รายงานโครงการการตั้งราคาสินค้า

วุฒิภัทร คำนวนสินธุ์

20 เมษายน 2563

1. หลักการ และ เหตุผล

บริษัทแห่งหนึ่งมีการจัดจำหน่ายสินค้าในคลังประมาณ 500 รายการ (SKUs) โดยมีพนักงานขายรับหน้าที่เป็นผู้กำหนดราคาขายประจำสัปดาห์ให้กับสินค้าดังกล่าว ซึ่งพิจารณาจากราคาต้นทุนของสินค้าที่มีการเปลี่ยนแปลงเป็นประจำทุกวัน และส่วนต่างกำไรที่ไม่ได้กำหนดให้คงที่ในแต่ละครั้ง ทำให้การเปลี่ยนแปลงราคาต้นทุนสินค้านี้ส่งผลโดยตรงต่อผลกำไรโดยรวมของบริษัท ดังนั้นพนักงานฝ่ายขายจึงได้รับมอบหมายให้เป็นผู้กำหนดราคาสินค้าใหม่เป็นประจำทุกสัปดาห์

การกำหนดราคาสินค้าในแต่ละครั้งจำเป็นต้องพึ่งพาอำนาจการตัดสินใจของพนักงานคนดังกล่าวแต่เพียงผู้เดียว ฉะนั้นหากเกิดเหตุสุดวิสัย เช่น พนักงานคนดังกล่าวลาออกจากบริษัท ย่อมส่งผลกระทบโดยตรงต่อการกำหนดราคาสินค้า เนื่องด้วยบริษัทไม่สามารถเลียนแบบพฤติกรรมการตั้งราคาของพนักงานคนดังกล่าวได้ กล่าวคือ แนวคิด หลักการ วิธีคำนวณ และการตัดสินใจ เป็นสิ่งที่พนักงานคนดังกล่าวรับรู้แต่เพียงผู้เดียว ฉะนั้นเพื่อเป็นการรักษาเสถียรภาพของกลไกการตั้งราคาสินค้าของบริษัท จึงมีความจำเป็นต้องสร้างตัวแบบทำนายขึ้นเพื่อเลียนแบบพฤติกรรมการกำหนดราคาสินค้าของพนักงานดังกล่าว และเพื่อนำไปดำเนินการกำหนดราคาให้กับสินค้าในคลังของบริษัทต่อไป

โดยมีขั้นตอนการดำเนินโครงการ ดังนี้

1. วิเคราะห์ข้อมูล
2. ทำความสะอาดข้อมูลเพื่อนำมาเป็นชุดข้อมูลฝึกสอน
3. ทำการทดลองสร้างตัวแบบทำนาย
4. ทดสอบ และ วิเคราะห์ประสิทธิภาพของตัวแบบทำนาย
5. สรุปผล และ เลือกตัวแบบทำนายที่มีประสิทธิภาพมากที่สุด
6. วิเคราะห์ข้อมูลชุดทดสอบ และ ทำความสะอาดข้อมูล
7. ทำนายข้อมูลที่ต้องการจากข้อมูลชุดทดสอบ โดยใช้ตัวแบบทำนายที่เลือกไว้
8. สรุปผลการดำเนินงาน และจัดทำรายงาน

2. การทำความสะอาดข้อมูล

บริษัทให้ข้อมูลการตั้งราคาสินค้ามาทั้งหมด 50 สัปดาห์ ในแต่ละสัปดาห์มีข้อมูล ดังนี้

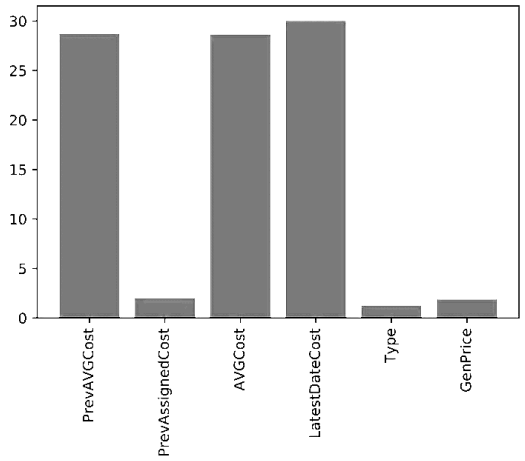
1. รหัสสินค้า (SKU) = ชนิดสินค้า – ชนิดย่อยสินค้า – เลขลำดับที่
2. วันที่ตั้งราคา (Date)
3. ราคาต้นทุนซื้อเฉลี่ยของสัปดาห์ก่อนหน้า (PrevAVGCost)
4. ราคาต้นทุนหลังหักส่วนสูญเสียของสัปดาห์ก่อนหน้า (PrevAssignedCost)
5. ราคาต้นทุนซื้อเฉลี่ยของทั้งสัปดาห์ (AVGCost)
6. ราคาต้นทุนซื้อวันล่าสุด (LatestDateCost)
7. ชนิดของสินค้า (Type)
8. ราคาตั้งของสัปดาห์นี้ (GenPrice)

เมื่อพิจารณาข้อมูลเบื้องต้น พบว่ามีข้อมูลการตั้งราคา 25116 รายการจาก 50 สัปดาห์ มีสินค้าทั้งหมด 529 รายการ (SKUs) อย่างไรก็ตาม พบว่ามีข้อมูลบางส่วนที่ไม่ได้มีการบันทึกไว้ หรือบันทึกไว้เป็น 0 เนื่องจากสัปดาห์นั้นอาจจะไม่มีการซื้อวัตถุดิบ จึงไม่มีราคาต้นทุน รวมถึงสินค้าบางชิ้นไม่มีการบันทึกข้อมูลของการตั้งราคาสินค้าอีกด้วย ซึ่งคาดเดาได้ว่าในสัปดาห์นั้นไม่มีการผลิตสินค้าชนิดดังกล่าว

	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

รูปที่ 1 : ตารางแสดงตัวอย่างข้อมูลที่ขาดหายไป

เมื่อนำข้อมูลไปทำการวิเคราะห์ พบว่าลักษณะการขาดหายของข้อมูล สามารถจำแนกได้ 2 รูปแบบ ได้แก่ รูปแบบที่ 1 คือ การบันทึกข้อมูลเป็น 0 และ รูปแบบที่ 2 คือ ไม่มีการบันทึกข้อมูล (NaN) ดังรูปที่ 1 และเมื่อนำข้อมูลมาแจกแจง และทำการนับจำนวนของข้อมูลที่ขาดหายไป จากนั้นคำนวณเป็นร้อยละ แสดงผลในรูปแบบของแผนภูมิแท่ง จะได้ผลลัพธ์ดังรูปที่ 2



รูปที่ 2 : ร้อยละข้อมูลที่ขาดหายไป

ข้อมูลประมาณร้อยละ 28 ไม่มีต้นทุนซื้อเฉลี่ยของสัปดาห์ปัจจุบัน (AVGCost) หรือสัปดาห์ก่อนหน้า (PrevAVGCost) และข้อมูลร้อยละ 30 ไม่มีราคาต้นทุนซื้อของวันล่าสุด (LatestDateCost) ต้นทุนหลักหักส่วนสูญเสียของสัปดาห์ก่อนหน้านั้น (PrevAssignedCost) ขาดไปร้อยละ 2 ซึ่งอาจเกิดจากผู้ตั้งราคาสำเนาค่ามาใส่ โดยไม่ได้คำนวณมา และอาจจะทำไม่ครบ

ร้อยละ 1.25 ของรายการไม่มีราคาตั้ง (GenPrice) ซึ่งเป็นไปได้ว่าไม่มีการขายสินค้านั้นในสัปดาห์นั้น เนื่องจากราคาตั้งนี้เป็นราคาเป้าหมายที่ต้องการทำนาย เมื่อไม่มีราคาตั้ง เราจึงไม่สามารถทำนายได้ จึงจะตัดรายการเหล่านี้ออกทั้งหมด เหลือ 24639 รายการหลังจากนั้น พิจารณาสถิติเบื้องต้นของข้อมูลที่เหลืออยู่ พบว่า

1. มี 1 รายการที่ราคาทุนหลังหักส่วนสูญเสียของสัปดาห์ก่อนหน้าสูงถึงสิบล้านบาท และอีก 1 รายการที่ราคาตั้งสูงถึงสิบล้านบาท ซึ่งผิดปกติ และอาจจะเกิดจากข้อผิดพลาดขณะกรอกข้อมูล จึงตัดรายการทั้งสองนี้ทิ้งไป
2. พิจารณาร้อยละของราคาที่บวกเพิ่มขึ้นจากราคาทุน แต่เนื่องจากราคาทุนเฉลี่ยนั้นขาดหายไปเป็นจำนวนมาก จึงคิดเบื้องต้นจากราคาทุนหลังหักส่วนสูญเสียของสัปดาห์ก่อนหน้าแทน ดังสมการ

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

ส่วนกรณีที่ไม่มีราคาทุนหลังหักส่วนสูญเสีย หากยังมีราคาทุนเฉลี่ย ให้ใช้ราคาเฉลี่ยไปก่อน ส่วนกรณีอื่นๆ ซึ่งไม่มีราคาทั้งสองให้ตัดทิ้งทั้งหมด หลังตัดข้อมูลที่ไม่สามารถหาส่วนต่างการบวกราคาได้แล้ว พบว่าส่วนต่างนี้กระจายอยู่ในช่วงร้อยละ -60 ถึง 610 แต่มีอยู่หนึ่งรายการที่มีส่วนต่างสูงถึงร้อยละ 1900 ซึ่งเมื่อดูในรายละเอียดแล้ว พบว่าราคาทุนหลังหักส่วนสูญเสียต่ำเกินความ

เป็นจริง แต่ราคาเฉลี่ยของสัปดาห์ปัจจุบันของรายการนี้ไม่ผิดปกติ จึงกำหนดให้ข้อมูลราคาทุนหลังหักส่วนสูญเสียของรายการนี้เป็นข้อมูลที่ขาดหายไปแทน

```
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
```

รูปที่ 3 : สถิติที่ได้จากการคำนวณค่า Margin

เมื่อทำการพิจารณารหัสสินค้า (SKU) พบว่าบริษัทกำหนดว่ารหัสสินค้าแต่ละชนิดนั้นมีโครงสร้างดังนี้

ชื่อกลุ่มสินค้าหลัก — ชื่อกลุ่มสินค้าย่อย — ลำดับของสินค้าในกลุ่มย่อย

เช่น A-A-00001 หมายถึง สินค้ากลุ่ม A กลุ่มย่อย A ลำดับที่ 1 ซึ่งสินค้ากลุ่มย่อยแต่ละกลุ่มนั้นมักจะใช้วัตถุดิบเดียวกันหรือคล้ายๆ กัน จึงจะสกัดชื่อกลุ่มสินค้าย่อยนี่ออกมาด้วย พบว่า สินค้ากลุ่มหลักมีทั้งหมด 2 กลุ่ม ได้แก่ A และ E โดยกลุ่ม A มี 11 กลุ่มย่อย และกลุ่ม E มี 2 กลุ่มย่อย

จากข้อมูลที่ได้รับมาสามารถแปลงรหัสสินค้า (SKU) ให้เป็นคอลัมน์ กลุ่มหลัก (Category) และ กลุ่มย่อย (Sub Category) โดยมีจำนวนสมาชิกที่เป็นไปได้ทั้งหมด 2 และ 11 ค่า ตามลำดับ ได้แก่ A, E สำหรับกลุ่มหลัก และ A, B, C, D, E, F, G, H, I, J, K สำหรับกลุ่มย่อย ซึ่งเก็บข้อมูลอยู่ในรูปแบบ String ทั้งหมด

เมื่อทำการพิจารณาชนิดของสินค้า (Type) พบว่ามีจำนวนสมาชิกที่เป็นไปได้ทั้งหมด 7 ค่า ได้แก่ A, B, C, D, E, F, G ซึ่งเก็บข้อมูลอยู่ในรูปแบบ String ทั้งหมด

จึงจำเป็นต้องทำการแปลงข้อมูลในคอลัมน์ทั้ง 3 ได้แก่ กลุ่มหลัก กลุ่มย่อย และ ชนิดของสินค้า ซึ่งเก็บในรูปแบบ String ให้อยู่ในรูปแบบ ที่สามารถนำมาใช้ฝึกสอนตัวแบบทำนายได้ เช่น ตัวเลข 0 และ 1 ที่เรียกว่า **การสร้างตัวแปรหุ่น (Dummy Variables)** โดยผลลัพธ์ของการสร้างตัวแปรหุ่น จะทำให้มีคอลัมน์เพิ่มขึ้นมาเท่ากับค่าที่เป็นเป็นได้ทั้งหมดของคอลัมน์นั้น เช่น การสร้างตัวแปรหุ่นสำหรับชนิดของสินค้า (Type) จะทำให้เกิดคอลัมน์เพิ่มมา 7 คอลัมน์ ได้แก่ A, B, C, D, E, F, G โดยการบ่งชี้ว่าสินค้ารายการหนึ่งๆ เป็นสินค้าชนิดใด จะเปลี่ยนไป เช่น สินค้าชนิด A จะถูกบ่งชี้ด้วย Dummy Variables คือ 1 0 0 0 0 0 0 ในแต่ละคอลัมน์ A, B, C, D, E, F, G ตามลำดับ ทำเช่นนี้กับข้อมูลในคอลัมน์ทั้ง 3 ที่ได้กล่าวไป

จากข้อมูลการตั้งราคาสินค้า และ วันที่ตั้งราคา (Date) อาจมีความสัมพันธ์กันบางอย่าง ทำให้ไม่อาจตัดสินใจตัดคอลัมน์วันที่ตั้งราคาทิ้งไปได้ จึงได้มีการแปลงข้อมูลวันที่ตั้งราคา ซึ่งอยู่ในรูปแบบ YYYY-MM-DD เช่น 2019-01-04 ให้อยู่ในรูปแบบของปี และทำการเก็บในคอลัมน์ปี (Year) และทำการสร้างตัวแปรหุ่น เพื่อทำการวิเคราะห์ต่อไป โดยสันนิษฐานว่าการเพิ่มคอลัมน์ดังกล่าว อาจเพิ่มประสิทธิภาพให้การทำนายของโมเดลได้ ทำการทดลองในหัวข้อที่ 4

หลังจากพิจารณาเพิ่มคอลัมน์ที่จำเป็น และทำการตัดแถวข้อมูลในทุกกรณีแล้ว พบว่าเราเหลือข้อมูลที่สามารถนำไปทำนายการตั้งราคา ได้ทั้งหมด 24626 รายการ จากนั้นจึงทำการบันทึกข้อมูล และนำออก (Export) ให้อยู่ในรูปแบบไฟล์ .csv เพื่อนำไปเป็นชุดข้อมูลฝึกสอนโมเดลต่อไป

	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 × 27 columns

รูปที่ 4 : ตัวอย่างชุดข้อมูลฝึกสอนที่ได้หลังจากการทำความสะอาดข้อมูล

ชุดข้อมูลฝึกสอนที่ได้หลังการทำความสะอาดของข้อมูล มีลักษณะดังต่อไปนี้

```
<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
```

รูปที่ 5 : ลักษณะของชุดข้อมูลฝึกสอน

3. การทดลอง

การทำนายการกำหนดราคาสินค้า อาจพิจารณาเป็น**งานประเภทถดถอย (Regression task)** ซึ่งการวิเคราะห์แบบถดถอยเป็นเทคนิคที่ใช้ข้อมูลทางสถิติในการประเมินความสัมพันธ์ระหว่างตัวแปรเป้าหมาย ในโครงการนี้ ได้แก่ ราคาสินค้าที่ตั้ง (GenPrice) และ ตัวแปรอิสระ ซึ่งมีความสัมพันธ์กับตัวแปรเป้าหมาย ในโครงการนี้ ได้แก่ ตัวแปรอื่นๆ ทั้งหมดนอกเหนือจากตัวแปรเป้าหมาย

โครงการนี้จะใช้**วิธีการวิเคราะห์เชิงทำนาย (Predictive analysis)** โดยอาศัยการวิเคราะห์ข้อมูลจากในอดีต และปัจจุบัน รวมถึงอาศัยความสามารถในการเรียนรู้ของเครื่อง (Machine Learning) เพื่อคาดเดาความน่าจะเป็นของเหตุการณ์ที่จะเกิดขึ้นในอนาคต โดยข้อมูลที่เป็นข้อมูลรับเข้า (Input) ได้แก่ ราคาต้นทุนสินค้าในสัปดาห์ก่อนหน้า สัปดาห์ปัจจุบัน และประวัติการตั้งราคาสินค้าในอดีตมาประกอบกัน เพื่อทำนายราคาสินค้าที่ควรจะเป็นในสัปดาห์ถัดไป

โครงการนี้จะทดลองการสร้างตัวแบบทำนายโดยใช้ตัวแบบต่อไปนี้

1. ตัวแบบทำนายเชิงเส้น (Linear regression)
2. ซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย (Support vector regression)
3. ข่ายงานประสาทเทียมแบบป้อนหน้า (Feed-forward neural network)

3.1. ตัวแบบทำนายเชิงเส้น (Linear regression)

ตัวแบบทำนายเชิงเส้น เป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรตั้งแต่ 2 ตัวขึ้นไป เมื่อพิจารณาจากฐานข้อมูลการกำหนดราคาสินค้าของบริษัท พบว่ามีจำนวนของข้อมูลที่มากพอ เหมาะสมที่จะนำมาใช้เป็นตัวอย่าง (Sample) ในการทำ Linear regression และเมื่อมีค่าประมาณการ (Predictor) มากกว่า 1 ตัว จะเรียกว่า Multiple linear regression มีรูปแบบสมการ ดังนี้

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \varepsilon$$

โดยกำหนดให้ตัวแปร y เป็นข้อมูลจากคอลัมน์ราคาตั้งของสัปดาห์นี้ (GenPrice) และตัวแปร X เป็นข้อมูลจากคอลัมน์ทั้งหมดที่เหลือ จากนั้นทำการแบ่งข้อมูลชุดทดสอบออกเป็นข้อมูลฝึกสอน (Training set) และข้อมูลทดสอบ (Test set) โดยกำหนดให้อัตราส่วนของข้อมูลทั้งสอง เป็น 80:20 และ random_state = 0

ในการสร้างตัวแบบทำนายเชิงเส้นนี้ ได้ใช้คลาส linear_model ของ scikit-learn ด้วยคำสั่ง LinearRegression() และมีการทำ Grid Search เพื่อหาค่าที่ดีที่สุดของพารามิเตอร์ ในที่นี้เราสนใจ พารามิเตอร์ fit_intercept และ normalize ซึ่งพารามิเตอร์ทั้ง 2 ตัว มีค่าเป็น Boolean

นอกจากนี้ยังมีการทำ Backward Elimination เพื่อกำจัดตัวแปรที่ไม่มีผลต่อสมการเส้นตรง และความแม่นยำของการทำนาย และทำการฝึกสอนตัวแบบทำนายใหม่อีกครั้ง แต่จากผลการทดลองพบว่า ตัวแปรทุกตัวล้วนมีผลต่อสมการเส้นตรง กล่าวคือ ไม่มีตัวแปรใดที่มีค่า P > | t | ที่มากกว่า 0.50 จึงไม่สามารถตัดตัวแปรใดๆ ออกไปเพื่อเพิ่มประสิทธิภาพให้กับตัวแบบทำนายเชิงเส้นนี้ได้

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	0.381	-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

รูปที่ 6 : สถิติของตัวแปรที่ใช้ในการทำนายตัวแบบทำนายเชิงเส้น

3.2. ซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย (Support vector regression)

ซัพพอร์ตเวกเตอร์แมชชีน เทคนิคหนึ่งที่ได้รับความ นิยมอย่างแพร่หลายในงานที่เกี่ยวข้องกับการจดจำรูปแบบตลอดจนการแก้ปัญหาการจัดกลุ่ม (Classification problem) โดยอาศัยหลักการของการหาสัมประสิทธิ์ของสมการเพื่อสร้างเส้นแบ่งแยกกลุ่มข้อมูลที่ถูกป้อนเข้าสู่กระบวนการสอนให้ระบบเรียนรู้ โดยเน้นไปยังเส้นแบ่งแยกกลุ่มข้อมูลได้ดีที่สุด (Optimal separating hyperplane) เมื่อเราพิจารณาข้อมูล ที่ประกอบด้วยข้อมูล 2 กลุ่ม ซึ่งเป็นการกำหนดกลุ่มเป้าหมายให้ SVM โดยที่ SVM นั้นมุ่งเป้าเพื่อหาฟังก์ชันการตัดสินใจที่ สามารถแบ่งแยกค่าที่ไม่ทราบได้

วิธีการที่ใช้ในการหาเส้นแบ่งที่ดีที่สุดคือการเพิ่มเส้นขอบ (Margin) ให้กับเส้นแบ่งทั้งสองข้าง และสร้างเส้นขอบที่สัมผัสกับค่าข้อมูลใน feature space ที่ใกล้ที่สุดดังนั้นเส้นแบ่งที่มีเส้นขอบกว้างที่สุดจึงเป็นเส้นแบ่งที่ดีที่สุดและเรียกตำแหน่งการสัมผัสข้อมูลที่ใกล้ที่สุดจากการเพิ่มขอบนี้ว่า ซัพพอร์ตเวกเตอร์ (Support vector) เนื่องจากในบางกรณีการแบ่งแยกกลุ่มไม่สามารถทำได้ถูกต้องโดยสมบูรณ์ ดังนั้นจึงต้องมีการกำหนดตัวแปรสำหรับยอมรับค่าความผิดพลาด

โดยกำหนดให้ตัวแปร y เป็นข้อมูลจากคอลัมน์ราคาตั้งของสัปดาห์นี้ (GenPrice) และตัวแปร X เป็นข้อมูลจากคอลัมน์ทั้งหมดที่เหลือ จาดนั้นทำการแบ่งข้อมูลชุดทดสอบออกเป็นข้อมูลฝึกสอน (Training set) และข้อมูลทดสอบ (Test set) โดยกำหนดให้อัตราส่วนของข้อมูลทั้งสอง เป็น 80:20 และ random_state = 0

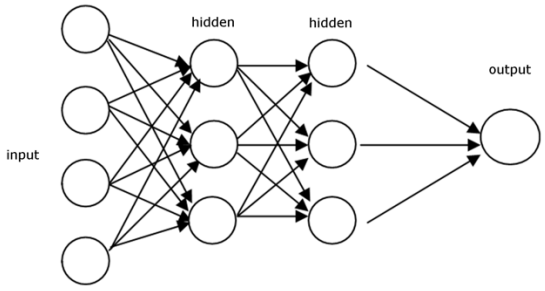
ในการสร้างซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย ด้วยฟังก์ชันเคอร์เนล Radial basis function (RBF) ซึ่งเป็นฟังก์ชันเคอร์เนลที่มีจุดศูนย์กลางอยู่จุดนึง ที่จุดนั้นฟังก์ชันนั้นมีค่าสูงสุด และเมื่อไกลออกไปจะมีค่าต่ำลง โดยสมการฟังก์ชัน RBF นั้นมีหลายรูปแบบ แต่ที่เป็นพื้นฐานและนิยมใช้กันอย่างแพร่หลาย คือ สมการ Gaussian

$$k(x, y) = \exp(-\gamma \|x - y\|^2)$$

ซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอยนี้ ได้ใช้ฟังก์ชัน SVR() ของคลาส svm จากไลบรารี scikit-learn โดยมีการทำ Grid Search เพื่อหาค่าที่ดีที่สุดของพารามิเตอร์ ในที่นี้เราสนใจ พารามิเตอร์ C ในที่นี้กำหนดให้ทดลองทั้งหมด 4 ค่า ได้แก่ 1, 10, 100, 1000 โดยกำหนดพารามิเตอร์ kernel = ‘rbf’

3.3. ข่ายงานประสาทเทียมแบบป้อนหน้า (Feed-forward neural network)

ข่ายงานประสาทเทียมแบบป้อนหน้าเป็นรูปแบบของ Neural Network ที่ง่ายที่สุดกล่าวคือ Perceptron จะถูกแบ่งออกเป็นกลุ่มๆ โดยแต่ละกลุ่มจะเรียกเป็น Layer โดยข้อมูลที่เข้ามาจะไหลไปในทิศทางเดียว ไม่ไหลย้อนกลับ จาก Layer หนึ่งไปสู่อีก Layer หนึ่ง



รูปที่ 7 : โครงสร้างของ Feed-forward neural network

โดยกำหนดให้ตัวแปร y เป็นข้อมูลจากคอลัมน์ราคาตั้งของสัปดาห์นี้ (GenPrice) และตัวแปร X เป็นข้อมูลจากคอลัมน์ทั้งหมดที่เหลือ จาดนั้นทำการแบ่งข้อมูลชุดทดสอบออกเป็นข้อมูลฝึกสอน (Training set) และข้อมูลทดสอบ (Test set) โดยกำหนดให้อัตราส่วนของข้อมูลทั้งสอง เป็น 80:20 และ random_state = 0

ในการสร้างข่ายงานประสาทเทียมแบบป้อนหน้า ได้ใช้ฟังก์ชัน MLPRegressor จากไลบรารี scikit-learn โดยกำหนดค่าค่าพารามิเตอร์ activation = 'relu', solver = 'adam', max_iter = 500 และ learning_rate = 'adaptive' และทำการทดลองเพื่อหาค่าที่ดีที่สุดของพารามิเตอร์ ในที่นี้เราสนใจพารามิเตอร์ hidden_layer_sizes กำหนดให้ทดลองทั้งหมด 4 ครั้ง ได้แก่

1. hidden_layer_sizes = (500, 500, 500, 500)
2. hidden_layer_sizes = (250, 250, 250, 250)
3. hidden_layer_sizes = (100, 100, 100, 100)
4. hidden_layer_sizes = (10, 10, 10, 10)

4. ผลการพัฒนา และ วัดผลตัวแบบทำนาย

การฝึกสอนตัวแบบจะใช้ฟังก์ชันเป้าหมายเป็น **ค่าผิดพลาดกำลังสอง (Mean squared error: MSE)** โดยเราต้องการค่าผิดพลาดกำลังสองที่น้อยที่สุด และวัดความถูกต้องของตัวแบบโดยใช้ **ร้อยละความผิดพลาดสัมบูรณ์เฉลี่ย (Mean absolute percentage error: MAPE)** ซึ่งเป็นค่าพยากรณ์ความแม่นยำในการทำนายของการวิเคราะห์จากสถิติ และยังใช้ฟังก์ชันการสูญเสีย (loss function) สำหรับงานประเภทถดถอย (Regression task)

$$MSE = \frac{1}{n}\Sigma(y - \hat{y})^2$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y_i}}{y_i}\right|$$

4.1. ผลการพัฒนาตัวแบบทำนายเชิงเส้น (Linear regression)

จากผลการทดสอบตัวแบบทำนายเชิงเส้นโดยใช้ค่าพารามิเตอร์ที่ดีที่สุดที่ได้จากการทำ Grid Search พบว่าตัวแบบทำนายเชิงเส้นมีประสิทธิภาพการทำนายดีที่สุดเมื่อกำหนดพารามิเตอร์ fit_intercept = True และ normalize = False จากนั้นจึงทำการทดลองโดยใช้ชุดข้อมูลทดสอบที่ต่างกันทั้งหมด 3 ชุด เพื่อหาว่าข้อมูลในคอลัมน์ใดที่ไม่มีผลต่อประสิทธิภาพของตัวแบบทำนายเชิงเส้น ดังนี้

การทดลองฝึกสอนตัวแบบทำนายเชิงเส้นครั้งที่ 1 โดยใช้ชุดข้อมูลทดสอบชุดที่ 1 ซึ่งประกอบไปด้วยข้อมูลทั้งหมด 27 คอลัมน์ ได้แก่ Index, PrevAVGCost, PrevAssignedCost, AVGCost, LatestDateCost, Type ตั้งแต่ A ถึง G (7 คอลัมน์), Category A และ E (2 คอลัมน์), Sub Category ตั้งแต่ A ถึง K (11 คอลัมน์), Year 2019 และ 2020 (2 คอลัมน์) ซึ่งมีรายละเอียดการทำความสะอาดข้อมูล ดังที่ได้กล่าวไปในหัวข้อที่ 2 ได้ค่าผิดพลาดกำลังสอง ร้อยละความผิดพลาดสัมบูรณ์เฉลี่ย ค่าความแม่นยำ (Accuracy) ของ Training set และ Test set ของการทดสอบดังนี้

Mean squared error: MSE	340.62457907161854
Mean absolute percentage error: MAPE	10.426928878335127
Training set accuracy	0.9745747838580836
Test set accuracy	0.966365303540365

การทดลองฝึกสอนตัวแบบทำนายเชิงเส้นครั้งที่ 2 โดยใช้ชุดข้อมูลทดสอบชุดที่ 2 ซึ่งประกอบไปด้วยข้อมูลทั้งหมด 25 คอลลัมน์ คือได้มีการตัดคอลัมน์ Year 2019 และ 2020 จำนวน 2 คอลัมน์ ออกไปจากข้อมูลทดสอบชุดที่ 1 ได้ผลการทดสอบดังนี้

Mean squared error: MSE	340.6658543632263
Mean absolute percentage error: MAPE	10.41205751001983
Training set accuracy	0.9745718167010757
Test set accuracy	0.9663612278453335

การทดลองฝึกสอนตัวแบบทำนายเชิงเส้นครั้งที่ 3 โดยใช้ชุดข้อมูลทดสอบชุดที่ 3 ซึ่งประกอบไปด้วยข้อมูลทั้งหมด 8 คอลลัมน์ คือได้มีการตัดคอลัมน์ Category A และ E (2 คอลัมน์), Sub Category ตั้งแต่ A ถึง K (11 คอลัมน์), Year 2019 และ 2020 (2 คอลัมน์) จำนวนทั้งสิ้น 15 คอลัมน์ ออกไปจากข้อมูลทดสอบชุดที่ 1 ได้ผลการทดสอบดังนี้

Mean squared error: MSE	341.5515637134753
Mean absolute percentage error: MAPE	10.297197104084978
Training set accuracy	0.9744550159807572
Test set accuracy	0.9662737691973751

จากผลการทดสอบตัวแบบทำนายด้วยชุดข้อมูลทดสอบที่มีจำนวนคอลัมน์แตกต่างกัน 3 ชุด ทำให้ทราบว่าประสิทธิภาพของตัวแบบทำนายที่ได้ แตกต่างกันไม่เกิน 1.0 จึงสรุปได้ว่า ข้อมูลปี (Year) กลุ่มหลัก (Category) และ กลุ่มย่อย (Sub Category) ไม่มีผลต่อประสิทธิภาพของตัวแบบทำนาย การพัฒนาตัวแบบทำนายหลังจากนี้เป็นต้นไป จะใช้ชุดข้อมูลทดสอบชุดที่ 3 ซึ่งประกอบด้วยข้อมูลทั้งหมด 11 คอลัมน์ ได้แก่ PrevAVGCost, PrevAssignedCost, AVGCost, LatestDateCost, Type ตั้งแต่ A ถึง G (7 คอลัมน์)

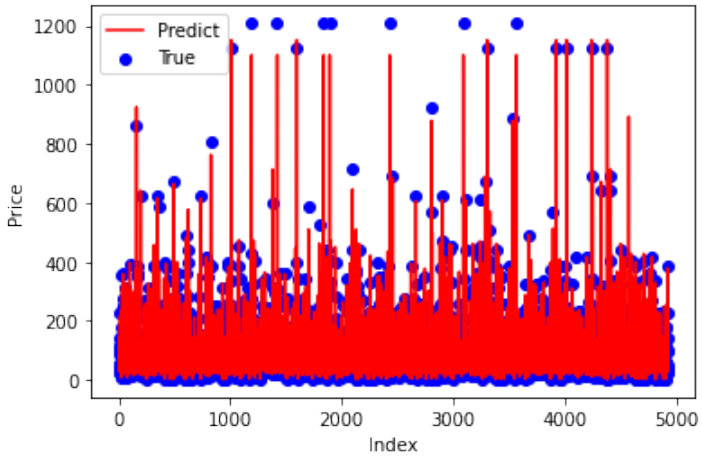
	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	A	B	C	D	E	F	G
0	27.919192	33.0	28.545455	20.535354	1	0	0	0	0	0	0
1	57.333333	63.0	56.121212	61.838384	1	0	0	0	0	0	0
2	50.777778	54.0	45.131313	50.000000	1	0	0	0	0	0	0
3	45.747475	56.0	40.525253	38.080808	0	1	0	0	0	0	0
4	45.747475	56.0	40.525253	38.080808	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

24626 rows × 11 columns

รูปที่ 8 : ตัวอย่างข้อมูลจากชุดข้อมูลทดสอบ

เมื่อพิจารณาผลการทำนายของตัวแบบทำนายเชิงเส้นที่ดีที่สุด พบว่าสามารถทำนายราคาสินค้าได้ใกล้เคียงกับราคาจริงในชุดข้อมูลฝึกสอน แต่เมื่อวิเคราะห์จากแผนภูมิพบว่า ประสิทธิภาพการทำนายโดยรวมยังไม่ดีนัก โดยเฉพาะอย่างยิ่งข้อมูลในช่วงราคา 1000 ขึ้นไป หรือกล่าวคือตัวแบบทำนายนี้ยังรับมือกับข้อมูลที่เป็น Outlier ได้ไม่ดีนัก

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



รูปที่ 9 : ตัวอย่างผลการทำนาย และ แผนภูมิแสดงประสิทธิภาพของตัวแบบทำนายเชิงเส้น

4.2. ผลการพัฒนาซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย (Support vector regression)

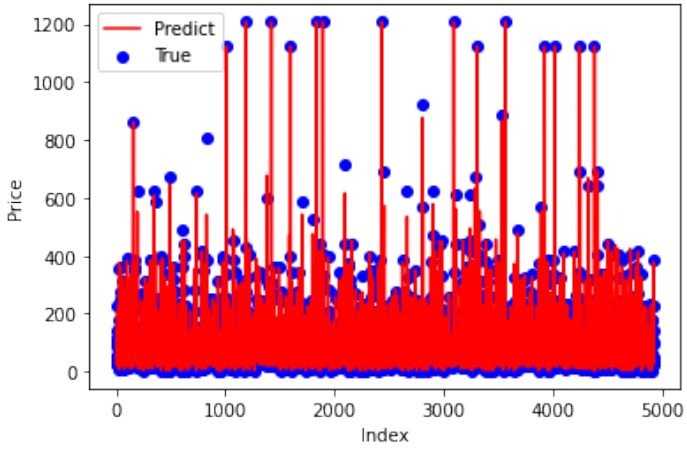
จากผลการทดสอบซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอยโดยใช้ค่าพารามิเตอร์ที่ดีที่สุดที่ได้จากการทำ Grid Search พบว่าซัพพอร์ตเวกเตอร์แมชชีนมีประสิทธิภาพการทำนายดีที่สุดเมื่อกำหนดพารามิเตอร์ C = 1000 และ kernel = 'rbf' โดยมีค่า

ผิดพลาดกำลังสอง ร้อยละความผิดพลาดสัมบูรณ์เฉลี่ย ค่าความแม่นยำ (Accuracy) ของ Training set และ Test set ของการทดสอบดังนี้

Mean squared error: MSE	162.3914396460034
Mean absolute percentage error: MAPE	6.457370539718754
Training set accuracy	0.9932171490653403
Test set accuracy	0.9839647896372504

เมื่อพิจารณาผลการทำนายของซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอยที่ดีที่สุด พบว่าสามารถทำนายราคาสินค้าได้ใกล้เคียงกับราคาจริงในชุดข้อมูลฝึกสอนเป็นอย่างมาก พบปัญหา Overfit เพียงเล็กน้อย และมีประสิทธิภาพการทำนายที่ค่อนข้างแม่นยำในทุกช่วงราคา

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



รูปที่ 10 : ตัวอย่างผลการทำนาย และ แผนภูมิแสดงประสิทธิภาพของซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย

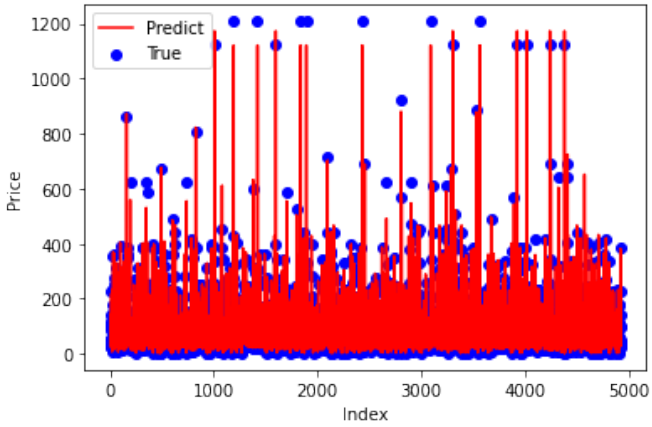
4.3. ผลการพัฒนาข่ายงานประสาทเทียมแบบป้อนหน้า (Feed-forward neural network)

จากผลการทดสอบข่ายงานประสาทเทียมแบบป้อนหน้าโดยใช้ค่าพารามิเตอร์ที่ดีที่สุด พบว่าข่ายงานประสาทเทียมแบบป้อนหน้า มีประสิทธิภาพการทำนายดีที่สุดเมื่อกำหนดพารามิเตอร์ hidden_layer_sizes = (250, 250, 250, 250) และ activation = 'relu'

Mean squared error: MSE	173.18494783761967
Mean absolute percentage error: MAPE	6.84081953098462
Training set accuracy	0.9885445650077587
Test set accuracy	0.9829187423043221

เมื่อพิจารณาผลการทำนายข่ายงานประสาทเทียมแบบป้อนหน้าที่ดีที่สุด พบว่าสามารถทำนายราคาสินค้าได้ใกล้เคียงกับราคาจริงในชุดข้อมูลฝึกสอนในระดับปานกลาง แต่เมื่อวิเคราะห์จากแผนภูมิพบว่า ประสิทธิภาพการทำนายโดยรวมยังไม่ดีนัก และพบปัญหา Overfit โดยเฉพาะอย่างยิ่งข้อมูลในช่วงราคา 1000 ขึ้นไป

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



รูปที่ 11 : ตัวอย่างผลการทำนาย และ แผนภูมิแสดงประสิทธิภาพของข่ายงานประสาทเทียมแบบป้อนหน้า

4.4. เปรียบเทียบ และ สรุปผลตัวแบบทำนาย

	ตัวแบบทำนายเชิงเส้น (Linear regression)	ซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย (Support vector regression)	ข่ายงานประสาทเทียมแบบป้อนหน้า (Feed-forward neural network)
Mean squared error: MSE	341.5515637134753	162.3914396460034	173.18494783761967
Mean absolute percent. error: MAPE	10.297197104084978	6.457370539718754	6.84081953098462
Training set accuracy	0.9744550159807572	0.9932171490653403	0.9885445650077587
Test set accuracy	0.9662737691973751	0.9839647896372504	0.9829187423043221

พบว่าซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย (Support vector regression) มีค่าผิดพลาดกำลังสอง (MSE) ร้อยละความผิดพลาดสัมบูรณ์เฉลี่ย (MAPE) ที่น้อยที่สุด เมื่อเทียบกับตัวแบบทำนายอื่น จึงสรุปได้ว่า เลือกซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย เป็นตัวแบบทำนายที่จะนำมาใช้ทำนายราคาสินค้าสำหรับชุดทดสอบ

5. การทำนายราคาสินค้าในชุดข้อมูลทดสอบ

จากชุดข้อมูลทดสอบที่กำหนดให้ พบว่ามีการเก็บข้อมูล ดังนี้

1. รหัสสินค้า (SKU) = ชนิดสินค้า – ชนิดย่อยสินค้า – เลขลำดับที่
2. ราคาต้นทุนซื้อเฉลี่ยของสัปดาห์ก่อนหน้า (PrevAVGCost)
3. ราคาต้นทุนหลังหักส่วนสูญเสียของสัปดาห์ก่อนหน้า (PrevAssignedCost)
4. ราคาต้นทุนซื้อเฉลี่ยของทั้งสัปดาห์ (AVGCost)
5. ราคาต้นทุนซื้อวันล่าสุด (LatestDateCost)
6. ชนิดของสินค้า (Type)
7. วันที่ตั้งราคา (Date)

5.1. การทำความสะอาดข้อมูล

สำหรับชุดข้อมูลทดสอบ มีลักษณะเดียวกันกับชุดข้อมูลฝึกสอน ประกอบด้วยข้อมูล 2 ชุด คือ testData1.csv และ testData2.csv แต่ละชุดประกอบด้วยข้อมูล 500 และ 511 รายการตามลำดับ โดยพบว่ามีข้อมูล 8 และ 11 รายการตามลำดับ ที่ไม่มีลักษณะสำคัญ กล่าวคือ มีอย่างน้อย 1 ลักษณะที่หายไป จึงได้ทำการตัดรายการดังกล่าวทิ้งไป เหลือข้อมูลทั้งสิ้น 492 และ 500 รายการตามลำดับ

	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

รูปที่ 12 : ตารางแสดงตัวอย่างข้อมูลจากชุดข้อมูลทดสอบที่ขาดหายไป

หลังจากทำการทำความสะอาดข้อมูลด้วยวิธีการเดียวกันที่ใช้กับชุดข้อมูลฝึกสอน ทำให้ได้ชุดข้อมูลทดสอบที่มีตัวอย่างและลักษณะดังต่อไปนี้

	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

489 rows × 12 columns

รูปที่ 13 : ตัวอย่างชุดข้อมูลทดสอบที่ได้หลังจากการทำความสะอาดข้อมูล

```
<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
```

```
<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
```

รูปที่ 14 : ลักษณะของข้อมูลชุดทดสอบที่ 1 และ 2

5.2. การทำนายชุดข้อมูลทดสอบ

ทำนายราคาสินค้าของชุดข้อมูลทดสอบทั้ง 2 ชุด โดยใช้ซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย เก็บผลลัพธ์ที่ได้จากการทำนายในคอลัมน์ราคาตั้งของสัปดาห์นี้ (GenPrice) จากนั้นจึงทำการบันทึกข้อมูล และนำออก (Export) ให้อยู่ในรูปแบบไฟล์ .csv เพื่อใช้เป็นไฟล์คำตอบ

	SKU	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	date	GenPrice
0	A-A-00001	22.646465	30.0	24.646465	23.767677	A	2020-01-20	41.942576
1	A-B-00001	51.101010	56.0	49.565657	49.838384	A	2020-01-20	77.933715
2	A-B-00002	40.010101	41.0	17.323232	38.727273	A	2020-01-20	55.737651
3	A-C-00001	77.484848	81.0	78.545455	80.050505	B	2020-01-20	113.008982
4	A-C-00019	77.484848	81.0	78.545455	80.050505	B	2020-01-20	113.008982
...	...	...	...	...	...	...	...	...
484	E-E-00130	0.000000	60.0	0.000000	0.000000	G	2020-01-20	84.148757
485	E-A-00412	0.000000	90.0	0.000000	0.000000	G	2020-01-20	123.455584
486	E-A-00402	25.131313	25.0	0.000000	0.000000	G	2020-01-20	37.754660
487	E-E-00010	181.818182	201.0	156.565657	181.818182	G	2020-01-20	237.105699
488	E-A-00082	0.000000	70.0	0.000000	0.000000	G	2020-01-20	97.353564

489 rows × 8 columns

	SKU	PrevAVGCost	PrevAssignedCost	AVGCost	LatestDateCost	Type	date	GenPrice
0	A-A-00001	25.767677	27.0	14.313131	14.313131	A	2020-03-27	33.563156
1	A-B-00001	43.424242	45.0	43.313131	44.131313	A	2020-03-27	66.281486
2	A-B-00002	40.656566	42.0	39.838384	40.606061	A	2020-03-27	61.238372
3	A-C-00001	84.777778	87.0	86.060606	87.656566	B	2020-03-27	120.625811
4	A-C-00019	84.777778	87.0	86.060606	87.656566	B	2020-03-27	120.625811
...	...	...	...	...	...	...	...	...
490	E-A-00082	0.000000	70.0	0.000000	0.000000	G	2020-03-27	97.353564
491	A-E-00191	0.000000	60.0	0.000000	0.000000	C	2020-03-27	91.215495
492	A-D-00166	0.000000	99.0	0.000000	0.000000	C	2020-03-27	140.197757
493	A-A-00199	0.000000	59.0	0.000000	0.000000	C	2020-03-27	89.913528
494	A-A-00200	0.000000	75.0	0.000000	0.000000	C	2020-03-27	110.522950

495 rows × 8 columns

รูปที่ 15 : ตัวอย่างข้อมูลภายในไฟล์คำตอบ

6. บทสรุป

โครงการตั้งราคาสินค้า ผู้จัดทำได้ทดลองสร้างโมเดลการเรียนรู้ของเครื่อง (Machine learning) ทั้งหมด 3 ประเภท เพื่อใช้ทำนายการกำหนดราคาสินค้า ได้แก่ ตัวแบบทำนายเชิงเส้น (Linear regression) ซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอย (Support vector regression) และ ข่ายงานประสาทเทียมแบบป้อนหน้า (Feed-forward neural network)

จากผลการพัฒนาและวัดผลตัวแบบทำนายของโมเดลแต่ละประเภท พบว่าผลการทดสอบพบว่าซัพพอร์ตเวกเตอร์แมชชีนแบบถดถอยให้ค่าผิดพลาดกำลังสอง (MSE) และร้อยละความผิดพลาดสัมบูรณ์เฉลี่ย (MAPE) ที่น้อยที่สุด จึงเลือกใช้ตัวแบบทำนายนี้ในการทำนายราคาสินค้าในชุดข้อมูลทดสอบ ทั้งนี้ผลการทดลองอาจแตกต่างออกไป หากทดลองด้วยตัวแบบทำนายประเภทอื่น ทำความสะอาดข้อมูลด้วยวิธีที่แตกต่างออกไป รวมไปถึงหากผู้จัดทำมีความรู้ด้านการตลาดก็จะสามารถวิเคราะห์ผลการทำนายราคาสินค้าได้ดียิ่งขึ้น ทั้งนี้หากรายงานฉบับนี้มีข้อผิดพลาดประการใด ผู้จัดทำขออภัยมา ณ ที่นี้ด้วย

7. ภาคผนวก

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/1eliQ05XPm-mbUQllb9UDg8VqKah4xqXV
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