดสอบประสิทธิภาพของโมเดลแต่ละโมเดลแบ่งเป็น 2 ตอนได้แก่ 1) ค่าของ R-Squared 2) ค่า MAPE (Mean Absolute Percentage Error)

สารบัญ

บทคัดย่อ	ก
สารบัญ	ข
บทที่ 1 บทนำ	1
ที่มาและความสำคัญของโครงการ	1
วัตถุประสงค์	1
ขอบเขตของการทำโครงการ	1
ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ภาพรวมการออกแบบระบบ	3
ภาพรวมขั้นตอนการทำงานของระบบ	3
รายละเอียดข้อมูลที่เกี่ยวข้อง	4
อธิบายขั้นตอนย่อยแต่ละขั้น	5
บทที่ 3 การประยุกต์ใช้ทฤษฎี	10
การประยุกต์ใช้ทฤษฎีเวกเตอร์	10
การประยุกต์ใช้ทฤษฎีเมทริกซ์	11
บทที่ 4 ผลการทดลอง	12
ผลการทดลองชุดที่ 1	12
ผลการทดลองชุดที่ 2	13
ผลการทดลองชุดที่ 3	14
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	15
สรุปผลการทดลอง	15
ข้อเสนอแนะ	15
รายการอ้างอิง	16

ภาคผนวก	17
ภาคผนวก ก	17
ภาคผนวก ข	18

บทที่ 1

บทนำ

ที่มาและความสำคัญของโครงการ

ในบางครั้งการคำนวณราคาของเพชรสำหรับบุคคลที่ไม่เชี่ยวชาญกลไกและราคาตลาดนั้นเป็นเรื่องที่ทำได้ยาก ดังนั้นเราจึงได้คิดค้นและอยากนำเสนอ วิธีการแก้ไขปัญหาโดยใช้โมเดลคำนวณราคาของเพชรจากองค์ประกอบและรายละเอียดต่างๆ เพื่อให้บุคคลที่สนใจสามารถที่จะทำนายราคาของเพชร เพื่อทราบราคาเพชรเบื้องต้นในการลงทุนซื้อ-ขายได้ รวมถึงอาจเป็นตัวช่วยในการตัดสินใจต่างๆได้อีกด้วย

ผู้จัดทำโครงการ จึงได้คิดค้นและเสนอวิธีการแก้ปัญหาที่เกิดขึ้น จึงคิดที่จะทำโครงการเรื่องนี้ขึ้น โดยโมเดลของการทำนายราคาเพชรตัวนี้ มีการกระบวนการทำงานโดยรับข้อมูลรายละเอียดต่างๆของเพชรที่ผู้ใช้ส่งเข้ามา และทำการคำนวณในโมเดลที่สร้างขึ้นมาไว้เพื่อใช้ในการทำนายราคาของเพชรให้มีค่าใกล้เคียงที่สุด และสามารถนำโมเดลไปใช้ในการทำนายราคาเบื้องต้นได้

วัตถุประสงค์

1. เพื่อศึกษาการนำ Linear algebra มาประยุกต์ใช้ในการทำโมเดลทำนาย
2. เพื่อสร้างโมเดล Prediction Diamond Price ให้มีประสิทธิภาพในการทำนายราคาของเพชรได้แม่นยำ

ขอบเขตของการทำโครงการ

ขอบเขตด้านข้อมูล

1. ชุดข้อมูล “Diamonds” จาก <https://www.kaggle.com/shivam2503/diamonds>

ขอบเขตด้านระยะเวลาในการดำเนินโครงการ

1. ระยะเวลาในการดำเนินการ 28 สิงหาคม 2564 – 19 พฤศจิกายน 2564

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำโมเดล Prediction Diamond Price มาใช้ในการทำนายได้จริง
2. สามารถทำนายราคาของเพชรที่มีความแม่นยำออกมาได้

บทที่ 2

ภาพรวมการออกแบบระบบ

ภาพรวมขั้นตอนการทำงานของระบบ

ขั้นที่ 1 การเตรียมข้อมูล

คณะผู้จัดทำได้ดำเนินการค้นหาแหล่งข้อมูลจาก <https://www.kaggle.com/shivam2503/diamonds> เพื่อใช้ในการดำเนินการจัดทำ โมเดล Prediction Diamond Price ในครั้งนี้

ต่อมาได้ทำการ Clean data โดยการตัดข้อมูลในส่วนที่ไม่ต่อเนื่อง และชุดข้อมูลที่เป็นข้อมูลแบบสุดโต่าง

ขั้นที่ 2 การศึกษาทฤษฎีที่เกี่ยวข้อง

1. Linear regression

Linear regression คืออะไร

Linear regression เป็นโมเดลความสัมพันธ์ระหว่างตัวแปรกับสิ่งที่เราสนใจ เช่นปัจจัยต่างๆ ทางธุรกิจกับยอดขายในไตรมาสต่างๆ หรืออาจจะเป็นส่วนประกอบต่างๆ ในสินค้ากับเกรดหรือคุณภาพของสินค้า เป็นต้น โดย linear regression นั้นเป็นโมเดลที่ได้รับความนิยมมากที่สุด เนื่องจากเป็นโมเดลที่เข้าใจง่าย อีกทั้งยังสามารถนำไปใช้งานจริงได้อย่างรวดเร็วอีกด้วย

2. Pearson similarity (correlation)

Pearson similarity (correlation) คืออะไร

Pearson similarity (correlation) เป็นเทคนิคที่ใช้ในการหาขนาดและทิศทางของความสัมพันธ์เชิงเส้นตรงระหว่างตัวแปรที่มีมาตรวัดแบบช่วง (interval scale) หรืออัตราส่วน (Ratio scale) สองตัว หรือเรียกตัวแปรประเภทนี้ว่าตัวแปรเชิงปริมาณ ค่าที่ได้เรียกว่า "สัมประสิทธิ์สหสัมพันธ์" โดยปกติจะมีค่าอยู่ระหว่าง -1.00 ถึง 1.00

ขั้นที่ 3 ออกแบบและปฏิบัติการ

หลังจากที่ได้ทำการ Clean data เรียบร้อยแล้ว จากนั้นเป็นการนำข้อมูลเข้าสู่โปรแกรม เพื่อนำมาสร้างโมเดลในการทำนายราคา

ขั้นที่ 4 ทดสอบและปรับปรุงแก้ไข

วิธีการที่ 1 ค่าของ R-Squared

วิธีการที่ 2 ค่า MAPE (Mean Absolute Percentage Error) ยิ่งมีค่าเข้าใกล้ 0 ยิ่งมีประสิทธิภาพสูง

ขั้นที่ 5 การประเมินผลและแสดงผล

หลังจากการปรับปรุงจนใช้งานได้ดีตามวิธีการที่ออกแบบแล้วก็นำมาประเมินผลโดยรวม และนำเข้าสู่ Website เพื่อแสดงผล

รายละเอียดข้อมูลที่เกี่ยวข้อง

ข้อมูลที่เรานำมาใช้มีแหล่งที่มาจาก <https://www.kaggle.com/shivam2503/diamonds> โดยเป็นข้อมูลเกี่ยวกับรายละเอียดของเพชร และราคาของเพชร

อธิบายขั้นตอนย่อยแต่ละขั้น

ขั้นตอนย่อยของขั้นที่ 1 การเตรียมข้อมูล

1. นำ data ที่ได้มาเข้าสู่โปรแกรม Visual Studio Code
2. นำ data ที่ได้มาทำการ Clean ในโปรแกรมโดยการเปลี่ยนจาก ข้อมูลเชิงคุณภาพให้เป็นตัวเลขเพื่อให้
ง่ายต่อการใช้งาน และทำการตัด data ที่สุดโต่งรวมถึง data ที่ไม่ต่อเนื่องด้วย

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53940 entries, 0 to 53939
Data columns (total 10 columns):
#   Column  Non-Null Count  Dtype  
---  -
0   carat    53940 non-null   float64
1   cut      53940 non-null   object  
2   color    53940 non-null   object  
3   clarity  53940 non-null   object  
4   depth    53940 non-null   float64
5   table    53940 non-null   float64
6   price    53940 non-null   int64   
7   x        53940 non-null   float64
8   y        53940 non-null   float64
9   z        53940 non-null   float64
dtypes: float64(6), int64(1), object(3)
memory usage: 4.1+ MB
```

```
new_cut = {"Ideal": 0, "Premium": 1, "Very Good": 2, "Good": 3, "Fair": 4}
data["cut"].replace(new_cut, inplace=True)
data.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	0	1	7	61.5	55.0	326	3.95	3.98	2.43
1	0.21	1	1	6	59.8	61.0	326	3.89	3.84	2.31
2	0.23	3	1	4	56.9	65.0	327	4.05	4.07	2.31
3	0.29	1	5	5	62.4	58.0	334	4.20	4.23	2.63
4	0.31	3	6	7	63.3	58.0	335	4.34	4.35	2.75

ขั้นตอนย่อยของขั้นที่ 2 การศึกษาทฤษฎีที่เกี่ยวข้อง

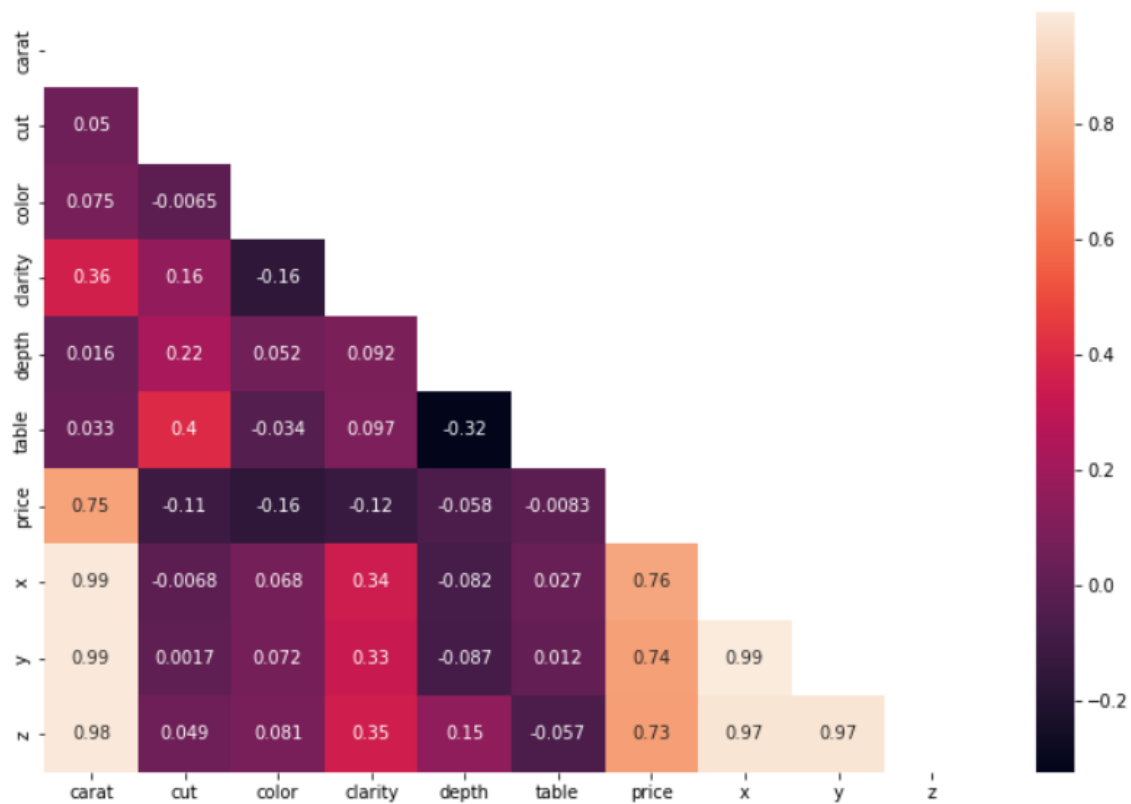
$$\text{similarity}(A, B) = \frac{\text{cov}(A, B)}{\sigma_A \sigma_B} = \frac{\sum_{i=1}^n (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^n (A_i - \bar{A})^2 (B_i - \bar{B})^2}}$$

คณะผู้จัดทำได้ดำเนินการค้นหาข้อมูลจากแหล่งต่างๆ เพื่อใช้ในการจัดทำโมเดล Prediction Diamond Price ในครั้งนี้ ซึ่งได้ทำการศึกษาและสืบค้นข้อมูลจากอินเทอร์เน็ต และได้รวบรวมข้อมูลจากแหล่งต่างๆ เพื่อใช้ในการดำเนินการ และศึกษาข้อมูล

ขั้นตอนย่อยของขั้นที่ 3 ออกแบบและปฏิบัติการ

1. นำ data ที่ได้ทำการ Clean แล้วมาหาความสัมพันธ์ระหว่างตัวแปรแต่ละตัว

	carat	cut	color	clarity	depth	table	price	x	y	z
carat	1.000000	0.049582	0.074952	0.355949	0.016252	0.032659	0.747830	0.987340	0.986201	0.984408
cut	0.049582	1.000000	-0.006503	0.159045	0.221849	0.399029	-0.111005	-0.006807	0.001672	0.049201
color	0.074952	-0.006503	1.000000	-0.163526	0.051920	-0.033861	-0.162964	0.068171	0.071632	0.081010
clarity	0.355949	0.159045	-0.163526	1.000000	0.091535	0.097027	-0.124267	0.339305	0.326861	0.352613
depth	0.016252	0.221849	0.051920	0.091535	1.000000	-0.324234	-0.057646	-0.082016	-0.087250	0.152262
table	0.032659	0.399029	-0.033861	0.097027	-0.324234	1.000000	-0.008267	0.027399	0.011792	-0.057083
price	0.747830	-0.111005	-0.162964	-0.124267	-0.057646	-0.008267	1.000000	0.759364	0.737978	0.730446
x	0.987340	-0.006807	0.068171	0.339305	-0.082016	0.027399	0.759364	1.000000	0.992551	0.970394
y	0.986201	0.001672	0.071632	0.326861	-0.087250	0.011792	0.737978	0.992551	1.000000	0.969128
z	0.984408	0.049201	0.081010	0.352613	0.152262	-0.057083	0.730446	0.970394	0.969128	1.000000



2. กำหนดข้อมูลที่จะนำไปเทรน โดยใช้ฟังก์ชัน train_test_split แบ่งข้อมูลสำหรับเทรนโมเดล 70% และอีก 30% เป็นข้อมูลชุดทดสอบ กำหนดค่า random_state = 101

```
1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 101)
```

3. นำข้อมูลที่จะใช้เทรนเข้าสู่โมเดล และตรวจสอบผลลัพธ์

```
1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 101)
3
4 mod2 = sm.OLS(y_train, X_train) # Create Model
5 results2 = mod2.fit()
6
7 results2.summary()
```

OLS Regression Results						
Dep. Variable:	price		R-squared:	0.879		
Model:	OLS		Adj. R-squared:	0.879		
Method:	Least Squares		F-statistic:	1.077e+04		
Date:	Wed, 17 Nov 2021		Prob (F-statistic):	0.00		
Time:	20:50:59		Log-Likelihood:	-79262.		
No. Observations:	13351		AIC:	1.585e+05		
Df Residuals:	13341		BIC:	1.586e+05		
Df Model:	9					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	-2215.3961	575.603	-3.849	0.000	-3343.659	-1087.133
carat	3877.2509	103.613	37.420	0.000	3674.154	4080.348
x	1721.7128	65.168	26.420	0.000	1593.975	1849.451
y	-1136.8569	64.945	-17.505	0.000	-1264.158	-1009.556
z	-1190.1302	199.100	-5.978	0.000	-1580.395	-799.865
table	-1.7343	0.531	-3.268	0.001	-2.775	-0.694
depth	47.4263	9.178	5.167	0.000	29.436	65.416
cut	-16.5138	0.998	-16.555	0.000	-18.469	-14.558
color	-51.5837	0.506	-101.985	0.000	-52.575	-50.592
clarity	-81.4655	0.522	-156.182	0.000	-82.488	-80.443
Omnibus:	350.292	Durbin-Watson:	1.974			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	755.620			
Skew:	0.140	Prob(JB):	8.30e-165			
Kurtosis:	4.131	Cond. No.	6.52e+04			

4. นำค่าที่ได้จาก modal มาทำนายราคาจากข้อมูลชุดทดสอบที่เตรียมไว้

```
1 price_predict = []
2 X_test = X_test.reset_index()
3 for i in range(len(X_test)):
4     price_predict.append(-2215.3961
5         +3877.2509 *X_test["carat"][i]
6         +1721.7128 *X_test["x"][i]
7         -1136.8569 *X_test["y"][i]
8         -1190.1302 *X_test["z"][i]
9         -1.7343 *X_test["table"][i]
10        +47.4263 *X_test["depth"][i]
11        -16.5138 *X_test["cut"][i]
12        -51.5837 *X_test["color"][i]
13        -81.4655 *X_test["clarity"][i])
14
15 indexs = [i for i in range(len(X_test))]
16 d = dict(zip(indexs,price_predict) )
17
18 predict_data = pd.concat([y_test.reset_index() , pd.Series(d, name='predict')], axis = 'columns')
19 predict_data
```

	index	price	predict
0	13441	1050	1091.936728
1	5416	732	717.073696
2	7331	794	912.796608
3	9141	862	891.184598
4	8798	847	929.778020
...
5718	3894	684	701.445828
5719	14735	492	600.211732
5720	11421	952	987.045678
5721	3643	679	646.202061
5722	12045	982	997.712783

ขั้นตอนย่อยของขั้นที่ 4 ทดสอบและปรับปรุงแก้ไข

1. ตรวจสอบค่า Mean Absolute Percentage Error

```
summm = 0
for i in predict_data.index:
    d = predict_data["price"][i] - predict_data["predict"][i]
    summm += abs(d) / predict_data["price"][i]

mape=(summm / len(predict_data))*100
print("mape =",mape)

mape = 8.65186738879971
```

2. ตรวจสอบค่า R - squared

OLS Regression Results			
Dep. Variable:	price	R-squared:	0.879
Model:	OLS	Adj. R-squared:	0.879

ขั้นที่ 5 การประเมินผลและแสดงผล

เลือกโมเดลที่จะใช้ และนำสมการที่ได้จากโมเดลมาแสดงผลบนเว็บไซต์



Predict Diamond Price

Carat	0.4	Length	4.8
Width	4.76	Height	2.89
Table	59	Depth	60.5
Color	0	Cut	1
Clarity	5		

Predicted

~ 1091.94 \$

บทที่ 3

การประยุกต์ใช้ทฤษฎี

การประยุกต์ใช้ทฤษฎีเวกเตอร์

Pearson's Similarity

	carat	cut	color	clarity	depth	table	price	x	y	z
carat	1.000000	0.049582	0.074952	0.355949	0.016252	0.032659	0.747830	0.987340	0.986201	0.984408
cut	0.049582	1.000000	-0.006503	0.159045	0.221849	0.399029	-0.111005	-0.006807	0.001672	0.049201
color	0.074952	-0.006503	1.000000	-0.163526	0.051920	-0.033861	-0.162964	0.068171	0.071632	0.081010
clarity	0.355949	0.159045	-0.163526	1.000000	0.091535	0.097027	-0.124267	0.339305	0.326861	0.352613
depth	0.016252	0.221849	0.051920	0.091535	1.000000	-0.324234	-0.057646	-0.082016	-0.087250	0.152262
table	0.032659	0.399029	-0.033861	0.097027	-0.324234	1.000000	-0.008267	0.027399	0.011792	-0.057083
price	0.747830	-0.111005	-0.162964	-0.124267	-0.057646	-0.008267	1.000000	0.759364	0.737978	0.730446
x	0.987340	-0.006807	0.068171	0.339305	-0.082016	0.027399	0.759364	1.000000	0.992551	0.970394
y	0.986201	0.001672	0.071632	0.326861	-0.087250	0.011792	0.737978	0.992551	1.000000	0.969128
z	0.984408	0.049201	0.081010	0.352613	0.152262	-0.057083	0.730446	0.970394	0.969128	1.000000

ในส่วนของโครงงานเรานั้นเราได้นำในส่วนของ Pearson's Similarity มาประยุกต์ใช้ในโครงงานโดยการนำมาใช้หาค่าความสัมพันธ์ระหว่างตัวแปรต่างๆที่อยู่ในข้อมูล เพื่อดูว่าตัวแปรไหนส่งผลต่อกันมากน้อยเท่าไร เพื่อจะพิจารณาหาตัวแปรที่เหมาะสมมาเป็นส่วนหนึ่งในการสร้างModel ซึ่งหากว่าค่าของPearson's Similarity ยิ่งเข้าใกล้ 1 หรือ -1 แปลว่าตัวแปรคู่นั้นส่งผลต่อกันมาก หากค่าPearsonเป็นบวกแสดงว่าแปรผันตามกัน แต่หากเป็นลบแปลว่าแปรผกผันกัน

การประยุกต์ใช้ทฤษฎีเมทริกซ์

Regression Model

ใช้โมเดล Ordinary Least Squares Regression ซึ่งเป็นฟังก์ชันของ library statmodels

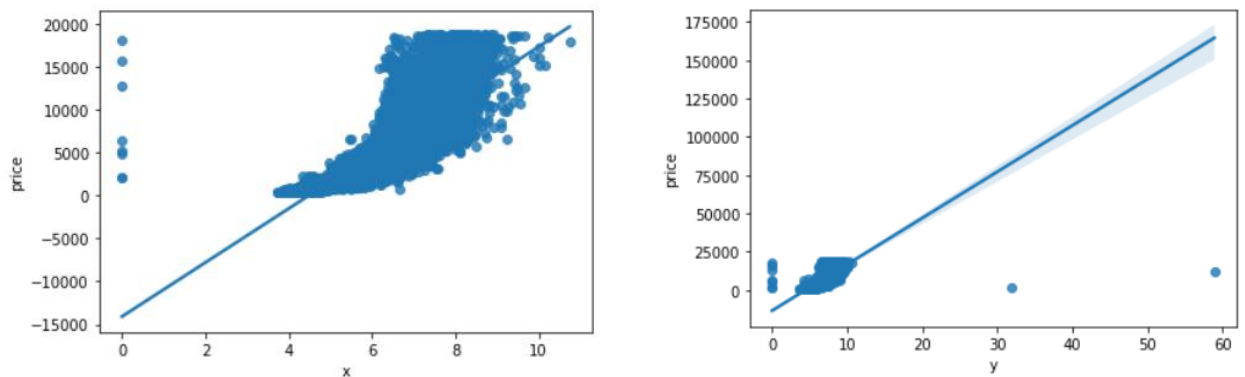
OLS Regression Results						
Dep. Variable:	price		R-squared:	0.879		
Model:	OLS		Adj. R-squared:	0.879		
Method:	Least Squares		F-statistic:	1.077e+04		
Date:	Wed, 17 Nov 2021		Prob (F-statistic):	0.00		
Time:	20:50:59		Log-Likelihood:	-79262.		
No. Observations:	13351		AIC:	1.585e+05		
Df Residuals:	13341		BIC:	1.586e+05		
Df Model:	9					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	-2215.3961	575.603	-3.849	0.000	-3343.659	-1087.133
carat	3877.2509	103.613	37.420	0.000	3674.154	4080.348
x	1721.7128	65.168	26.420	0.000	1593.975	1849.451
y	-1136.8569	64.945	-17.505	0.000	-1264.158	-1009.556
z	-1190.1302	199.100	-5.978	0.000	-1580.395	-799.865
table	-1.7343	0.531	-3.268	0.001	-2.775	-0.694
depth	47.4263	9.178	5.167	0.000	29.436	65.416
cut	-16.5138	0.998	-16.555	0.000	-18.469	-14.558
color	-51.5837	0.506	-101.985	0.000	-52.575	-50.592
clarity	-81.4655	0.522	-156.182	0.000	-82.488	-80.443
Omnibus:	350.292	Durbin-Watson:	1.974			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	755.620			
Skew:	0.140	Prob(JB):	8.30e-165			
Kurtosis:	4.131	Cond. No.	6.52e+04			

ภายในโครงงานเราได้ใช้ regression model ในการทำนายราคาจากสมการ regression โดยทางเราได้คำนวณหา model ordinary least squares regression ซึ่งเป็น function ภายในของ library statmodels โดยจะนำข้อมูลที่เราไปคำนวณหาสมการ regression และนำสมการนี้ไปใช้ในการ predict หาค่าตัวแปรที่ต้องการ

บทที่ 4

ผลการทดลอง

ผลการทดลองชุดที่ 1



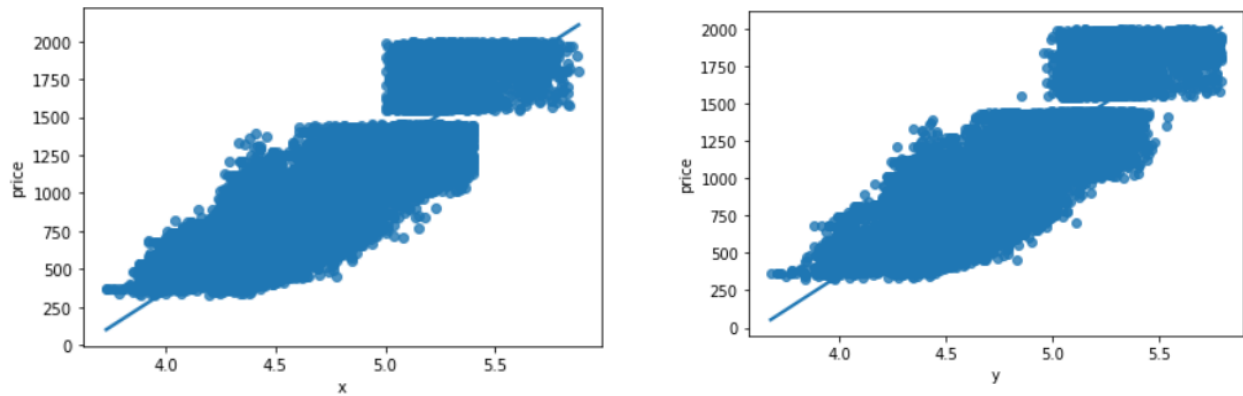
ในการทดลองครั้งที่ 1 กลุ่มของพวกเราได้ทำการเลือกใช้ข้อมูลร้อยละ 70 โดยการสุ่มในการ Train และ ร้อยละ 30 ที่เหลือในการ Test โดยยังไม่ได้ทำการตัดข้อมูลที่เป็นข้อมูลสุดโต่งและช่วงของข้อมูลที่ไม่ต่อเนื่อง ส่งผลให้ราคาที่ทำนายออกมาได้มีแนวโน้มสูงขึ้นจนเกินจริง และหลังจากนำมาหาค่า MAPE มีค่าประมาณร้อยละ 44.12

```
summm = 0
for i in predict_data.index:
    d = predict_data["price"][i] - predict_data["predict"][i]
    summm += abs(d)/predict_data["price"][i]

mape=(summm / len(predict_data)) * 100
print("mape =",mape)
```

```
mape = 44.12156000705811
```


ผลการทดลองชุดที่ 2



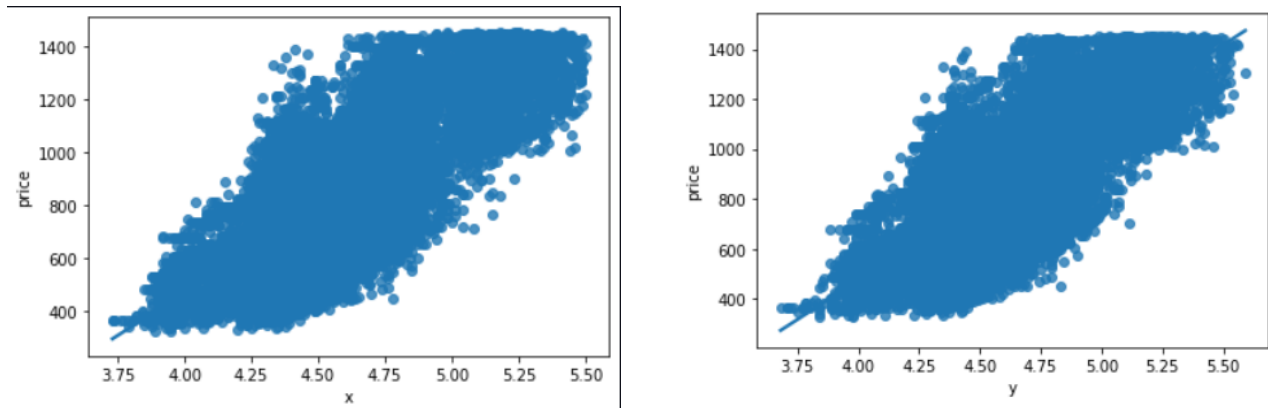
ในการทดลองครั้งที่ 2 กลุ่มของพวกเราได้ทำการเลือกใช้ข้อมูลร้อยละ 70 โดยการสุ่มในการ Train และ ร้อยละ 30 ที่เหลือในการ Test โดยทำการตัดข้อมูลส่วนที่เป็นข้อมูลสุดโต่งไป แต่ยังไม่ได้ทำการตัดข้อมูลช่วงของ ข้อมูลที่ไม่ต่อเนื่อง ส่งผลให้ราคาที่ทำนายออกมาได้นั้นมีความแม่นยำมากขึ้น และเมื่อนำมาหาค่า MAPE มี ค่าประมาณร้อยละ 10.70

```
summm=0
for i in predict_data.index:
    d = predict_data["price"][i] - predict_data["predict"][i]
    summm+=abs(d)/predict_data["price"][i]

mape=(summm/len(predict_data))*100
print("mape =",mape)

mape = 10.700792624162187
```

ผลการทดลองชุดที่ 3



ในการทดลองครั้งที่ 3 กลุ่มของพวกเราได้ทำการเลือกใช้ข้อมูลร้อยละ 70 โดยการสุ่มในการ Train และ ร้อยละ 30 ที่เหลือในการ Test โดยทำการตัดข้อมูลส่วนที่เป็นข้อมูลสุดโต่งและตัดตรงส่วนของข้อมูลที่แยกเป็นชั้นชัดเจน ส่งผลให้ราคาที่ทำนายออกมาได้มีแนวโน้มที่เหมาะสมไม่มากเกินไปและไม่น้อยเกินไป ทำให้เหมาะแก่การนำมาใช้งานมากกว่าทั้งสองแบบ และเมื่อนำมาหาค่า MAPE จะได้ประมาณร้อยละ 8.65 ซึ่งอยู่ในระดับที่สามารถยอมรับได้

```
summm = 0
for i in predict_data.index:
    d = predict_data["price"][i] - predict_data["predict"][i]
    summm += abs(d) / predict_data["price"][i]

mape=(summm / len(predict_data))*100
print("mape =",mape)

mape = 8.65186738879971
```

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

สรุปผลการทดลอง

จากผลการทดลองชุดที่ 3

1. โมเดลสามารถทำนายราคาของเพชรจากค่าขององค์ประกอบต่างๆที่รับเข้าไปได้
2. ค่าความคลาดเคลื่อนหรือค่า MAPE ได้ค่าโดยเฉลี่ยร้อยละ 8.65 ซึ่งอยู่ในเกณฑ์ที่สามารถรับได้
3. เว็บไซต์สามารถแสดงผลข้อมูลจากการเปิดผ่านเว็บเบราว์เซอร์ต่างชนิดกันบนคอมพิวเตอร์และ ผ่านโทรศัพท์ได้อย่างถูกต้องและสมบูรณ์

ข้อเสนอแนะ

1. หากมีจำนวนของชุดข้อมูลที่ไม่ใช่ข้อมูลสุดโตรงและความต่อเนื่องมากขึ้น อาจทำให้สามารถทำนายค่าออกมาได้ใกล้เคียงมากยิ่งขึ้น
2. หากมีระยะเวลาในการดำเนินการมากขึ้น อาจทำให้ผลลัพธ์ที่ได้มีประสิทธิภาพที่มากขึ้น

รายการอ้างอิง

เริงชัย ต้นสุชาติ. การวิเคราะห์การถดถอยเชิงเดี่ยว (Simple Regression Analysis). สืบค้น 8 ตุลาคม.

จาก : http://lms.mju.ac.th/courses/159/locker/Econometrics2/content_con/bfiles/2.htm

ดร. อานนท์ ศักดิ์วรวิชญ์. ว่าด้วยการวิเคราะห์ถดถอย 33 ชนิด (On 33 kinds of Regression Analyses).

สืบค้น 9 ตุลาคม.

จาก : <https://businessanalyticsnida.wordpress.com/2016/09/18/32-kinds-of-regression/>

Microsoft. ฟังก์ชัน COVARIANCE.P. สืบค้น 11 ตุลาคม.

จาก : <https://support.microsoft.com/th-th/office/%E0%B8%9F%E0%B8%B1%E0%B8%87%E0%B8%81%E0%B9%8C%E0%B8%8A%E0%B8%B1%E0%B8%99-covariance-p-6f0e1e6d-956d-4e4b-9943-cfef0bf9edfc>

CFI. Multiple Linear Regression. สืบค้น 12 ตุลาคม.

จาก : <https://corporatefinanceinstitute.com/resources/knowledge/other/multiple-linear-regression/>

ภาคผนวก

ภาคผนวก ก

ข้อมูลโครงงาน

[1] ข้อมูลที่ใช้ :

<https://www.kaggle.com/shivam2503/diamonds>

[2] Source code หรือ File ที่ใช้ในการคำนวณ :

<https://github.com/zantaclaus/Predict-Diamond-Price>

[3] Source code Website แสดงผล :

<https://github.com/zantaclaus/DiamondPrice-Frontend>

ภาคผนวก ข

วิดีโอและสไลด์นำเสนอโครงการ

[1] วิดีโอนำเสนอ :

<https://drive.google.com/drive/folders/1wHJevZtgLcTviOuzbvE85ZmMPowBBYW9?usp=sharing>

[2] สไลด์นำเสนอ :

<https://drive.google.com/drive/folders/1FMjqVcRcAL4OYdjKIBT2UFYE5SXSyz4Z?usp=sharing>

Website ทำนายราคาและบทความรายละเอียดการทำงาน

[1] Predict Diamond Price :

<https://predictdiamondprice.netlify.app/>

[2] การทำนายราคาเพชรโดยการใช้ Linear Regression (Medium)

https://medium.com/@justo_play/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%97%E0%B8%B3%E0%B8%99%E0%B8%B2%E0%B8%A2%E0%B8%A3%E0%B8%B2%E0%B8%84%E0%B8%B2%E0%B9%80%E0%B8%9E%E0%B8%8A%E0%B8%A3%E0%B9%82%E0%B8%94%E0%B8%A2%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89-linear-regression-daab31bb6b9b

ประวัติผู้จัดทำ

Group : KBub



ชื่อ : นายชินาธิป มีสวนนิล

รหัสนักศึกษา : 63010235

คณะ : วิศวกรรมศาสตร์

สาขา : วิศวกรรมคอมพิวเตอร์



ชื่อ : นายณัฐพงษ์ นาคสามัคคี

รหัสนักศึกษา : 63010326

คณะ : วิศวกรรมศาสตร์

สาขา : วิศวกรรมคอมพิวเตอร์

Group : ฟรีๆ



ชื่อ : นายจิรภัทร แก้วส่งแสง

รหัสนักศึกษา : 63010139

คณะ : วิศวกรรมศาสตร์

สาขา : วิศวกรรมคอมพิวเตอร์



ชื่อ : นายดิษฐพงษ์ จรัสชัยโรจน์

รหัสนักศึกษา : 63010354

คณะ : วิศวกรรมศาสตร์

สาขา : วิศวกรรมคอมพิวเตอร์