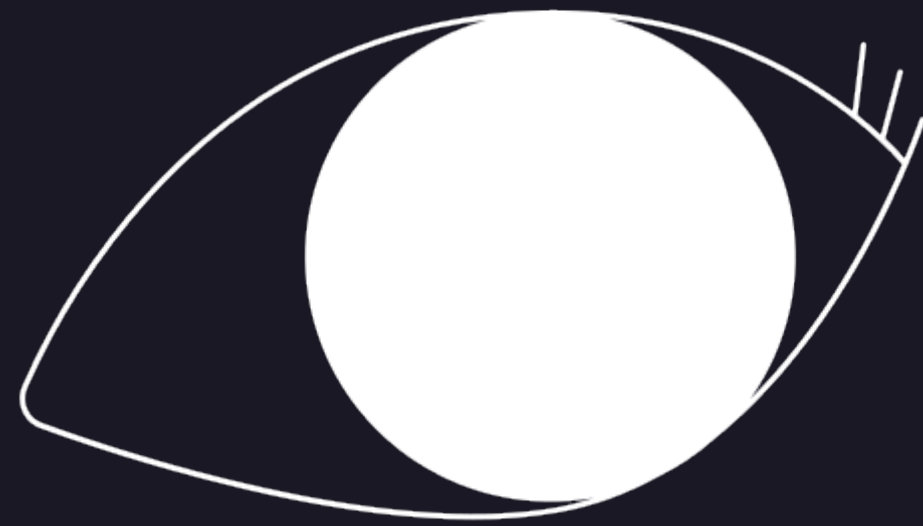


โครงการ การตั้งราคาสินค้า



บทที่ 1 : หลักการ และ วัตถุประสงค์

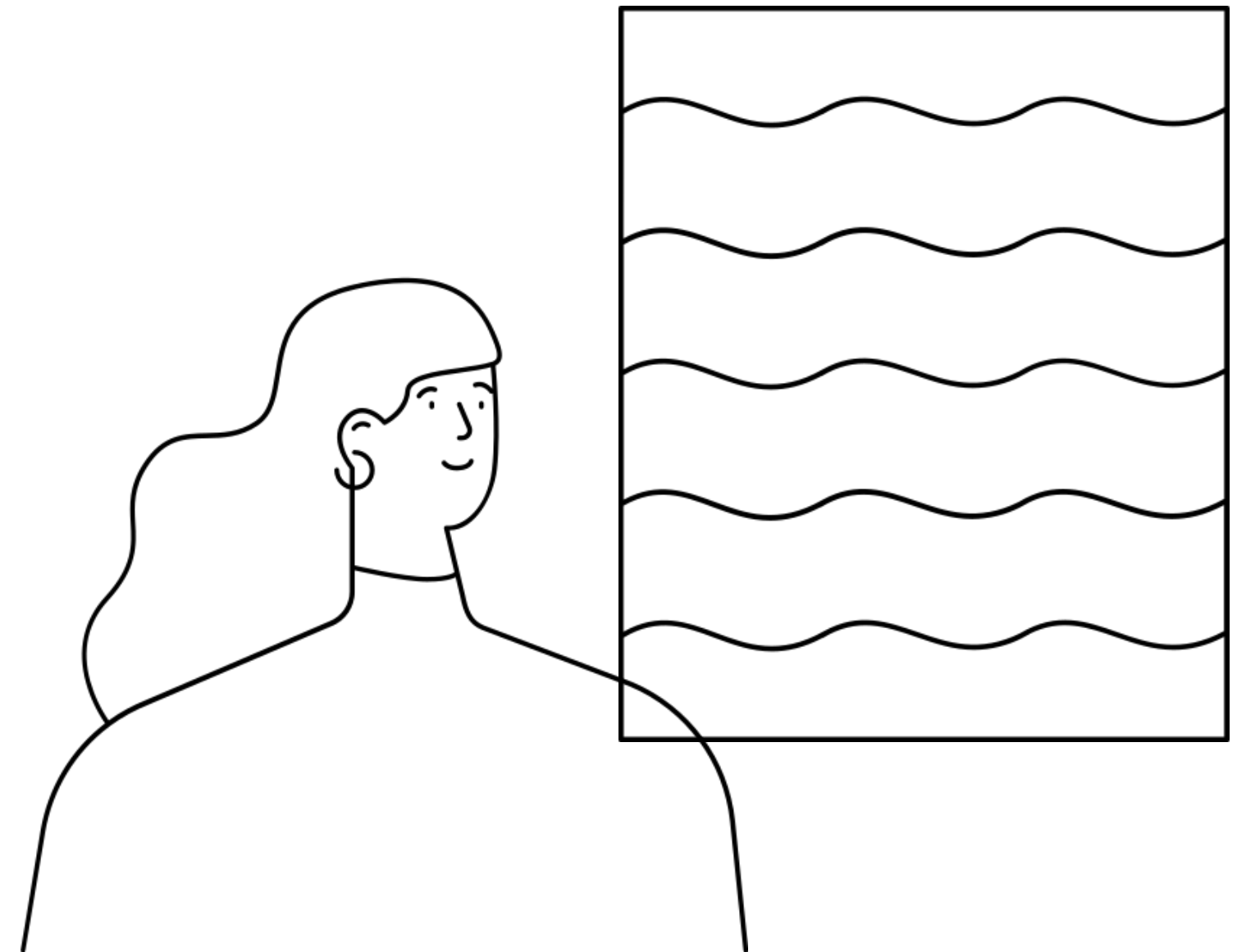


วัตถุประสงค์

- สร้างตัวแบบทำนายขึ้นเพื่อเลียนแบบพฤติกรรมในการกำหนดราคาสินค้า

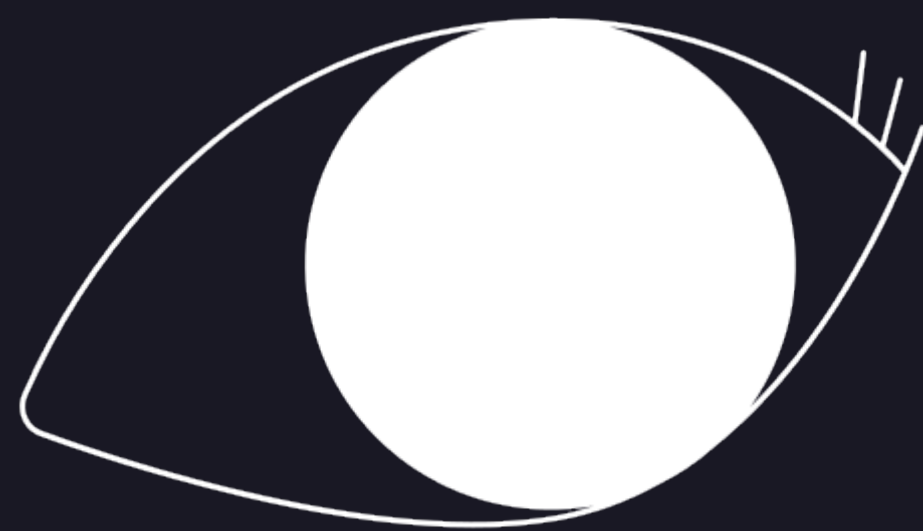
ข้อมูล

- พนักงานขายรับหน้าที่เป็นผู้กำหนดราคาขายประจำสัปดาห์ให้กับสินค้า
- บริษัทให้ข้อมูลการตั้งราคาสินค้ามาทั้งหมด 50 สัปดาห์



	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice
0	A-A-00001	2019-01-04	27.919192	33.0	28.545455	20.535354	A	41.0
1	A-B-00001	2019-01-04	57.333333	63.0	56.121212	61.838384	A	92.0
2	A-B-00002	2019-01-04	50.777778	54.0	45.131313	50.000000	A	76.0
3	A-C-00001	2019-01-04	45.747475	56.0	40.525253	38.080808	B	75.0
4	A-C-00019	2019-01-04	45.747475	56.0	40.525253	38.080808	B	75.0
5	A-C-00007	2019-01-04	59.373737	62.0	59.393939	58.272727	B	90.0
6	A-C-00002	2019-01-04	39.777778	45.0	39.111111	39.313131	B	68.0
7	A-C-00010	2019-01-04	61.222222	70.0	0.000000	0.000000	B	99.0
8	A-C-00003	2019-01-04	39.232323	43.0	39.707071	39.707071	B	64.0
9	A-C-00021	2019-01-04	48.818182	51.0	48.353535	49.242424	B	75.0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25116 entries, 0 to 25115
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   SKU                    25116 non-null  object
1   date                   25116 non-null  object
2   PrevAVGCost            24639 non-null  float64
3   PrevAssignedCost      24639 non-null  float64
4   AVGCost                24639 non-null  float64
5   LatestDateCost        24639 non-null  float64
6   Type                   24802 non-null  object
7   GenPrice               24639 non-null  float64
dtypes: float64(5), object(3)
memory usage: 1.5+ MB
```



บทที่ 2 : การทำความสะอาดข้อมูล

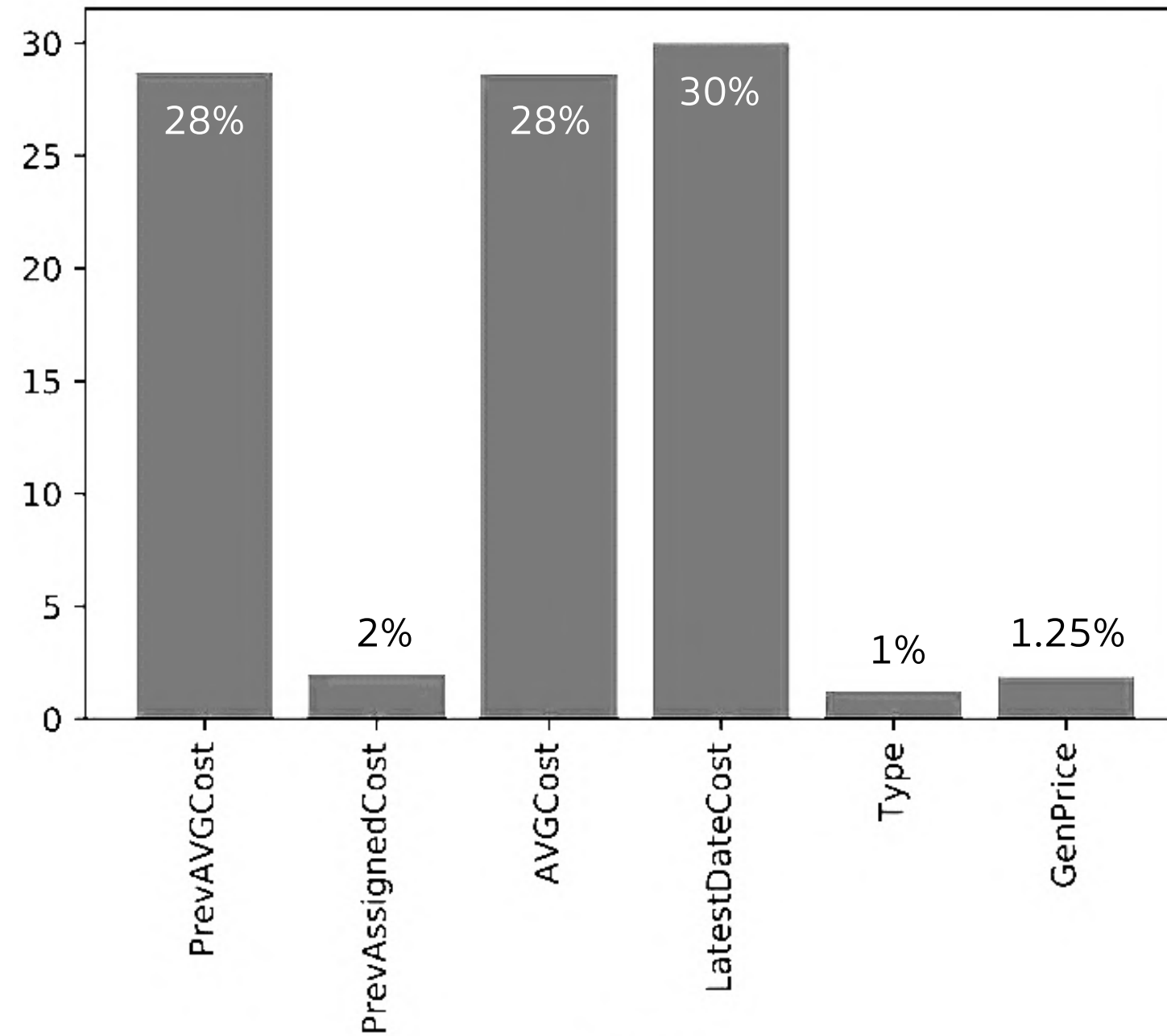
1. Taking Care of Missing Values

	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice			
11	A-C-00006	2019-01-04	NaN	NaN	NaN	NaN	A	NaN			
77	A-A-00013	2019-01-04	NaN	NaN	NaN	NaN	A	NaN			
303	E-A-00195	2019-01-04	NaN	NaN	NaN	NaN	G	NaN			
317	E-A-00399	2019-01-04	NaN	NaN	NaN	NaN	G	NaN			
455	E-A-00023	2019-01-04	NaN	NaN	NaN	NaN	G	NaN			
...								
	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice			
25111	A-E-00170	2020-01-13	NaN	4071 E-E-00084	2019-02-25	0.0	0.0	0.0	0.0	G	163.0
25112	A-E-00171	2020-01-13	NaN	4817 A-E-00092	2019-03-12	0.0	0.0	0.0	0.0	A	36.0
25113	A-D-00149	2020-01-13	NaN	5327 A-E-00092	2019-03-20	0.0	0.0	0.0	0.0	A	36.0
25114	A-B-00040	2020-01-13	NaN	6875 A-E-00092	2019-04-10	0.0	0.0	0.0	0.0	A	36.0
25115	A-I-00090	2020-01-13	NaN	7393 A-E-00092	2019-04-22	0.0	0.0	0.0	0.0	A	36.0
				7911 A-E-00092	2019-04-26	0.0	0.0	0.0	0.0	A	36.0
				11455 A-E-00092	2019-06-19	0.0	0.0	0.0	0.0	A	35.0
				12934 A-E-00092	2019-07-11	0.0	0.0	0.0	0.0	A	35.0
				13427 A-E-00092	2019-07-22	0.0	0.0	0.0	0.0	A	35.0
				13920 A-E-00092	2019-07-25	0.0	0.0	0.0	0.0	A	35.0
				14413 A-E-00092	2019-08-05	0.0	0.0	0.0	0.0	A	33.0

477 rows x 8 columns

1. Taking Care of Missing Values

ร้อยละข้อมูลที่ขาดหายไป



1. Taking Care of Missing Values

1.1. Remove missing GenPrice and GenPrice = 0

	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice
11	A-C-00006	2019-01-04	nan	nan	nan	nan	A	nan
77	A-A-00013	2019-01-04	nan	nan	nan	nan	A	nan
303	E-A-00195	2019-01-04	nan	nan	nan	nan	G	nan
317	E-A-00399	2019-01-04	nan	nan	nan	nan	G	nan
455	E-A-00023	2019-01-04	nan	nan	nan	nan	G	nan
...
25111	A-E-00170	2020-01-13	nan	nan	nan	nan	NaN	nan
25112	A-E-00171	2020-01-13	nan	nan	nan	nan	NaN	nan
25113	A-D-00149	2020-01-13	nan	nan	nan	nan	NaN	nan
25114	A-B-00040	2020-01-13	nan	nan	nan	nan	NaN	nan
25115	A-I-00090	2020-01-13	nan	nan	nan	nan	NaN	nan

เหลือ 24,639 รายการ

1. Taking Care of Missing Values

1.2. Remove PrevAssignedCost = 0 and AVGCost = 0

	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice
11	A-C-00006	2019-01-04	nan	nan	nan	nan	A	nan
77	A-A-00013	2019-01-04	nan	nan	nan	nan	A	nan
303	E-A-00195	2019-01-04	nan	nan	nan	nan	G	nan
317	E-A-00399	2019-01-04	nan	nan	nan	nan	G	nan
455	E-A-00023	2019-01-04	nan	nan	nan	nan	G	nan
...
25111	A-E-00170	2020-01-13	nan	nan	nan	nan	NaN	nan
25112	A-E-00171	2020-01-13	nan	nan	nan	nan	NaN	nan
25113	A-D-00149	2020-01-13	nan	nan	nan	nan	NaN	nan
25114	A-B-00040	2020-01-13	nan	nan	nan	nan	NaN	nan
25115	A-I-00090	2020-01-13	nan	nan	nan	nan	NaN	nan

เหลือ 24,628 รายการ

2. Taking Care of Excess Values

	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice
25010	A-A-00032	2020-01-13	0.0	10100002.0	0.0	0.0	A	51.0

	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice
24511	A-A-00032	2020-01-06	34.57732	35.0	0.0	0.0	A	12625006.0

ແລ້ວ 24,626 ກາຍກາຣ

3. Calculate Margin

$$\textit{margin} = \frac{\textit{GenPrice} - \textit{PrevAssignedCost}}{\textit{PrevAssignedCost}} \times 100$$

3. Calculate Margin

	SKU	date	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice	Margin
0	A-A-00001	2019-01-04	27.92	33.00	28.55	20.54	A	41.00	24.24
1	A-B-00001	2019-01-04	57.33	63.00	56.12	61.84	A	92.00	46.03
2	A-B-00002	2019-01-04	50.78	54.00	45.13	50.00	A	76.00	40.74
3	A-C-00001	2019-01-04	45.75	56.00	40.53	38.08	B	75.00	33.93
4	A-C-00019	2019-01-04	45.75	56.00	40.53	38.08	B	75.00	33.93
5	A-C-00007	2019-01-04	59.37	62.00	59.39	58.27	B	90.00	45.16
6	A-C-00002	2019-01-04	39.78	45.00	39.11	39.31	B	68.00	51.11
7	A-C-00010	2019-01-04	61.22	70.00	0.00	0.00	B	99.00	41.43
8	A-C-00003	2019-01-04	39.23	43.00	39.71	39.71	B	64.00	48.84
9	A-C-00021	2019-01-04	48.82	51.00	48.35	49.24	B	75.00	47.06

3. Calculate Margin

```
count      24626.000000
mean        50.258287
std         29.148326
min        -59.405941
25%         35.483871
50%         45.714286
75%         60.000000
max         1900.000000
Name: Margin, dtype: float64
```

4. Create Dummy Variables for Types

Type		A	B	C	D	E	F	G
A	0	1	0	0	0	0	0	0
A	1	1	0	0	0	0	0	0
A	2	1	0	0	0	0	0	0
B	3	0	1	0	0	0	0	0
B	4	0	1	0	0	0	0	0

4. Create Dummy Variables for Types

	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	GenPrice	A	B	C	D	E	F	G
0	27.92	33.00	28.55	20.54	A	41.00	1	0	0	0	0	0	0
1	57.33	63.00	56.12	61.84	A	92.00	1	0	0	0	0	0	0
2	50.78	54.00	45.13	50.00	A	76.00	1	0	0	0	0	0	0
3	45.75	56.00	40.53	38.08	B	75.00	0	1	0	0	0	0	0
4	45.75	56.00	40.53	38.08	B	75.00	0	1	0	0	0	0	0
...
25103	0.00	60.00	0.00	0.00	G	87.00	0	0	0	0	0	0	1
25104	80.81	96.00	0.00	0.00	G	126.00	0	0	0	0	0	0	1
25105	0.00	23.00	25.13	25.13	G	40.00	0	0	0	0	0	0	1
25106	151.52	201.00	181.82	181.82	G	271.00	0	0	0	0	0	0	1
25107	0.00	70.00	0.00	0.00	G	107.00	0	0	0	0	0	0	1

5. Extract MainCat and SubCat from SKU



Carrot



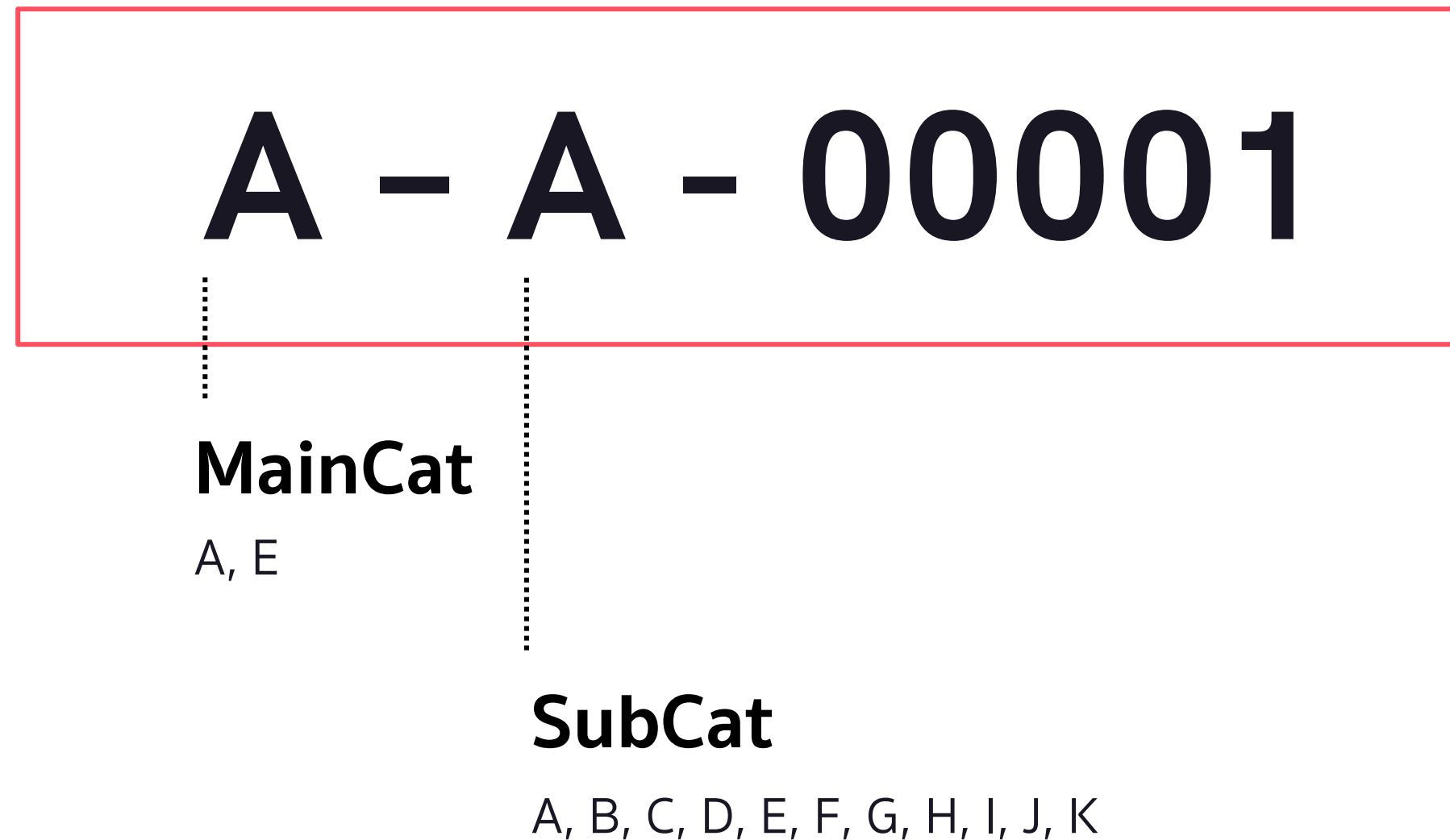
Minced Carrot



Baby Carrot

5. Extract MainCat and SubCat from SKU

ชื่อกลุ่มสินค้าหลัก – ชื่อกลุ่มสินค้าย่อย – ลำดับของสินค้าในกลุ่มย่อย



5. Extract MainCat and SubCat from SKU

MainCat

SubCat

	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	GenPrice	A	B	C	D	E	F	G	Cat A	Cat E	SubCat A	SubCat B	SubCat C	SubCat D	SubCat E	SubCat F	SubCat G	SubCat H	SubCat I	SubCat J	SubCat K	
0	27.919192	33.0	28.545455	20.535354	41.0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	
1	57.333333	63.0	56.121212	61.838384	92.0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	
2	50.777778	54.0	45.131313	50.000000	76.0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	
3	45.747475	56.0	40.525253	38.080808	75.0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	
4	45.747475	56.0	40.525253	38.080808	75.0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	
...
25103	0.000000	60.0	0.000000	0.000000	87.0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	
25104	80.808081	96.0	0.000000	0.000000	126.0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	
25105	0.000000	23.0	25.131313	25.131313	40.0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	
25106	151.515152	201.0	181.818182	181.818182	271.0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	
25107	0.000000	70.0	0.000000	0.000000	107.0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	

24626 rows x 27 columns

6. Extract Year from date

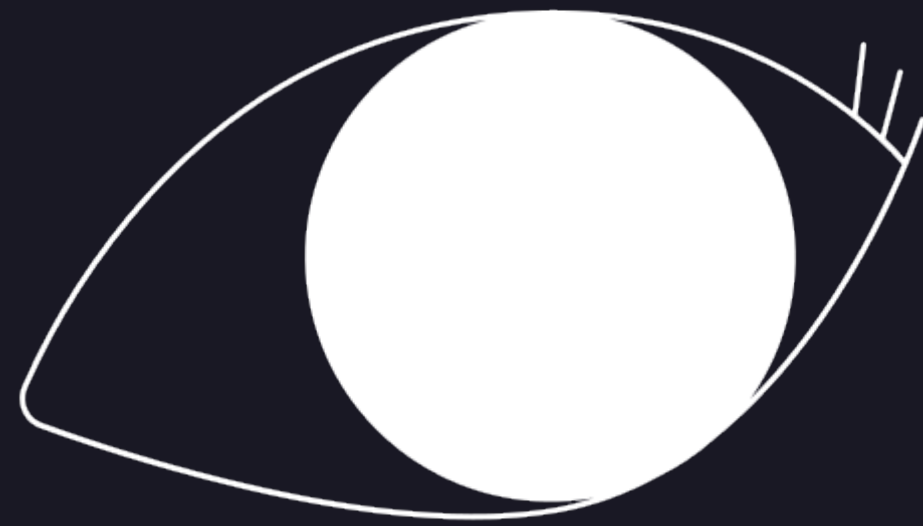
Year

	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	GenPrice	A	B	C	D	E	F	G	Cat A	Cat E	SubCat A	SubCat B	SubCat C	SubCat D	SubCat E	SubCat F	SubCat G	SubCat H	SubCat I	SubCat J	SubCat K	Year19	Year20
0	27.919192	33.0	28.545455	20.535354	41.0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
1	57.333333	63.0	56.121212	61.838384	92.0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0
2	50.777778	54.0	45.131313	50.000000	76.0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0
3	45.747475	56.0	40.525253	38.080808	75.0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0
4	45.747475	56.0	40.525253	38.080808	75.0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0
...
25103	0.000000	60.0	0.000000	0.000000	87.0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
25104	80.808081	96.0	0.000000	0.000000	126.0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
25105	0.000000	23.0	25.131313	25.131313	40.0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
25106	151.515152	201.0	181.818182	181.818182	271.0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
25107	0.000000	70.0	0.000000	0.000000	107.0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1

24626 rows x 27 columns

7. Exporting the cleaned dataset

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 24626 entries, 0 to 25107
Data columns (total 27 columns):
#   Column                Non-Null Count  Dtype
---  ---                -
0   PrevAVGCost           24626 non-null  float64
1   PrevAssignedCost     24626 non-null  float64
2   AVGCost               24626 non-null  float64
3   LatestDateCost       24626 non-null  float64
4   GenPrice              24626 non-null  float64
5   A                     24626 non-null  uint8
6   B                     24626 non-null  uint8
7   C                     24626 non-null  uint8
8   D                     24626 non-null  uint8
9   E                     24626 non-null  uint8
10  F                     24626 non-null  uint8
11  G                     24626 non-null  uint8
12  Cat A                 24626 non-null  uint8
13  Cat E                 24626 non-null  uint8
14  SubCat A             24626 non-null  uint8
15  SubCat B             24626 non-null  uint8
16  SubCat C             24626 non-null  uint8
17  SubCat D             24626 non-null  uint8
18  SubCat E             24626 non-null  uint8
19  SubCat F             24626 non-null  uint8
20  SubCat G             24626 non-null  uint8
21  SubCat H             24626 non-null  uint8
22  SubCat I             24626 non-null  uint8
23  SubCat J             24626 non-null  uint8
24  SubCat K             24626 non-null  uint8
25  Year19               24626 non-null  category
26  Year20               24626 non-null  category
dtypes: category(2), float64(5), uint8(20)
memory usage: 1.6 MB
```



บทที่ 3 : การทดลอง

ตัวแบบทำนายที่ใช้

01

Linear
Regression

02

Support Vector
Regression

03

Feed-forward
Neural Network

วัดผลโดยใช้

01

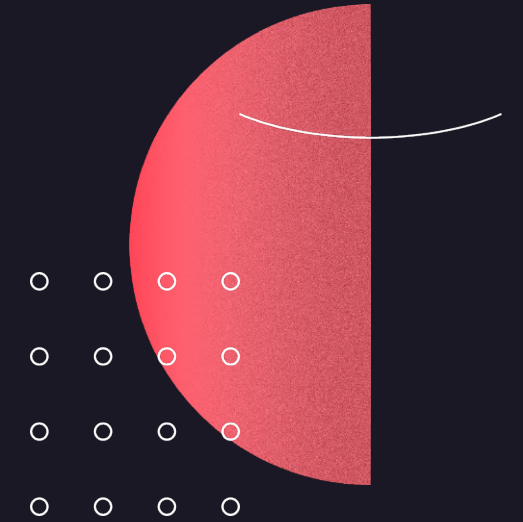
Mean Squared Error
(MSE)

02

Mean Absolute
Percentage Error
(MAPE)

01

Linear Regression



สร้างตัวแบบทำนายเชิงเส้น

- ใช้คลาส `linear_model` ของ `scikit-learn` ด้วยฟังก์ชัน `LinearRegression()`

Grid Search

- ทำ Grid Search เพื่อหาค่าที่ดีที่สุดของพารามิเตอร์
- ประสิทธิภาพการทำนายดีที่สุดเมื่อกำหนดพารามิเตอร์ `fit_intercept = True` และ `normalize = False`

แบ่งข้อมูลเพื่อฝึกสอนตัวแบบทำนาย

- แบ่งข้อมูลออกเป็น 2 ส่วน คือ Training Set 80% และ Test Set 20%
- ทดลองโดยใช้ข้อมูลทดสอบที่มีฟีเจอร์ต่างกัน 3 ชุด

ชุดข้อมูลทดสอบ

01

มีฟิลด์ครบ
ทั้ง 26 คอลัมน์

MustHave (4)

PrevAVGCost
PrevAssignedCost
AVGCost
LatestDateCost

Type (7)

A
B
C
D
E
F
G

MainCat
& SubCat (13)

Cat A
Cat E
SubCat A
SubCat B
SubCat C
SubCat D
SubCat E
SubCat F
SubCat G
SubCat H
SubCat I
SubCat J
SubCat K

Year (2)

Year19
Year20

02

ตัดฟิลด์
Year ออกไป

PrevAVGCost
PrevAssignedCost
AVGCost
LatestDateCost

A
B
C
D
E
F
G

Cat A
Cat E
SubCat A
SubCat B
SubCat C
SubCat D
SubCat E
SubCat F
SubCat G
SubCat H
SubCat I
SubCat J
SubCat K

03

ตัดฟิลด์ Category
และ Subcategory

PrevAVGCost
PrevAssignedCost
AVGCost
LatestDateCost

A
B
C
D
E
F
G

ชุดข้อมูลทดสอบ
ชุดที่ 1

ชุดข้อมูลทดสอบ
ชุดที่ 2

ชุดข้อมูลทดสอบ
ชุดที่ 3

MSE

340.62

340.66

341.55

MAPE

10.42

10.41

10.29

สรุปได้ว่า

ประสิทธิภาพที่ได้ แตกต่างกันไปไม่เกิน 1.0

- ข้อมูลปี (Year)
- กลุ่มหลัก (Category)
- กลุ่มย่อย (Subcategory)

ไม่มีผลต่อประสิทธิภาพของตัวแบบทำนาย

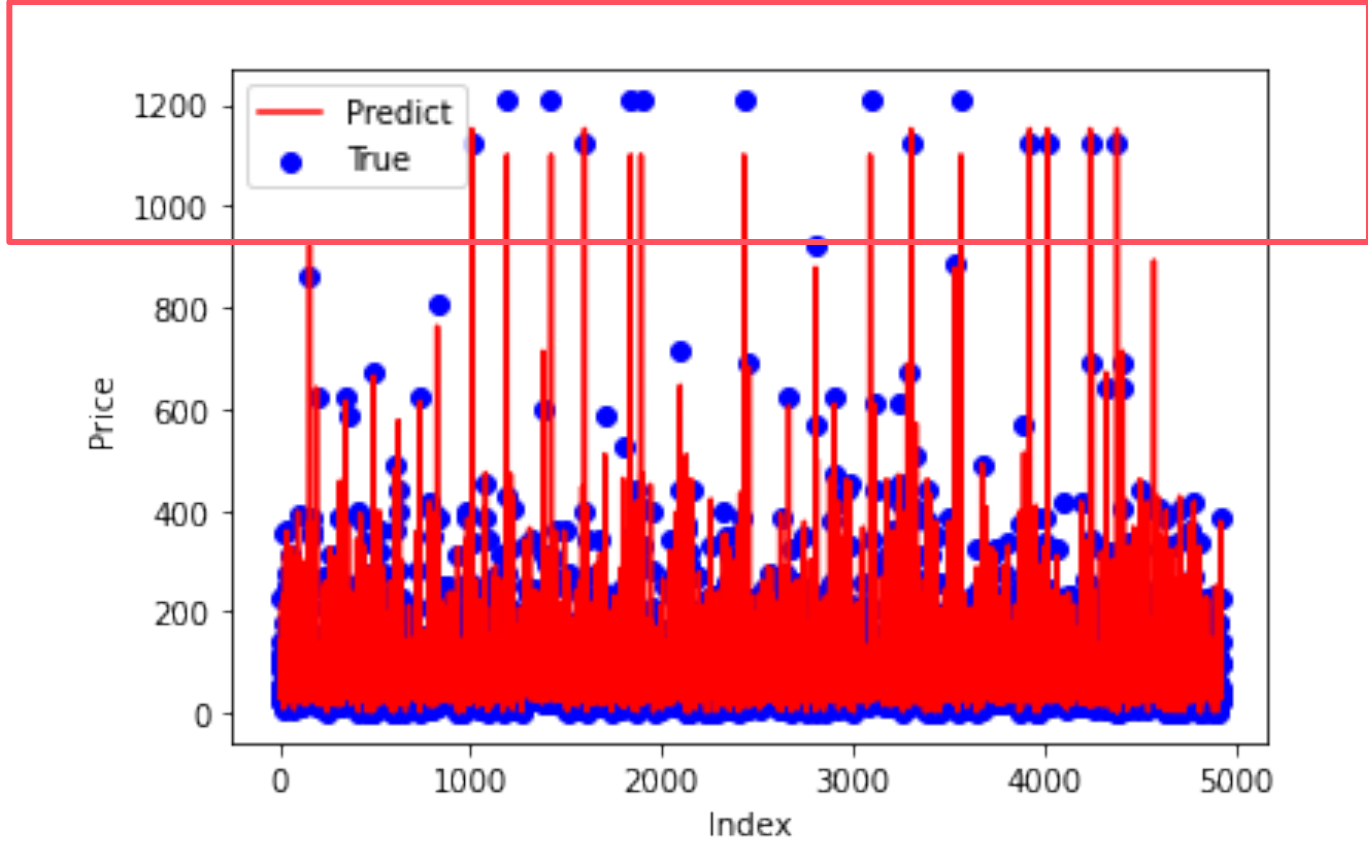


	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	A	B	C	D	E	F	G
0	27.919192	33.0	28.545455	20.535354	1	0	0	0	0	0	0
1	57.333333	63.0	56.121212	61.838384	1	0	0	0	0	0	0
2	50.777778	54.0	45.131313	50.000000	1	0	0	0	0	0	0
3	45.747475	56.0	40.525253	38.080808	0	1	0	0	0	0	0
4	45.747475	56.0	40.525253	38.080808	0	1	0	0	0	0	0
...
24621	0.000000	60.0	0.000000	0.000000	0	0	0	0	0	0	1
24622	80.808081	96.0	0.000000	0.000000	0	0	0	0	0	0	1
24623	0.000000	23.0	25.131313	25.131313	0	0	0	0	0	0	1
24624	151.515152	201.0	181.818182	181.818182	0	0	0	0	0	0	1
24625	0.000000	70.0	0.000000	0.000000	0	0	0	0	0	0	1

Linear Regression

MSE	341.55
MAPE	10.29

GenPrice	PredictedGenPrice
27.0	32.36
105.0	93.56
88.0	85.32
142.0	141.93
99.0	81.55
33.0	31.48
54.0	49.72
117.0	113.99
30.0	33.72
52.0	39.83



Backward Elimination

- ทำ Backward Elimination เพื่อกำจัดตัวแปรที่ไม่มีผลต่อสมการเส้นตรง
- $P > |t| \geq 0.500$

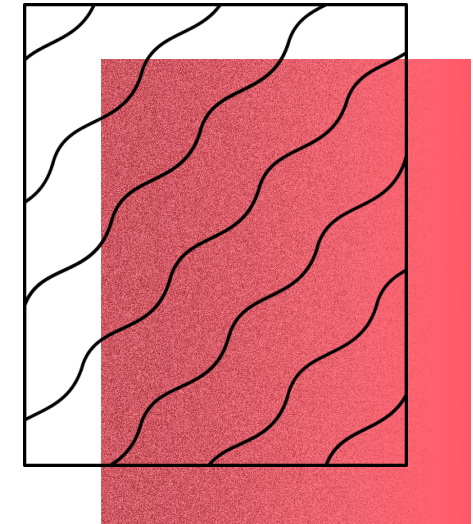
OLS Regression Results

```
=====
Dep. Variable:          GenPrice      R-squared:                0.973
Model:                  OLS           Adj. R-squared:           0.973
Method:                 Least Squares  F-statistic:              8.893e+04
Date:                   Sun, 19 Apr 2020  Prob (F-statistic):       0.00
Time:                   10:48:25      Log-Likelihood:           -1.0556e+05
No. Observations:      24626         AIC:                      2.112e+05
Df Residuals:          24615         BIC:                      2.112e+05
Df Model:               10
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
PrevAVGCost	-0.1911	0.004	-52.465	0.000	-0.198	-0.184
PrevAssignedCost	1.2725	0.002	800.950	0.000	1.269	1.276
AVGCost	0.1428	0.005	27.885	0.000	0.133	0.153
LatestDateCost	0.0304	0.005	6.009	0.000	0.020	0.040
A	7.4652	0.183	40.772	0.000	7.106	7.824
B	7.5636	0.493	15.347	0.000	6.598	8.530
C	15.1689	0.329	46.102	0.000	14.524	15.814
D	2.6529	1.113	2.383	0.017	0.471	4.835
E	0.8021	0.916	0.875	<u>0.381</u>	-0.994	2.598
F	10.3076	1.456	7.080	0.000	7.454	13.161
G	8.5874	0.241	35.679	0.000	8.116	9.059

```
=====
Omnibus:                19463.234  Durbin-Watson:           1.776
Prob(Omnibus):          0.000    Jarque-Bera (JB):        26410327.656
Skew:                   -2.428    Prob(JB):                 0.00
Kurtosis:               163.360    Cond. No.                 1.81e+03
=====
```

Support Vector Regression



สร้างซัพพอร์ตเวกเตอร์ แมชชีนแบบถดถอย

- ฟังก์ชันเคอร์เนล Radial Basis Function (RBF) และ สมการ Gaussian
- ฟังก์ชัน SVR() ของคลาส svm จากไลบรารี scikit-learn

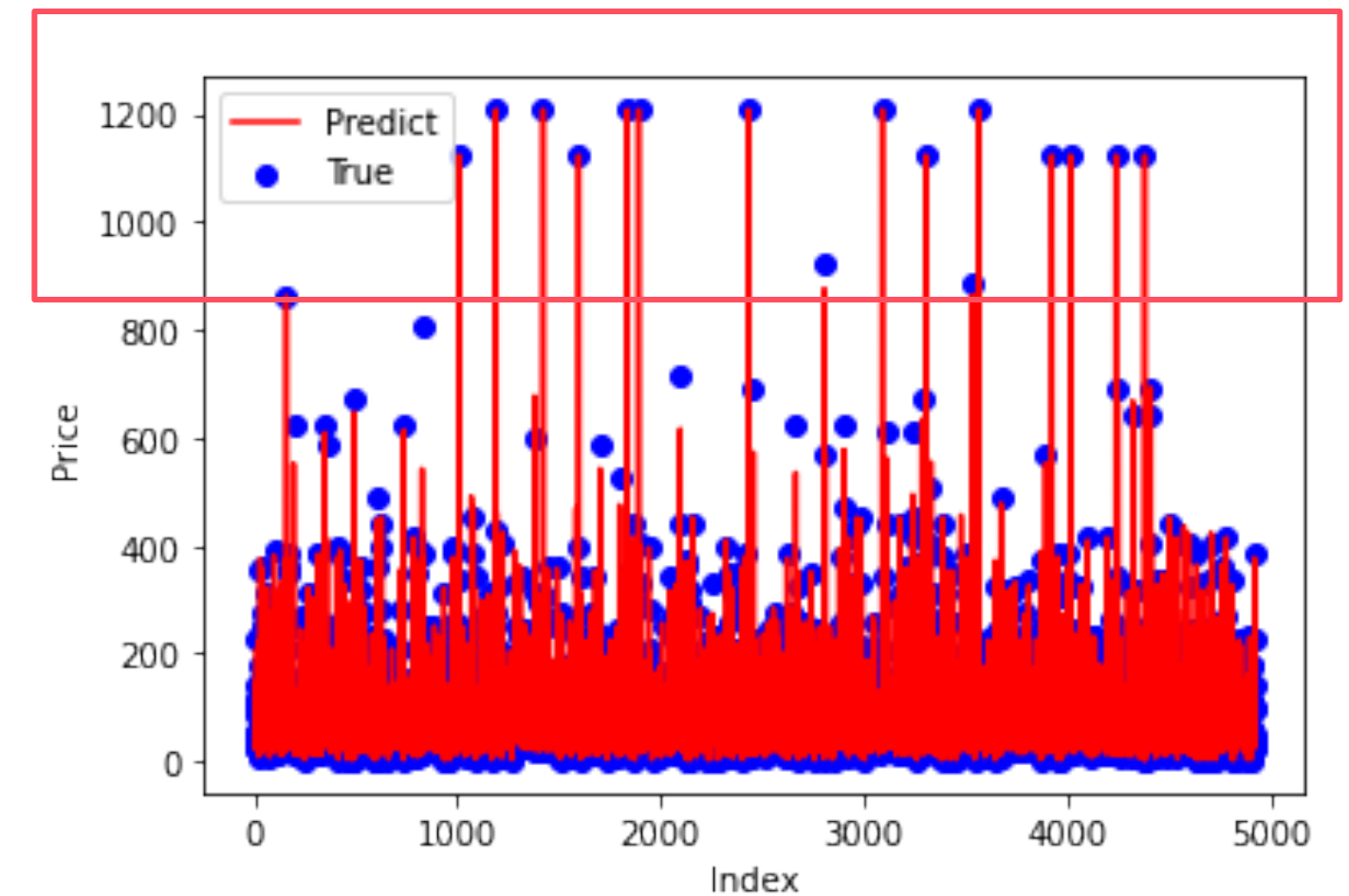
ทดลองเพื่อหาค่าที่ดีที่สุดของ พารามิเตอร์

- แบ่งข้อมูลออกเป็น 2 ส่วน คือ Training Set 80% และ Test Set 20%
- ในที่นี้เราสนใจพารามิเตอร์ C
- กำหนดให้ทดลองทั้งหมด 4 ค่า ได้แก่ 1, 10, 100, 1000
- ประสิทธิภาพการทำนายดีที่สุด เมื่อกำหนดพารามิเตอร์ C = 1000 และ kernel = 'RBF'

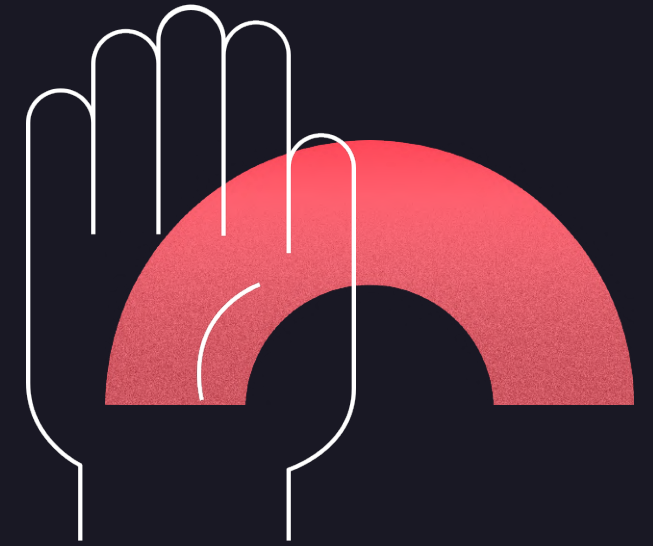
Support Vector Regression

MSE	162.39
MAPE	6.45

GenPrice	PredictedGenPrice
27.0	29.47
105.0	108.03
88.0	85.15
142.0	124.81
99.0	102.58
33.0	28.68
54.0	51.63
117.0	104.13
30.0	31.25
52.0	37.72



Feed-forward Neural Network



สร้างตัวแบบ ทำนายเชิงเส้น

- ใช้ฟังก์ชัน MLPRegressor จากไลบรารี scikit-learn
- แบ่งข้อมูลออกเป็น 2 ส่วน คือ Training Set 80% และ Test Set 20%

ทดลองเพื่อหาค่าที่ดีที่สุดของ พารามิเตอร์

- ในที่นี่เราสนใจพารามิเตอร์ hidden_layer_sizes
- กำหนดให้ทดลองทั้งหมด 4 ครั้ง ได้แก่
 - hidden_layer_sizes = (500, 500, 500, 500)
 - hidden_layer_sizes = (250, 250, 250, 250)
 - hidden_layer_sizes = (100, 100, 100, 100)
 - hidden_layer_sizes = (10, 10, 10, 10)

**hidden_layer_sizes =
(500, 500, 500, 500)**

**hidden_layer_sizes =
(250, 250, 250, 250)**

**hidden_layer_sizes =
(100, 100, 100, 100)**

**hidden_layer_sizes =
(10, 10, 10, 10)**

MSE

261.64

173.18

184.53

183.17

MAPE

10.12

6.84

7.15

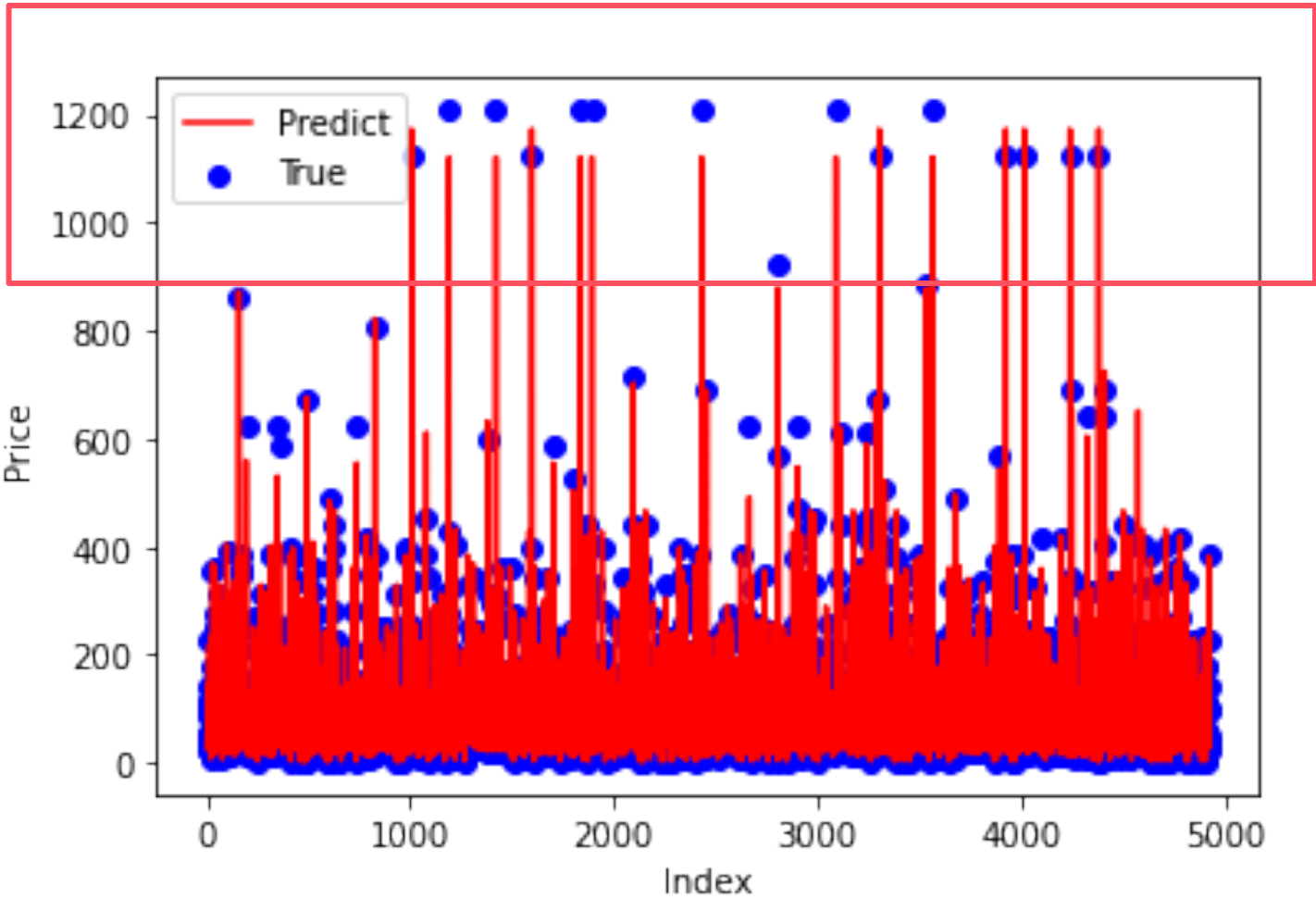
7.40

	hidden_layer_sizes = (500, 500, 500, 500)	hidden_layer_sizes = (250, 250, 250, 250)	hidden_layer_sizes = (100, 100, 100, 100)	hidden_layer_sizes = (10, 10, 10, 10)
MSE	261.64	173.18	184.53	183.17
MAPE	10.12	6.84	7.15	7.40

Feed-forward Neural Network

MSE	173.18
MAPE	6.84

GenPrice	PredictedGenPrice
27.0	30.51
105.0	101.99
88.0	83.53
142.0	123.29
99.0	95.62
33.0	28.65
54.0	51.78
117.0	90.66
30.0	34.87
52.0	38.87



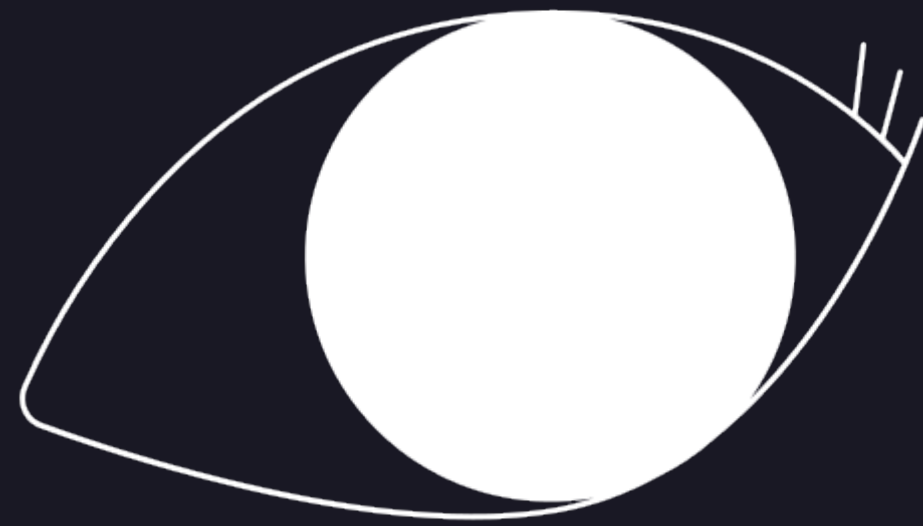
เปรียบเทียบ และ สรุปผลตัวแบบทำนาย

	Linear regression	Support vector regression	Feed-forward neural network
MSE	341.55	162.39	173.18
MAPE	10.29	6.45	6.84

สรุปได้ว่า

- เลือกชีพพอร์ตเวกเตอร์แมชชีนแบบถดถอย เป็นตัวแบบทำนายที่จะนำมาใช้ทำนายราคาสินค้าสำหรับชุดทดสอบ





บทที่ 4 : การทำนายราคาสินค้า

1. Taking Care of Missing Values

	SKU	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	date
276	E-A-00162	NaN	NaN	NaN	NaN	G	2020-03-27
325	A-K-00040	NaN	NaN	NaN	NaN	C	2020-03-27
441	E-A-00023	NaN	NaN	NaN	NaN	G	2020-03-27
446	E-A-00055	NaN	NaN	NaN	NaN	G	2020-03-27
447	E-A-00057	NaN	NaN	NaN	NaN	G	2020-03-27
500	A-E-00167	NaN	NaN	NaN	NaN	NaN	2020-03-27
501	A-E-00168	NaN	NaN	NaN	NaN	NaN	2020-03-27
502	A-E-00169	NaN	NaN	NaN	NaN	NaN	2020-03-27

1. Taking Care of Missing Values

testData1

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 489 entries, 0 to 491
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   SKU                   489 non-null    object
1   PrevAVGCost           489 non-null    float64
2   PrevAssignedCost      489 non-null    float64
3   AVGCost               489 non-null    float64
4   LatestDateCost        489 non-null    float64
5   Type                  489 non-null    object
6   date                  489 non-null    object
7   A                     489 non-null    uint8
8   B                     489 non-null    uint8
9   C                     489 non-null    uint8
10  D                     489 non-null    uint8
11  E                     489 non-null    uint8
12  F                     489 non-null    uint8
13  G                     489 non-null    uint8
dtypes: float64(4), object(3), uint8(7)
memory usage: 33.9+ KB
```

ไอ้แล้ว 492 รายการ

testData2

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 495 entries, 0 to 499
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   SKU                   495 non-null    object
1   PrevAVGCost           495 non-null    float64
2   PrevAssignedCost      495 non-null    float64
3   AVGCost               495 non-null    float64
4   LatestDateCost        495 non-null    float64
5   Type                  495 non-null    object
6   date                  495 non-null    object
7   A                     495 non-null    uint8
8   B                     495 non-null    uint8
9   C                     495 non-null    uint8
10  D                     495 non-null    uint8
11  E                     495 non-null    uint8
12  F                     495 non-null    uint8
13  G                     495 non-null    uint8
dtypes: float64(4), object(3), uint8(7)
memory usage: 34.3+ KB
```

ไอ้แล้ว 500 รายการ

2. Create Dummy Variables for Types

	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	A	B	C	D	E	F	G
0	22.646465	30.0	24.646465	23.767677	A	1	0	0	0	0	0	0
1	51.101010	56.0	49.565657	49.838384	A	1	0	0	0	0	0	0
2	40.010101	41.0	17.323232	38.727273	A	1	0	0	0	0	0	0
3	77.484848	81.0	78.545455	80.050505	B	0	1	0	0	0	0	0
4	77.484848	81.0	78.545455	80.050505	B	0	1	0	0	0	0	0
...
487	0.000000	60.0	0.000000	0.000000	G	0	0	0	0	0	0	1
488	0.000000	90.0	0.000000	0.000000	G	0	0	0	0	0	0	1
489	25.131313	25.0	0.000000	0.000000	G	0	0	0	0	0	0	1
490	181.818182	201.0	156.565657	181.818182	G	0	0	0	0	0	0	1
491	0.000000	70.0	0.000000	0.000000	G	0	0	0	0	0	0	1

3. Predict GenPrice with SVM model

GenPrice

41.942576

77.933715

55.737651

113.008982

113.008982

...

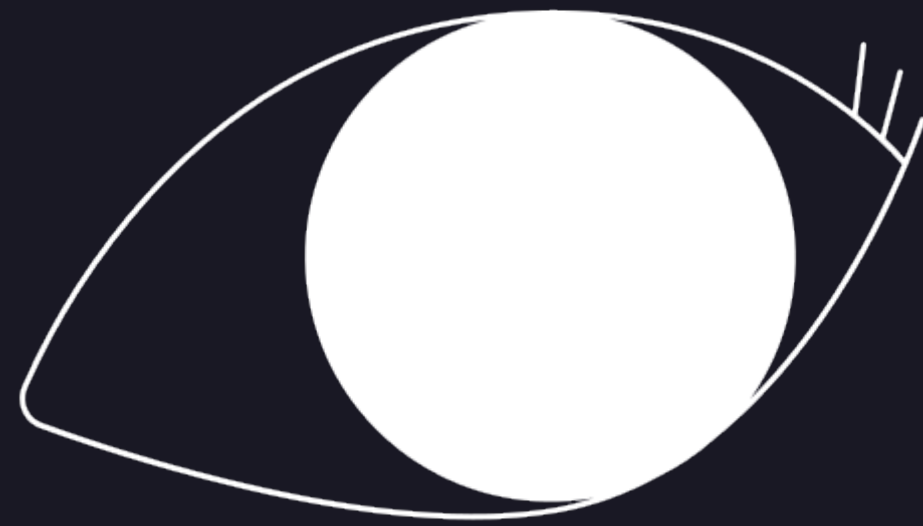
84.148757

123.455584

37.754660

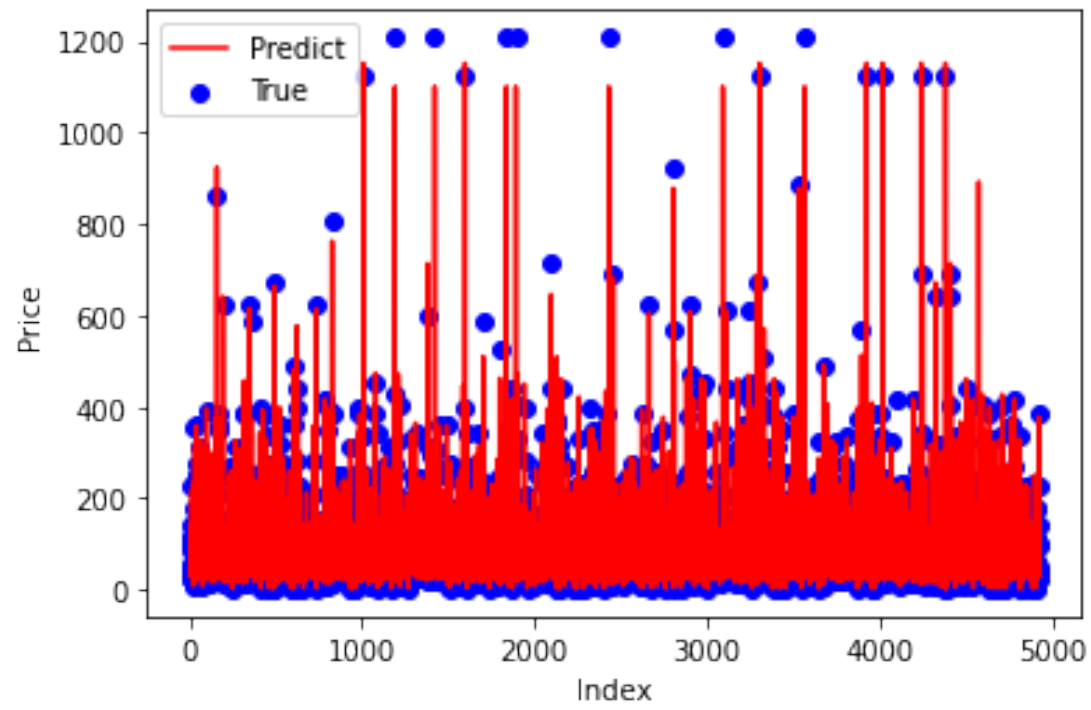
237.105699

97.353564



บทที่ 6 : บทสรุป

Linear Regression



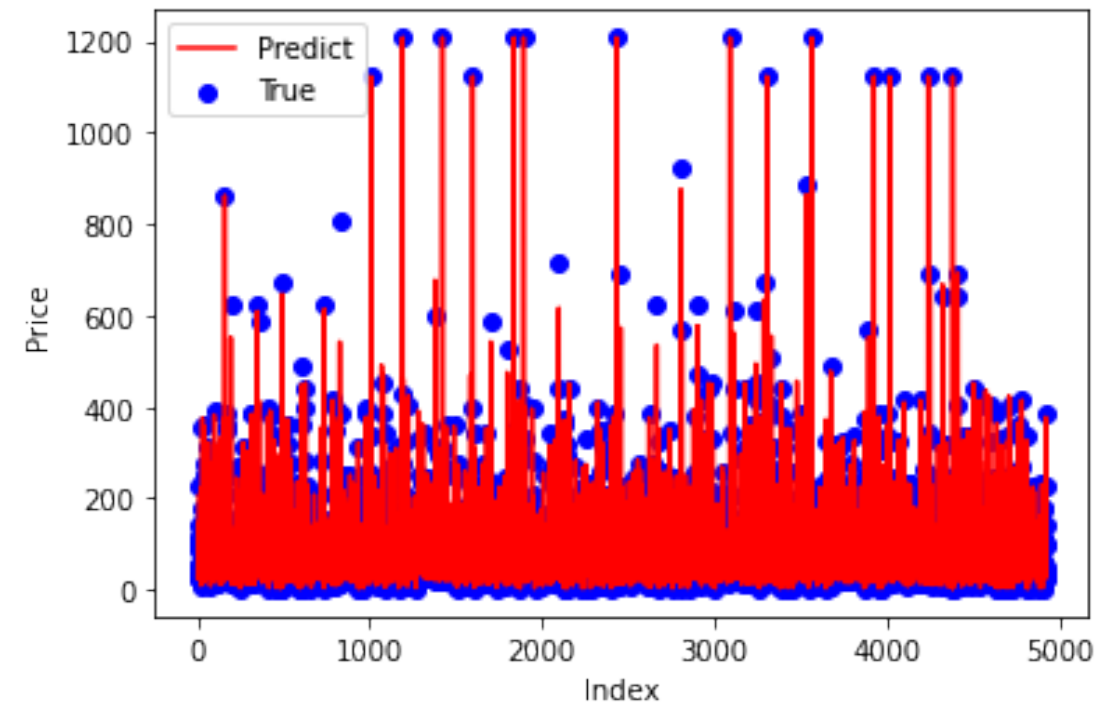
MSE

341.55

MAPE

10.29

SVR



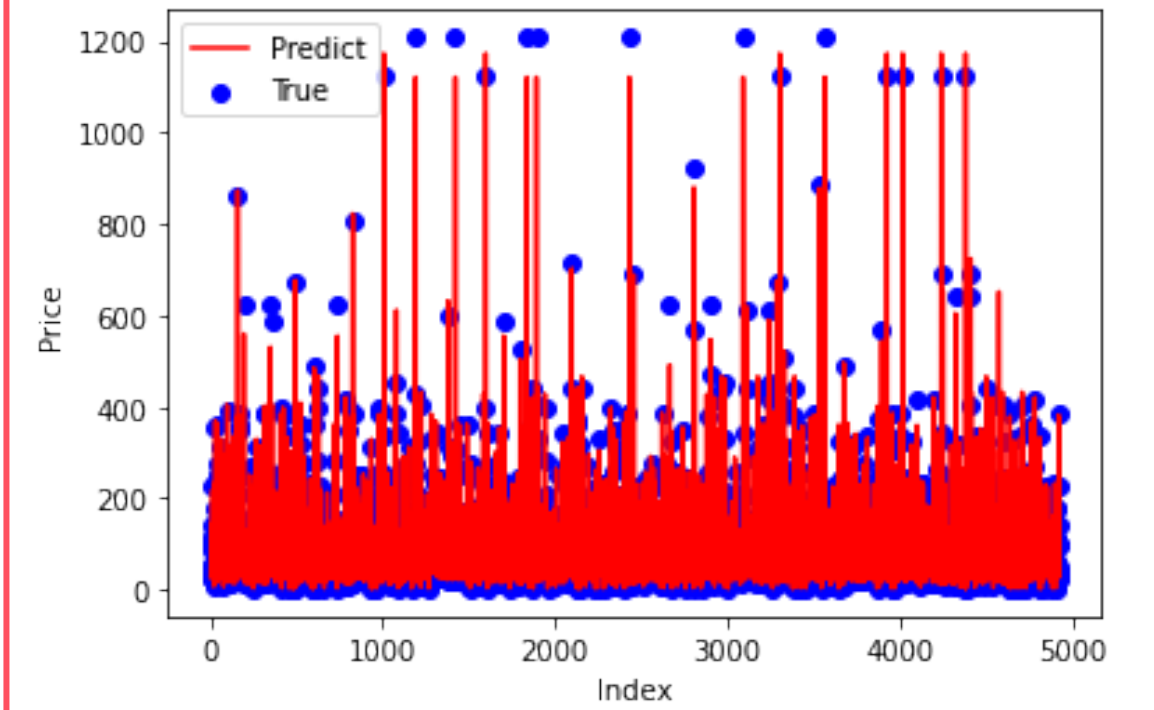
MSE

162.39

MAPE

6.45

FF Neural Network



MSE

172.98

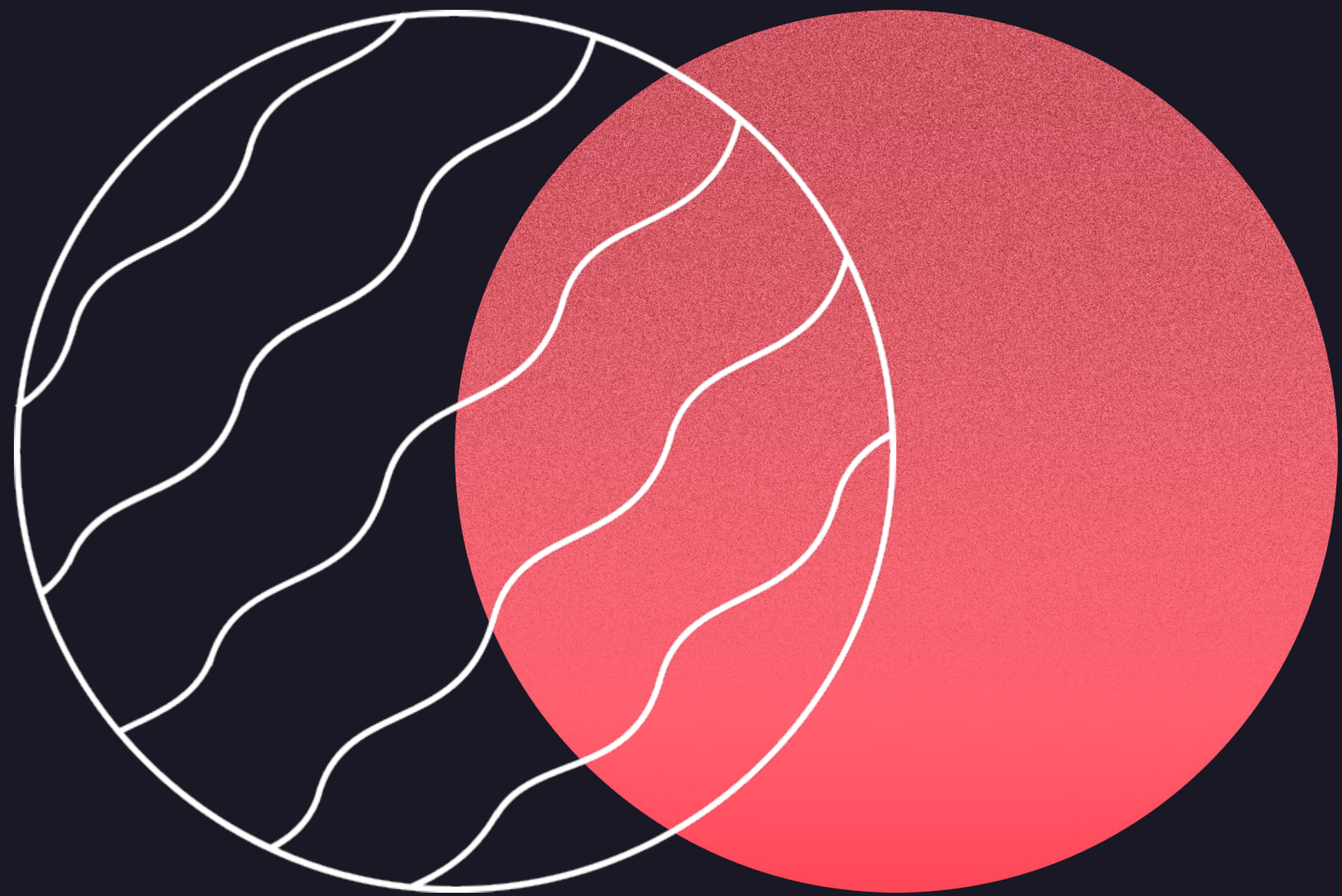
MAPE

7.34





1. ML Project Cleaning Data 6033657523.ipynb on Google Colaboratory
<https://colab.research.google.com/drive/15cYqMm6y8VWWPwhNnA2aFZJmPjUxIUZh>
2. ML Project Linear Regression 6033657523.ipynb on Google Colaboratory
<https://colab.research.google.com/drive/1eliO05XPm-mbUQlb9UDg8VqKah4xqXV>
3. ML Project SVR 6033657523.ipynb on Google Colaboratory
<https://colab.research.google.com/drive/1G4f8RtllYEFZPYS7POcfdsqGOvjpwdSx>
4. ML Project Feedforward neural network 6033657523.ipynb on Google Colaboratory
https://colab.research.google.com/drive/1CZFaD_0LdcZM3fXOt9gGK5uLTd4E0dR5



Thank You.