



Project_DWDM

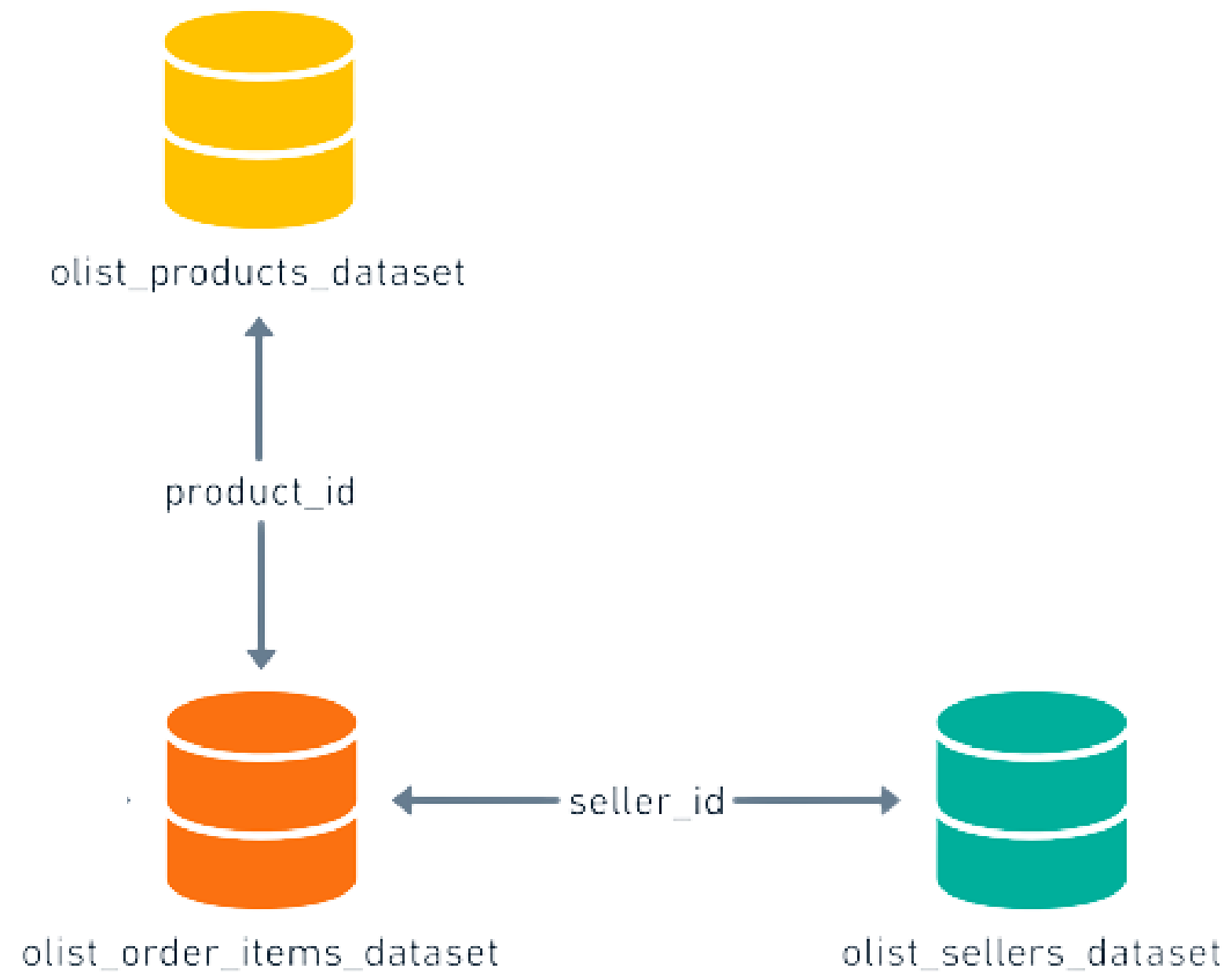
Olist Data Case

**(ข้อมูลการซื้อขายสินค้าผ่าน
Platform Olist ของประเทศบราซิล)**

BY แมวส้ม

แนะนำ Data

เว็บที่มาของข้อมูล : <https://www.kaggle.com/olistbr/brazilian-ecommerce>



ตาราง sellers



ตาราง sellers

เป็นตารางที่เกี่ยวกับผู้ขายสินค้า ที่ขายสินค้าใน Olist

```
sellers = pd.read_csv(os.path.join(path, 'olist_sellers_dataset.csv'))  
sellers
```

	seller_id	seller_zip_code_prefix	seller_city	seller_state
0	3442f8959a84dea7ee197c632cb2df15	13023	campinas	SP
1	d1b65fc7debc3361ea86b5f14c68d2e2	13844	mogi guacu	SP
2	ce3ad9de960102d0677a81f5d0bb7b2d	20031	rio de janeiro	RJ
3	c0f3eea2e14555b6faeea3dd58c1b1c3	4195	sao paulo	SP
4	51a04a8a6bdc23deccc82b0b80742cf	12914	braganca paulista	SP
...
3090	98dddbc4601dd4443ca174359b237166	87111	sarandi	PR
3091	f8201cab383e484733266d1906e2fdfa	88137	palhoca	SC
3092	74871d19219c7d518d0090283e03c137	4650	sao paulo	SP
3093	e603cf3fec55f8697c9059638d6c8eb5	96080	pelotas	RS
3094	9e25199f6ef7e7c347120ff175652c3b	12051	taubate	SP

3095 rows x 4 columns

seller_id คือ PK ของตาราง

seller_zip_code_prefix คือ FK (รหัสไปรษณีย์ 5 หลักแรกของผู้ขาย)

seller_city คือ ชื่อเมืองผู้ขาย

seller_state คือ ชื่อรัฐผู้ขาย

ตาราง Order_items

ตาราง order_items

ข้อมูลเกี่ยวกับสินค้าที่ซื้อในแต่ละคำสั่งซื้อ

```
order_items = pd.read_csv(os.path.join(path, 'olist_order_items_dataset.csv'))
order_items
```

	order_id	order_item_id	product_id	seller_id	shipping_limit_date
0	00010242fe8c5a6d1ba2dd792cb16214	1	4244733e06e7ecb4970a6e2683c13e61	48436dade18ac8b2bce089ec2a041202	2017-09-19 09:45:35
1	00018f77f2f0320c557190d7a144bdd3	1	e5f2d52b802189ee658865ca93d83a8f	dd7ddc04e1b6c2c614352b383efe2d36	2017-05-03 11:05:13
2	000229ec398224ef6ca0657da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b51032eddd242adc84c38acab88f23d	2018-01-18 14:48:30
3	00024acbcd0a6daa1e931b038114c75	1	7634da152a4610f1595efa32f14722fc	9d7a1d34a5052409006425275ba1c2b4	2018-08-15 10:10:18
4	00042b26cf59d7ce69dfabb4e55b4fd9	1	ac6c3623068f30de03045865e4e10089	df560393f3a51e74553ab94004ba5c87	2017-02-13 13:57:51
...
112645	fffc94f6ce00a00581880bf54a75a037	1	4aa6014eceb682077f9dc4bffe5bc05b0	b8bc237ba3788b23da09c0f1f3a3288c	2018-05-02 04:11:01
112646	ffcd46ef2263f404302a634eb57f7eb	1	32e07fd915822b0765e448c4dd74c828	f3c38ab652836d21de61fb8314b69182	2018-07-20 04:31:48
112647	fffce4705a9662cd70adb13d4a31832d	1	72a30483855e2eafc67aee5dc2560482	c3cfdc648177fdbbbb35635a37472c53	2017-10-30 17:14:25
112648	fffe18544ffabc95dfada21779c9644f	1	9c422a519119dcad7575db5af1ba540e	2b3e4a2a3ea8e01938cabda2a3e5cc79	2017-08-21 00:04:32
112649	fffe41c64501cc87c801fd61db3f6244	1	350688d9dc1e75ff97be326363655e01	f7ccf836d21b2fb1de37564105216cc1	2018-06-12 17:10:13

112650 rows × 7 columns

order_id คือ PK

order_item_id คือ หมายเลขลำดับระบุจำนวนรายการที่รวมอยู่ในลำดับเดียวกัน

product_id คือ FK (เป็น PK ของตาราง products)

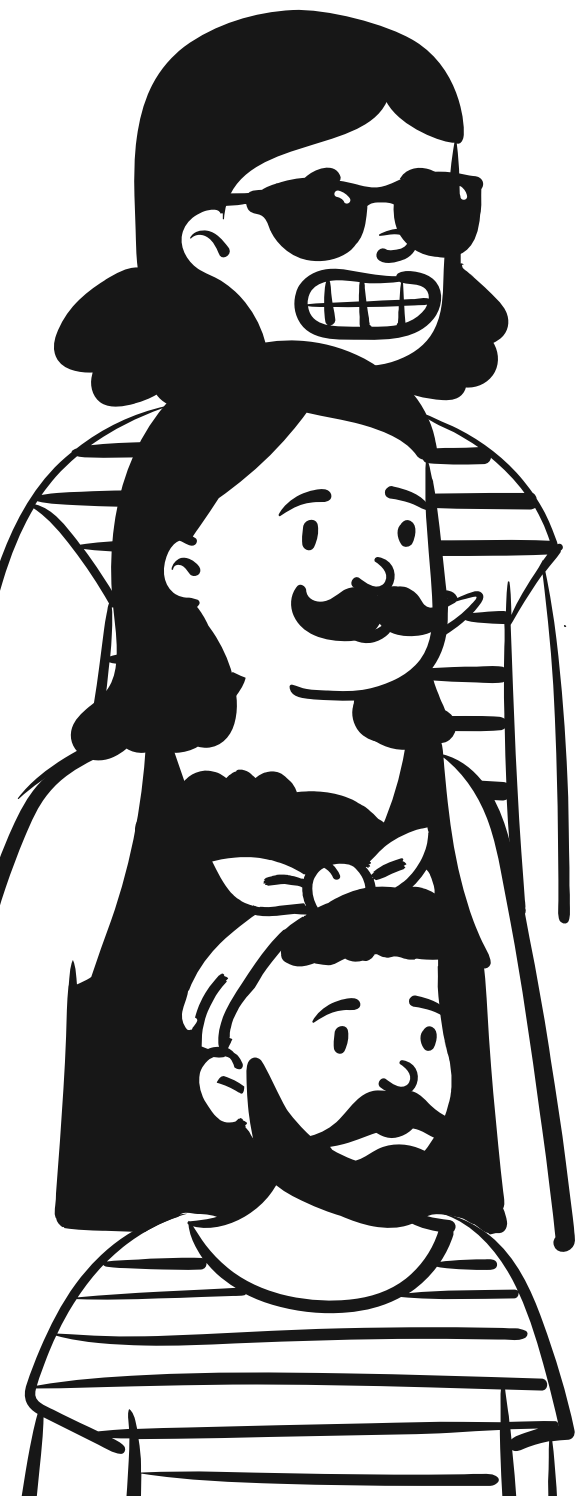
seller_id คือ FK (เป็น PK ของตาราง sellers)

shipping_limit_date คือ กำหนดการ deadline วันส่งสินค้าของผู้ขายให้กับบริษัทขนส่ง

price คือ ราคาสินค้า

freight_value คือ ค่าจัดส่ง

ตาราง products



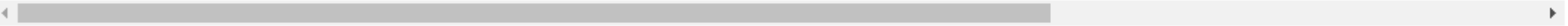
ตาราง products

ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ขายใน Olist

```
products = pd.read_csv(os.path.join(path, 'olist_products_dataset.csv'))
products
```

	product_id	product_category_name	product_name_lenght	product_description_lenght	product_photos_qty	prod
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	40.0	287.0	1.0	225.0
1	3aa071139cb16b67ca9e5dea641aaa2f	artes	44.0	276.0	1.0	1000
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer	46.0	250.0	1.0	154.0
3	cef67bcfe19066a932b7673e239eb23d	bebes	27.0	261.0	1.0	371.0
4	9dc1a7de274444849c219cff195d0b71	utilidades_domesticas	37.0	402.0	4.0	625.0
...
32946	a0b7d5a992ccda646f2d34e418fff5a0	moveis_decoracao	45.0	67.0	2.0	1230
32947	bf4538d88321d0fd4412a93c974510e6	construcao_ferramentas_iluminacao	41.0	971.0	1.0	1700
32948	9a7c6041fa9592d9d9ef6cfe62a71f8c	cama_mesa_banho	50.0	799.0	1.0	1400
32949	83808703fc0706a22e264b9d75f04a2e	informatica_acessorios	60.0	156.0	2.0	700.0
32950	106392145fca363410d287a815be6de4	cama_mesa_banho	58.0	309.0	1.0	2083

32951 rows x 9 columns



- product_id คือ PK
- product_category_name คือ ชื่อหมวดหมู่สินค้า (ภาษาโปรตุเกส)
- product_name_lenght คือ จำนวนอักษรของชื่อสินค้า
- product_description_lenght คือ จำนวนอักษรของคำอธิบายสินค้า
- product_photos_qty คือ จำนวนรูปภาพที่เผยแพร่ของสินค้า
- product_weight_g คือ น้ำหนักของสินค้า (g)
- product_length_cm คือ ความยาวของสินค้า (cm)
- product_height_cm คือ ความสูงของสินค้า (cm)
- product_width_cm คือ ความกว้างของสินค้า (cm)

Preprocessing

ต่อตารางทั้ง 3 ตารางเข้าด้วยกัน จากนั้นทำการตรวจสอบค่า **Missing**



```
### ต่อตาราง order_items, sellers, products
```

```
merged_ori_sellers_pro = merged_ori_sellers.merge(products_dropped,how='left',left_on='product_id',right_on='product_id')
merged_ori_sellers_pro
```

	order_id	order_item_id	product_id	seller_id	shipping_limit_date
0	00010242fe8c5a6d1ba2dd792cb16214	1	4244733e06e7ecb4970a6e2683c13e61	48436dade18ac8b2bce089ec2a041202	2017-09-19 09:45:35
1	00018f77f2f0320c557190d7a144bdd3	1	e5f2d52b802189ee658865ca93d83a8f	dd7ddc04e1b6c2c614352b383efe2d36	2017-05-03 11:05:13
2	000229ec398224ef6ca0657da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b51032eddd242adc84c38acab88f23d	2018-01-18 14:48:30
3	00024acbcd0a6daa1e931b038114c75	1	7634da152a4610f1595efa32f14722fc	9d7a1d34a5052409006425275ba1c2b4	2018-08-15 10:10:18
4	00042b26cf59d7ce69dfabb4e55b4fd9	1	ac6c3623068f30de03045865e4e10089	df560393f3a51e74553ab94004ba5c87	2017-02-13 13:57:51
...
112645	fffc94f6ce00a00581880bf54a75a037	1	4aa6014eceb682077f9dc4bffe5bc05b0	b8bc237ba3788b23da09c0f1f3a3288c	2018-05-02 04:11:01
112646	ffcd46ef2263f404302a634eb57f7eb	1	32e07fd915822b0765e448c4dd74c828	f3c38ab652836d21de61fb8314b69182	2018-07-20 04:31:48
112647	ffce4705a9662cd70adb13d4a31832d	1	72a30483855e2eafc67aee5dc2560482	c3cfdc648177fdbbbb35635a37472c53	2017-10-30 17:14:25
112648	fffe18544ffabc95dfada21779c9644f	1	9c422a519119dcad7575db5af1ba540e	2b3e4a2a3ea8e01938cabda2a3e5cc79	2017-08-21 00:04:32
112649	fffe41c64501cc87c801fd61db3f6244	1	350688d9dc1e75ff97be326363655e01	f7ccf836d21b2fb1de37564105216cc1	2018-06-12 17:10:13

112650 rows × 18 columns

```
### ดูว่าแต่ละ column มีค่า missing มั้ย
```

```
merged_ori_sellers_pro.isnull().any()
```

order_id	False
order_item_id	False
product_id	False
seller_id	False
shipping_limit_date	False
price	False
freight_value	False
seller_zip_code_prefix	False
seller_city	False
seller_state	False
product_category_name	True
product_name_lenght	True
product_description_lenght	True
product_photos_qty	True
product_weight_g	True
product_length_cm	True
product_height_cm	True
product_width_cm	True

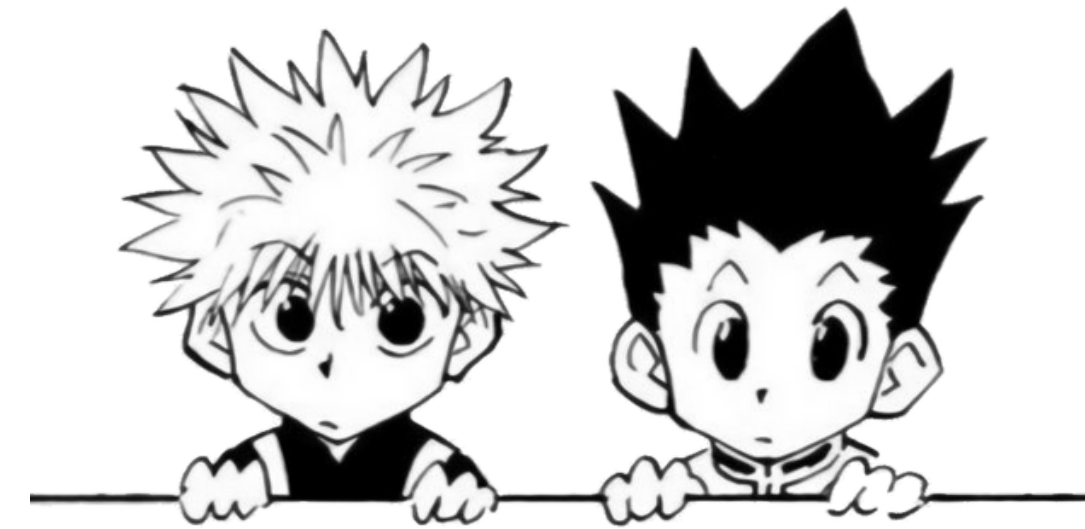
Preprocessing

ทำการตรวจสอบว่าถ้าทำการ ลบ missing ด้วยการ `.dropna()` จะเสียข้อมูลที่เปอร์เซ็นต์

```
#เสียข้อมูลเยอะแค่ไหนถ้า .dropna()  
loss_data(merged_ori_sellers_pro)
```

```
size before drop = 112650  
size after drop = 111046  
we loss 1.4238792720816689 % of data
```

จะเห็นได้ว่าเสียข้อมูลที่เปอร์เซ็นต์ ไป 1.42 %
จึงทำการ `.dropna()` เพื่อลบ missing ของข้อมูล



ดูว่าแต่ละ column มีค่า missing มั้ย

```
data_dropped.isnull().any()
```

order_id	False
order_item_id	False
product_id	False
seller_id	False
shipping_limit_date	False
price	False
freight_value	False
seller_zip_code_prefix	False
seller_city	False
seller_state	False
product_category_name	False
product_name_lenght	False
product_description_lenght	False
product_photos_qty	False
product_weight_g	False
product_length_cm	False
product_height_cm	False
product_width_cm	False
dtype:	bool

จากนั้นทำการเช็คดูว่าข้อมูลนั้นยังมีค่า missing อยู่หรือไม่

Association Rules

7

```
#จัดกลุ่ม product_id ด้วย order_id
transactions = data_dropped.groupby('order_id')['product_id'].apply(list)
transactions
```

```
order_id
00010242fe8c5a6d1ba2dd792cb16214    [4244733e06e7ecb4970a6e2683c13e61]
00018f77f2f0320c557190d7a144bdd3    [e5f2d52b802189ee658865ca93d83a8f]
000229ec398224ef6ca0657da4fc703e    [c777355d18b72b67abbeef9df44fd0fd]
00024acbcdf0a6daa1e931b038114c75    [7634da152a4610f1595efa32f14722fc]
00042b26cf59d7ce69dfabb4e55b4fd9    [ac6c3623068f30de03045865e4e10089]
...
fffc94f6ce00a00581880bf54a75a037    [4aa6014eceb682077f9dc4bffeabc05b0]
fffc46ef2263f404302a634eb57f7eb    [32e07fd915822b0765e448c4dd74c828]
fffce4705a9662cd70adb13d4a31832d    [72a30483855e2eafc67aee5dc2560482]
fffe18544ffabc95dfada21779c9644f    [9c422a519119dcad7575db5af1ba540e]
fffe41c64501cc87c801fd61db3f6244    [350688d9dc1e75ff97be326363655e01]
Name: product_id, Length: 97276, dtype: object
```

```
#หา association rules โดยกำหนด min_support เท่ากับ 0.5 %
frequentItemsets = list(apriori(transactions,min_support = 0.005))
frequentItemsets
```

```
[]
```

สรุปการทำ Association Rules ด้วยการจัดกลุ่ม product_id ด้วย order_id
สรุปได้ว่า ไม่มีอะไรน่าสนใจ เนื่องจากไม่มี product_id ที่มี min_support เกิน 0.5% เลย



Classification

ทำนาย `product_category_name`(ชื่อหมวดหมู่สินค้า) ด้วย
`price`(ราคาสินค้า)
`freight_value`(ค่าจัดส่ง)
`product_weight_g`(น้ำหนักของสินค้า (g))
`product_length_cm`(ความยาวของสินค้า (cm))
`product_height_cm`(ความสูงของสินค้า (cm))
`product_width_cm`(ความกว้างของสินค้า (cm))



ประโยชน์ของการทำนาย `product_category_name`(ชื่อหมวดหมู่สินค้า) นั้น มีประโยชน์
กับบริษัทขนส่งสินค้า เนื่องจากจะช่วยบอกถึงประเภทสินค้าที่ลูกค้าต้องการส่งโดยไม่ต้อง
เปิดพัสดุดูสิ่งของข้างใน

Split Data

ทำการแบ่งข้อมูลออกเป็นสองส่วน และกำหนดตัวแปรให้กับ Data

```
#กำหนด X และ y
X = data_dropped[['product_height_cm', 'product_length_cm', 'product_weight_g', 'product_width_cm', 'freight_value', 'price']]
y = data_dropped['product_category_name']
```

```
from sklearn.model_selection import train_test_split
```

Train-Test

```
#แบ่งข้อมูลออกเป็น 2 ส่วน ด้วยการสุ่ม
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=13)
```

X.shape

(111046, 6)

X_train.shape

(55523, 6)

X_test.shape

(55523, 6)

Train - Validation

```
#แบ่งข้อมูลจาก X_train, y_train ออกเป็น 2 ส่วน ด้วยการสุ่ม
X_train2, X_val, y_train2, y_val = train_test_split(X_train, y_train, test_size=0.3, random_state=13)
```



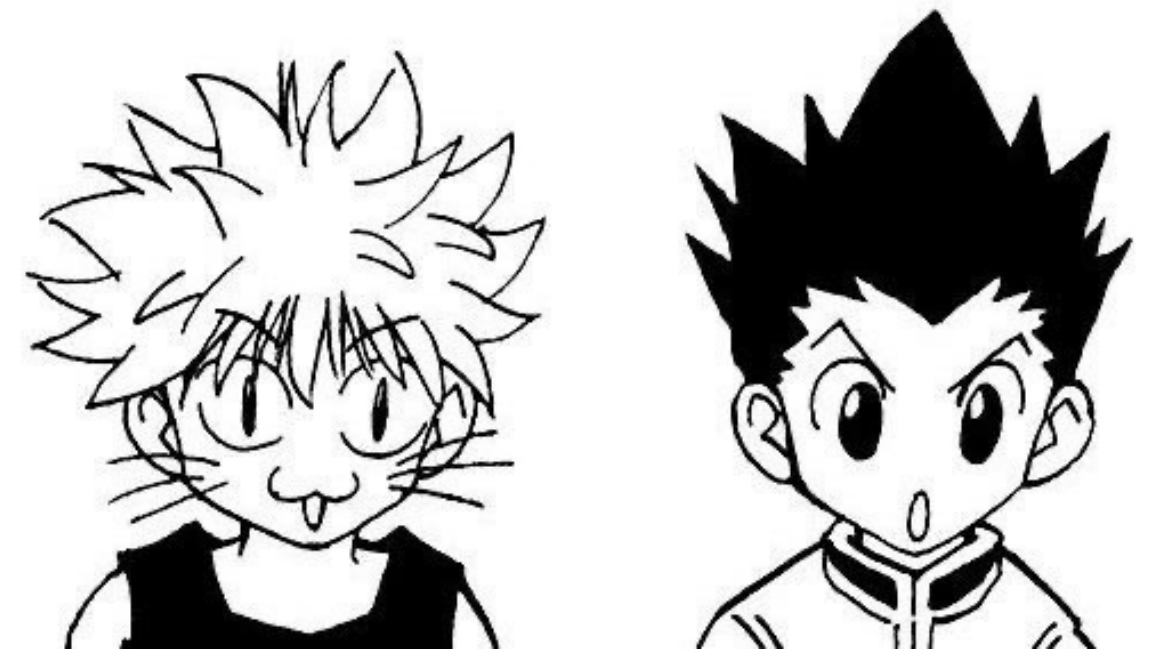
Decision Tree

ทำการ train model เพื่อหา model ที่ดีที่สุด

ใช้ค่า default

```
#define
mytree = DecisionTreeClassifier()
#train
mytree.fit(X_train2,y_train2)
#test
mytree_result = mytree.predict(X_val)
#accuracy_score
acc_tree1 = accuracy_score(y_val, mytree_result)
acc_tree1
```

0.7640031218106502



ใช้เกณฑ์ criterion='entropy', splitter='random', random_state=13

```
#define
mytree2 = DecisionTreeClassifier(criterion='entropy',splitter='random',random
_state=13)
#train
mytree2.fit(X_train2,y_train2)
#test
mytree_result2 = mytree2.predict(X_val)
#accuracy_score
acc_tree2 = accuracy_score(y_val, mytree_result2)
acc_tree2
```

0.7715074743351144

Decision Tree

ใช้เกณฑ์ splitter='random',random_state=13

```
#define
mytree3 = DecisionTreeClassifier(splitter='random',random_state=13)
#train
mytree3.fit(X_train2,y_train2)
#test
mytree_result3 = mytree3.predict(X_val)
#accuracy_score
acc_tree3 = accuracy_score(y_val, mytree_result3)
acc_tree3
```

0.77576994656901

ใช้เกณฑ์ min_samples_leaf=2,splitter='random',random_state=13

```
#define
mytree4 = DecisionTreeClassifier(min_samples_leaf=2,splitter='random',random_state=13)
#train
mytree4.fit(X_train2,y_train2)
#test
mytree_result4 = mytree4.predict(X_val)
#accuracy_score
acc_tree4 = accuracy_score(y_val, mytree_result4)
acc_tree4
```

0.7161553701146665



K-NN

ทำการ train modal เพื่อหา model ที่ดีที่สุด

ใช้เกณฑ์ n_neighbors=3, weights='uniform'

```
# Define
neigh1 = KNeighborsClassifier(n_neighbors=3, weights='uniform') #ถามเพื่อน
บ้านที่ใกล้สุด 3 คน และเชื่อทุกคนเท่ากัน
# Train
neigh1.fit(X_train2,y_train2)
# Test
knn1_result = neigh1.predict(X_val)
#accuracy_score
acc_KNN1 = accuracy_score(y_val, knn1_result)
acc_KNN1
```

0.6378099297592603

ใช้เกณฑ์ n_neighbors=25, weights='distance'

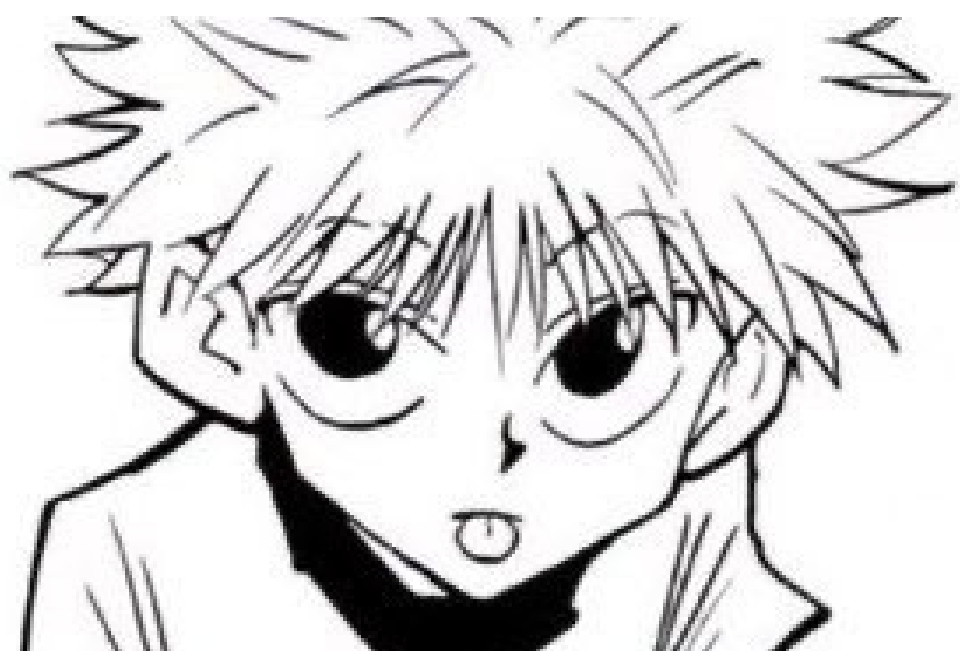
```
# Define
neigh2 = KNeighborsClassifier(n_neighbors=25, weights='distance') #ถามเพื่อ
บ้านที่ใกล้สุด 25 คน และเชื่อคนที่อยู่ใกล้มากกว่าไกล
# Train
neigh2.fit(X_train2,y_train2)
# Test
knn2_result = neigh2.predict(X_val)
#accuracy_score
acc_KNN2 = accuracy_score(y_val, knn2_result)
acc_KNN2
```

0.6530587740889716

ใช้เกณฑ์ n_neighbors=1

```
# Define
neigh3 = KNeighborsClassifier(n_neighbors=1) #เชื่อเพื่อนบ้านที่ใกล้ที่สุดคนเดียว
# Train
neigh3.fit(X_train2,y_train2)
# Test
knn3_result = neigh3.predict(X_val)
#accuracy_score
acc_KNN3 = accuracy_score(y_val, knn3_result)
acc_KNN3
```

0.7111124452182266



Neural Network

ทำการ train model เพื่อหา model ที่ดีที่สุด

ann1

```
#define
perceptron1 = MLPClassifier(random_state=13, max_iter=200, hidden_layer_
sizes=1,learning_rate_init=0.005)
#train
perceptron1.fit(X_train2,y_train2)
#test
ann1_result = perceptron1.predict(X_val)
#accuracy_score
acc_NN1 = accuracy_score(y_val, ann1_result)
acc_NN1
```

0.16545596445938646

ann2

```
#define
perceptron2 = MLPClassifier(random_state=13, max_iter=200, hidden_layer_
sizes=10,learning_rate_init=0.005)
#Train
perceptron2.fit(X_train2,y_train2)
#Test
ann2_result = perceptron2.predict(X_val)
#accuracy_score
acc_NN2 = accuracy_score(y_val, ann2_result)
acc_NN2
```

0.1830461667767305

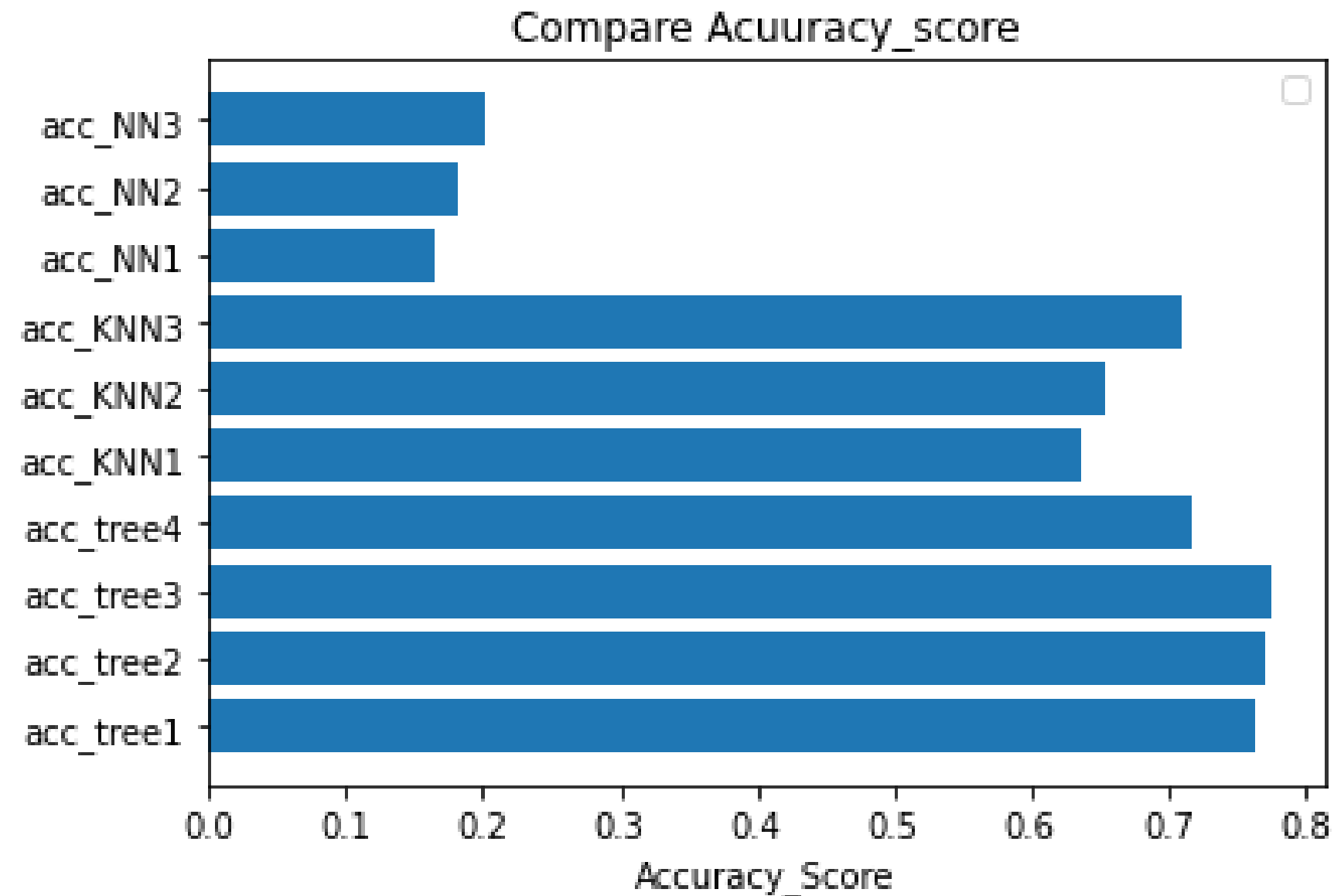
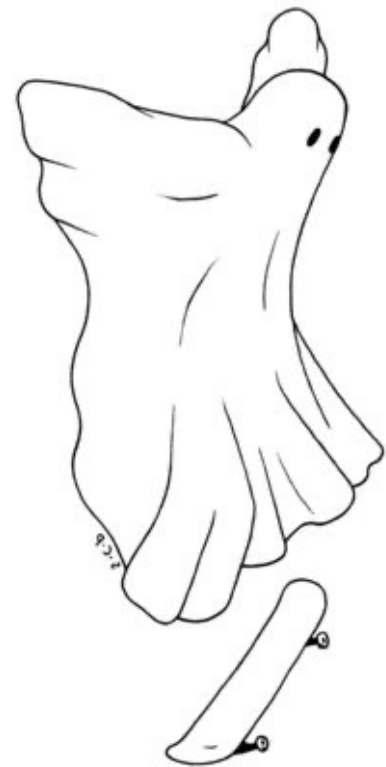
ann3

```
#define
perceptron3 = MLPClassifier(random_state=13, max_iter=20000, hidden_laye
r_sizes=10,learning_rate_init=0.001)
#Train
perceptron3.fit(X_train2,y_train2)
#Test
ann3_result = perceptron3.predict(X_val)
#accuracy_score
acc_NN3 = accuracy_score(y_val, ann3_result)
acc_NN3
```

0.2008164735546617



พล็อตกราฟเปรียบเทียบความแม่นยำของการวัดผล การทำ classification ด้วย accuracy_score



สรุป model model ที่เหมาะสมที่สุดกับข้อมูลชุดนี้ คือ
Decision Tree(mytree3 = DecisionTreeClassifier(splitter='random',random_state=13))
ซึ่งมีค่า accuracy = **77.58%**

Retrain & Evaluate



```
#import  
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

```
#Define  
mytree3_final = DecisionTreeClassifier(splitter='random', random_state=13)  
#Train  
mytree3_final.fit(X_train, y_train)  
#Predict  
mytree3_final_result = mytree3_final.predict(X_test)
```

```
#Evaluate  
acc_decision_tree = accuracy_score(y_test, mytree3_final_result)  
acc_decision_tree
```

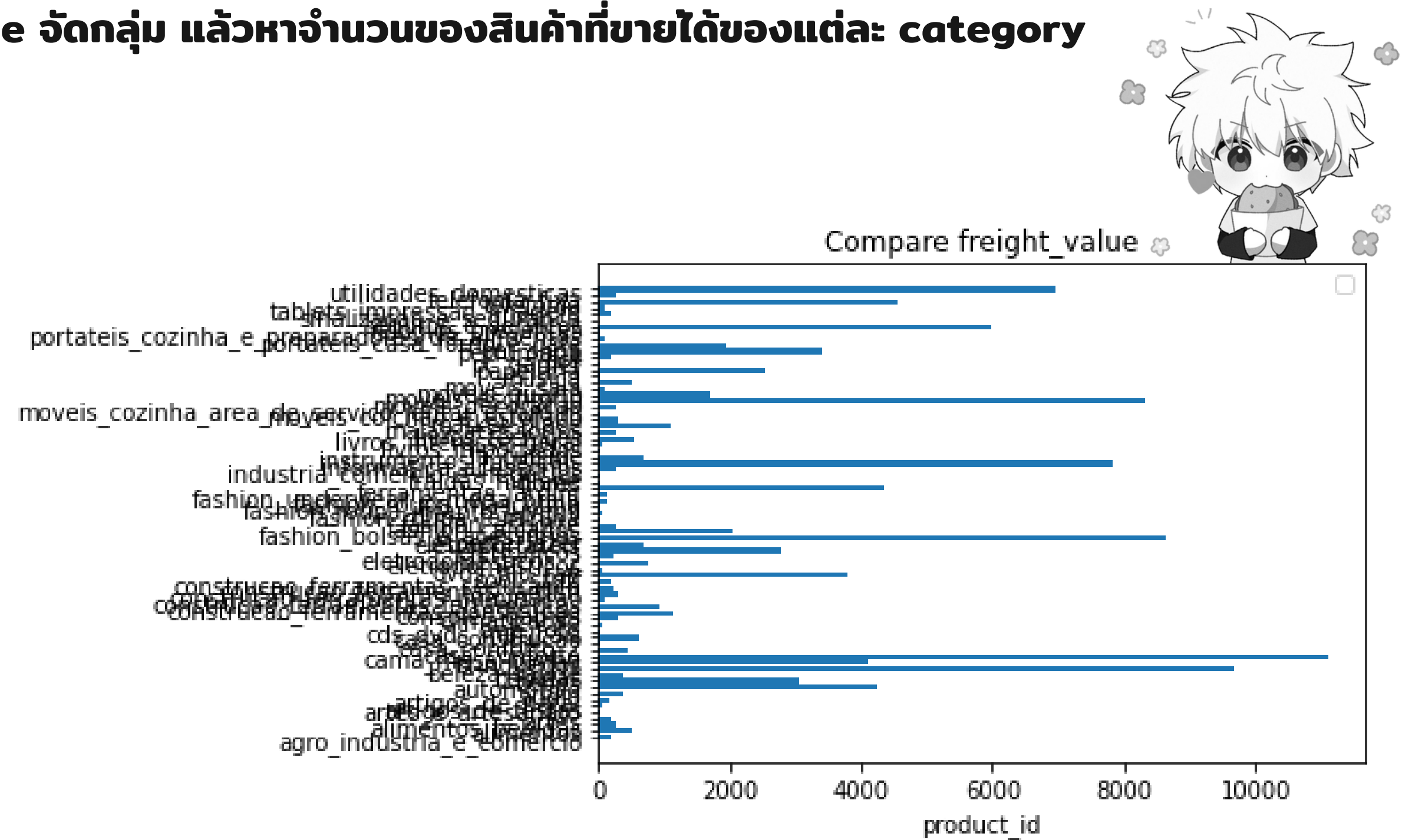
0.8109252021684707



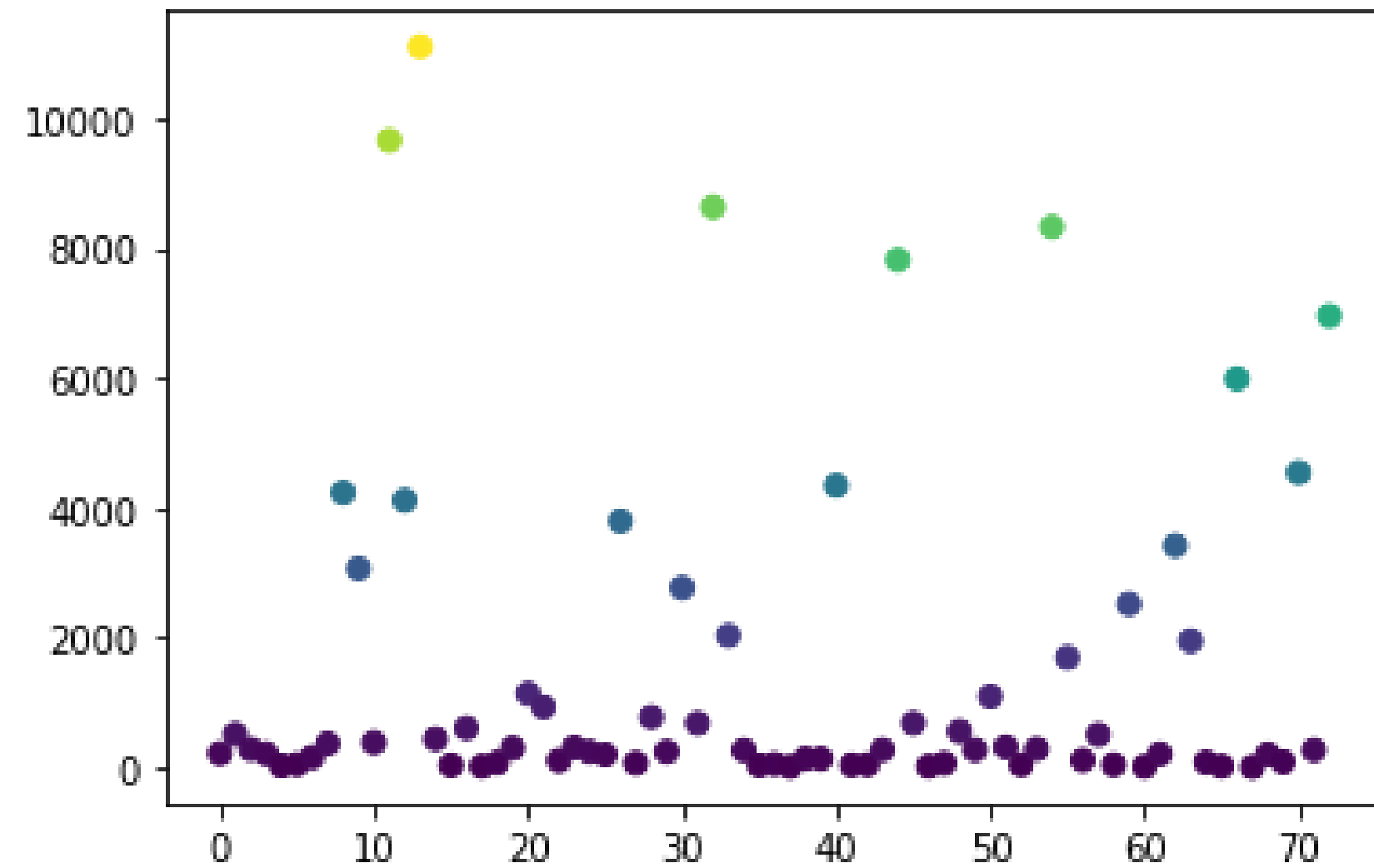
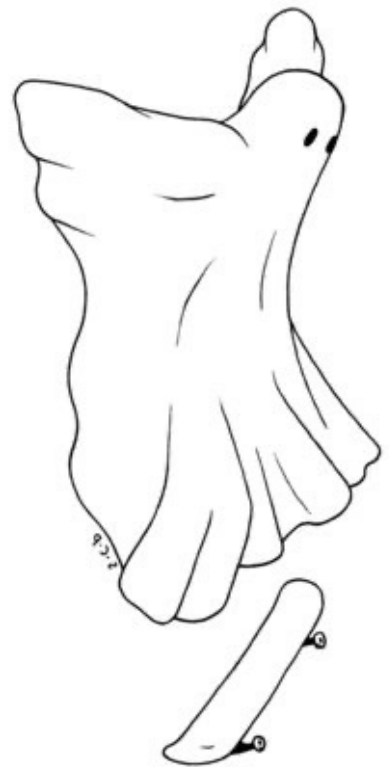
Visualization

ใช้ข้อมูล `product_category_name` จัดกลุ่ม แล้วหาจำนวนของสินค้าที่ขายได้ของแต่ละ `category`

	product_category_name	product_id
0	agro_industria_e_comercio	212
1	alimentos	510
2	alimentos_bebidas	278
3	artes	209
4	artes_e_artesanato	24
...
68	sinalizacao_e_seguranca	199
69	tablets_impressao_imagem	83
70	telefonica	4545
71	telefonica_fixa	264
72	utilidades_domesticas	6964



แสดงจำนวนของสินค้าที่ขายได้ของแต่ละ category



จากกราฟ จะเห็นว่า category ส่วนใหญ่ขายสินค้าได้ไม่เกิน 2000 ชิ้น

เปรียบเทียบ ราคา ,ค่าส่ง ,ความสูง ,ความยาว ,น้ำหนัก และ ความกว้าง เฉลี่ย ของแต่ละ category

หาค่าเฉลี่ยของ ราคา,ค่าส่ง,ความสูง,ความยาว,น้ำหนัก และ ความกว้าง



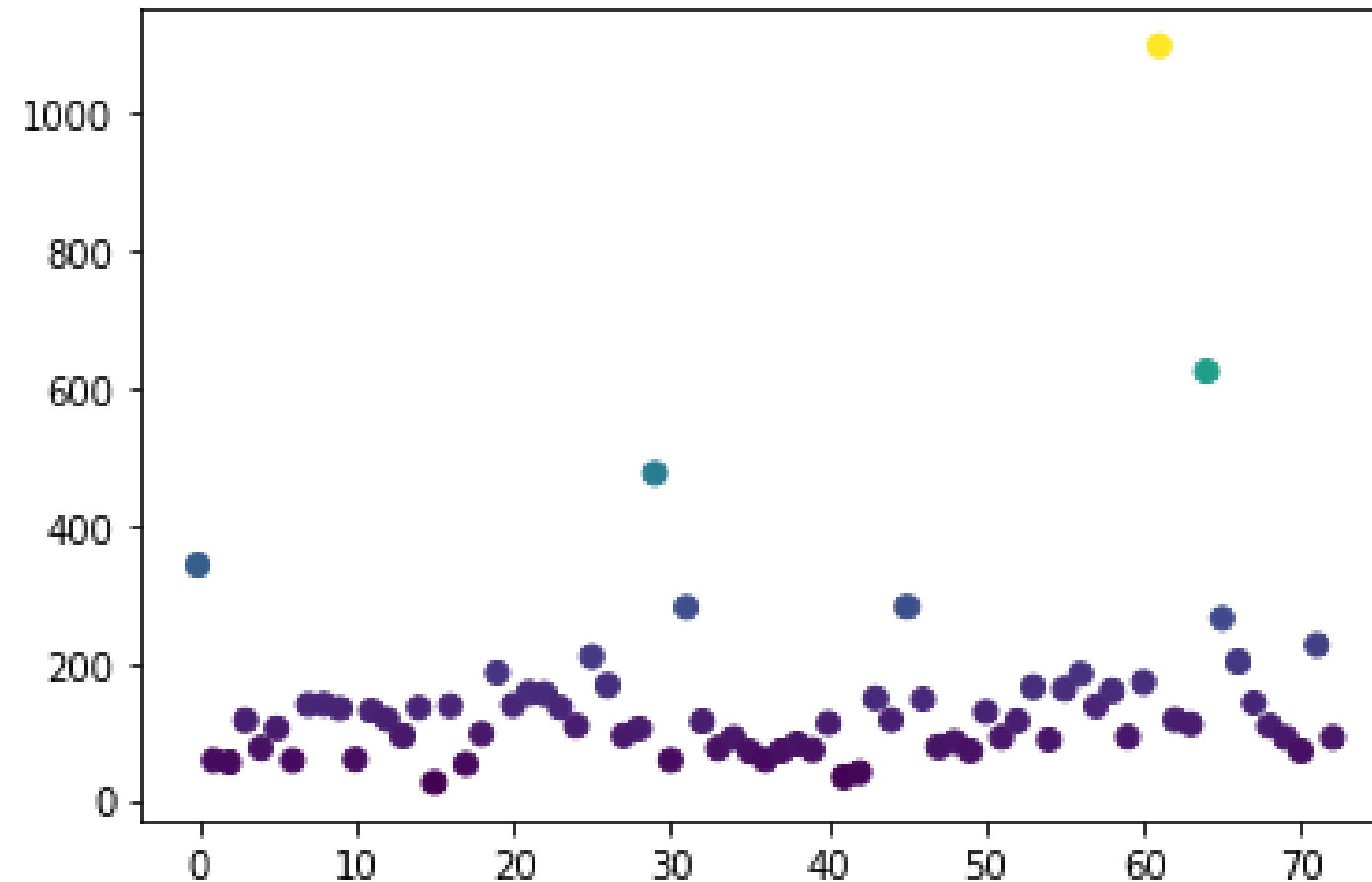
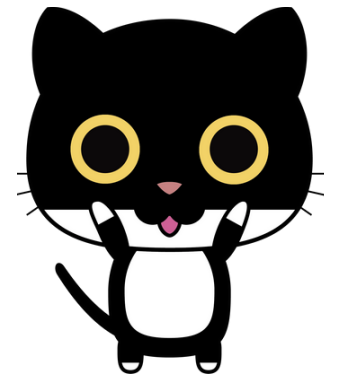
#หาค่าเฉลี่ยของ ราคา,ค่าส่ง,ความสูง,ความยาว,น้ำหนัก และ ความกว้าง

```
data_mean = data_dropped.groupby('product_category_name')[['price', 'freight_value', 'product_height_cm', 'product_weight_g', 'product_width_cm']].mean().reset_index()
data_mean
```

	product_category_name	price	freight_value	product_height_cm	product_length_cm	product_weight_g	product_width_cm
0	agro_industria_e_comercio	342.124858	27.564151	23.787736	28.938679	3625.113208	20.867925
1	alimentos	57.634137	14.256922	13.666667	19.327451	658.115686	15.631373
2	alimentos_bebidas	54.602446	16.215791	15.924460	21.751799	1104.654676	16.676259
3	artes	115.802105	19.354880	8.535885	49.043062	1557.545455	36.531100
4	artes_e_artesanato	75.583750	15.422083	10.916667	30.916667	1369.875000	21.541667
...
68	sinalizacao_e_seguranca	108.086583	32.702613	28.693467	34.296482	3886.723618	21.150754
69	tablets_impressao_imagem	90.703735	14.773494	9.626506	25.506024	307.096386	23.277108
70	telefonica	71.213978	15.669041	6.354675	18.389439	261.622222	13.399120
71	telefonica_fixa	225.693182	17.567462	10.428030	19.867424	565.344697	16.018939
72	utilidades_domesticas	90.788148	20.986374	24.209219	32.128518	3214.587881	24.931792

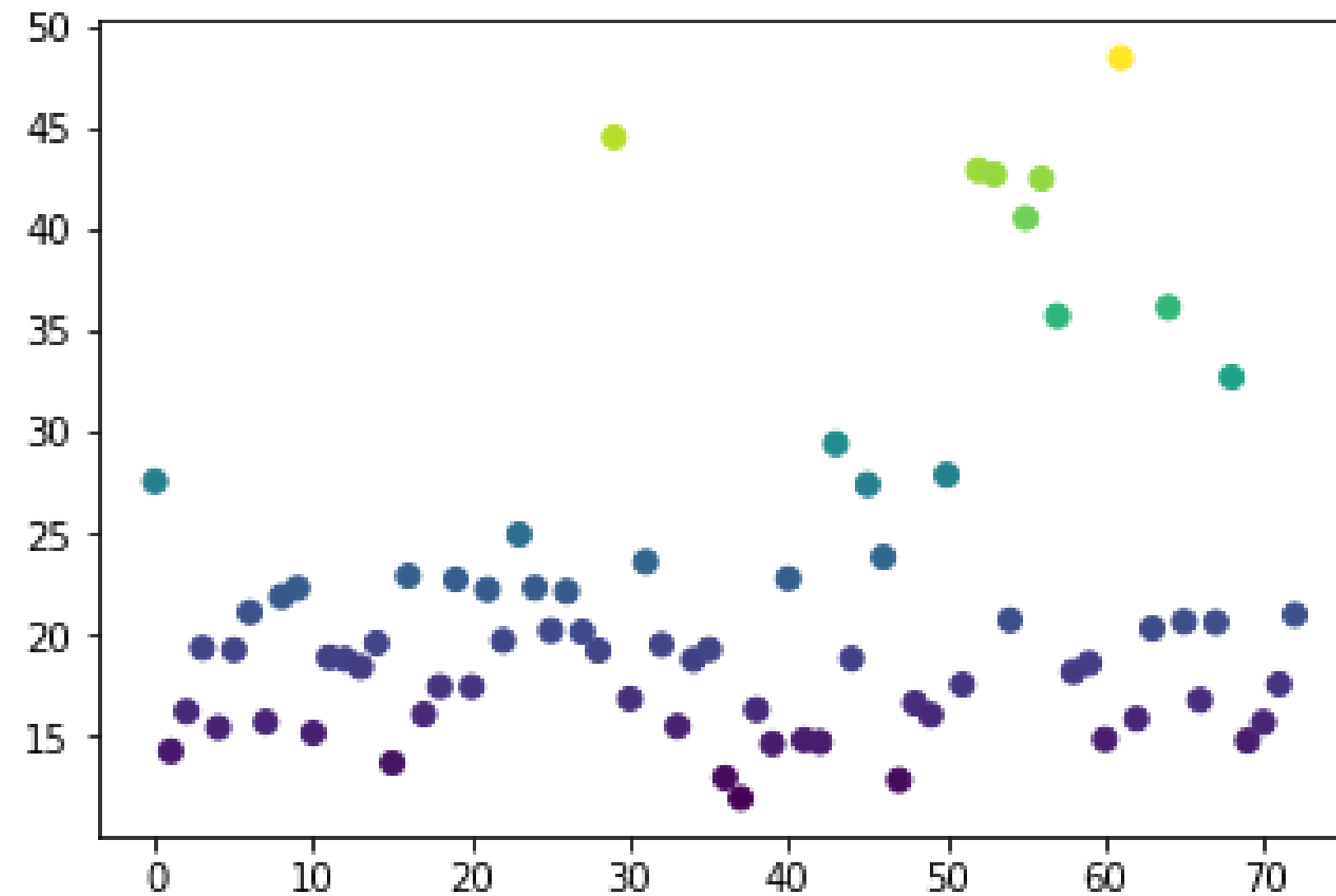
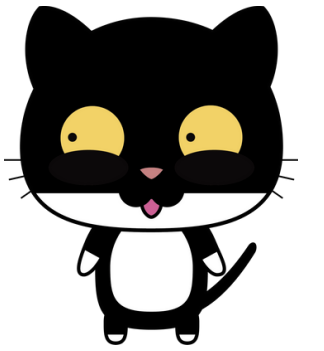
73 rows × 7 columns

ราคาเฉลี่ยของแต่ละ category



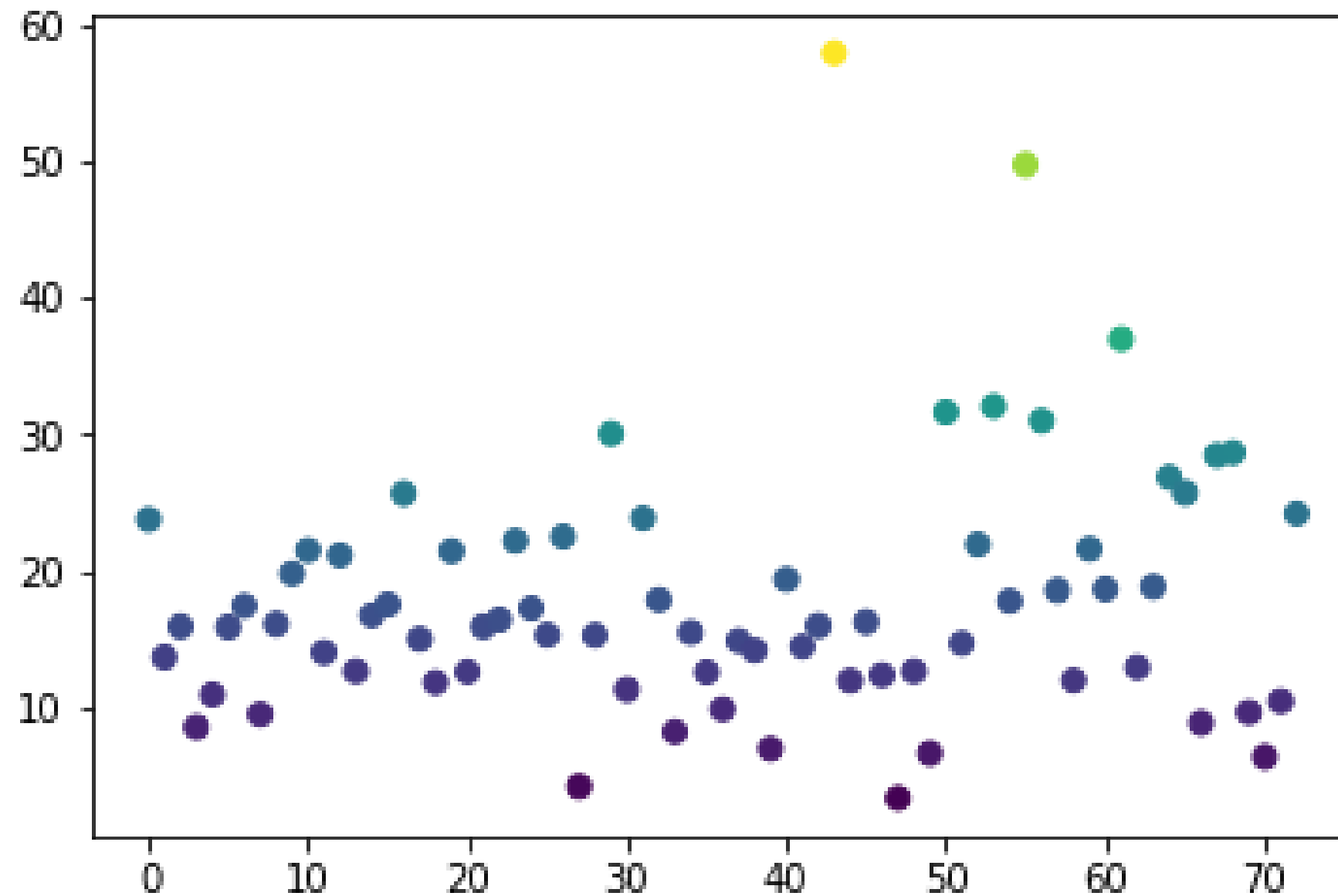
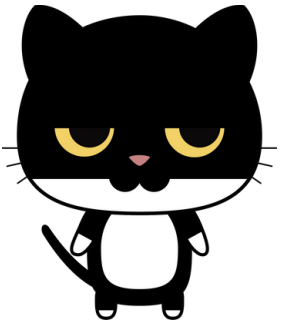
จากกราฟ จะเห็นว่า category ส่วนใหญ่มีราคาเฉลี่ยไม่เกิน 200 R\$

คำสั่งเฉลี่ยของแต่ละ category



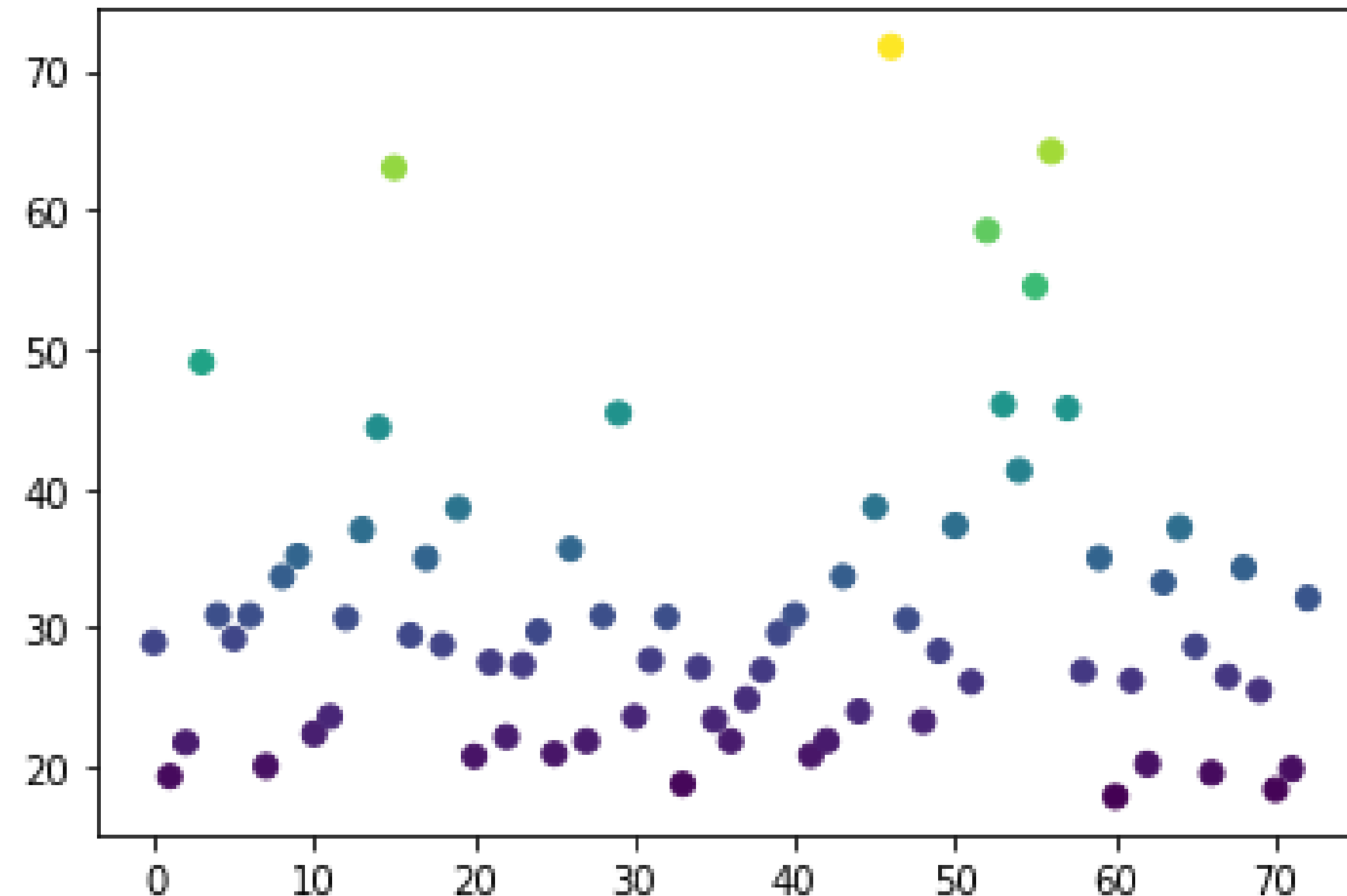
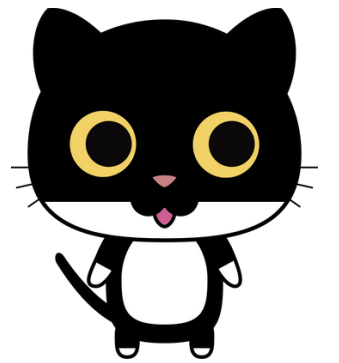
จากกราฟ จะเห็นว่า category ส่วนใหญ่มีคำสั่งเฉลี่ยไม่เกิน 25 R\$

ความสูงเฉลี่ยของแต่ละ category



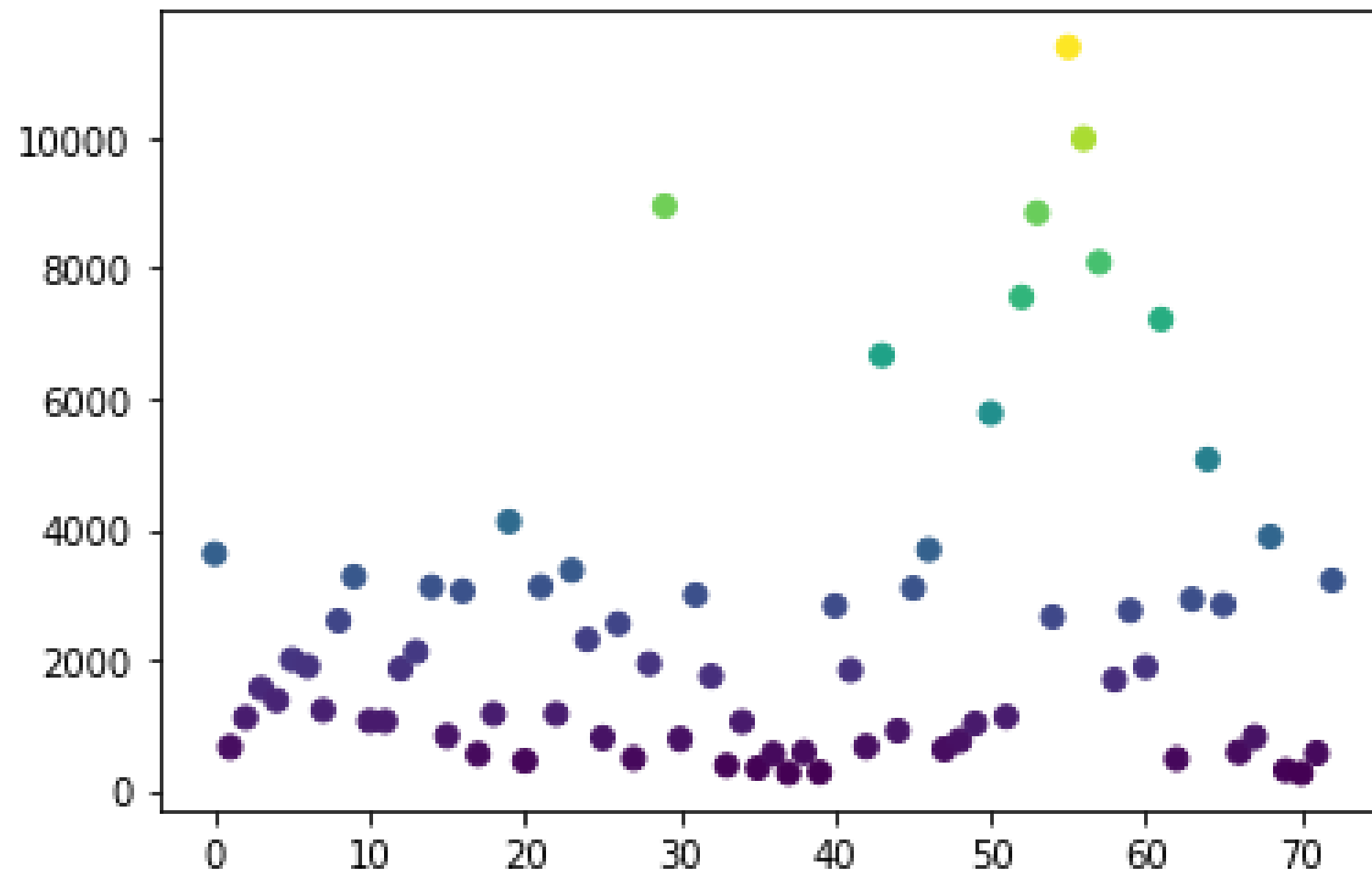
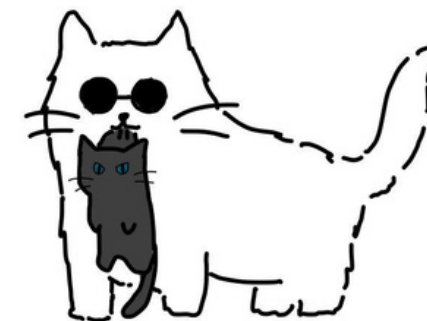
จากกราฟ จะเห็นว่า
category ส่วนใหญ่มีความสูงของสินค้าเฉลี่ยไม่เกิน 25 cm

ความยาวเฉลี่ยของแต่ละ category



จากกราฟ จะเห็นว่า category
ส่วนใหญ่มีความยาวของสินค้าเฉลี่ยไม่เกิน 40 cm

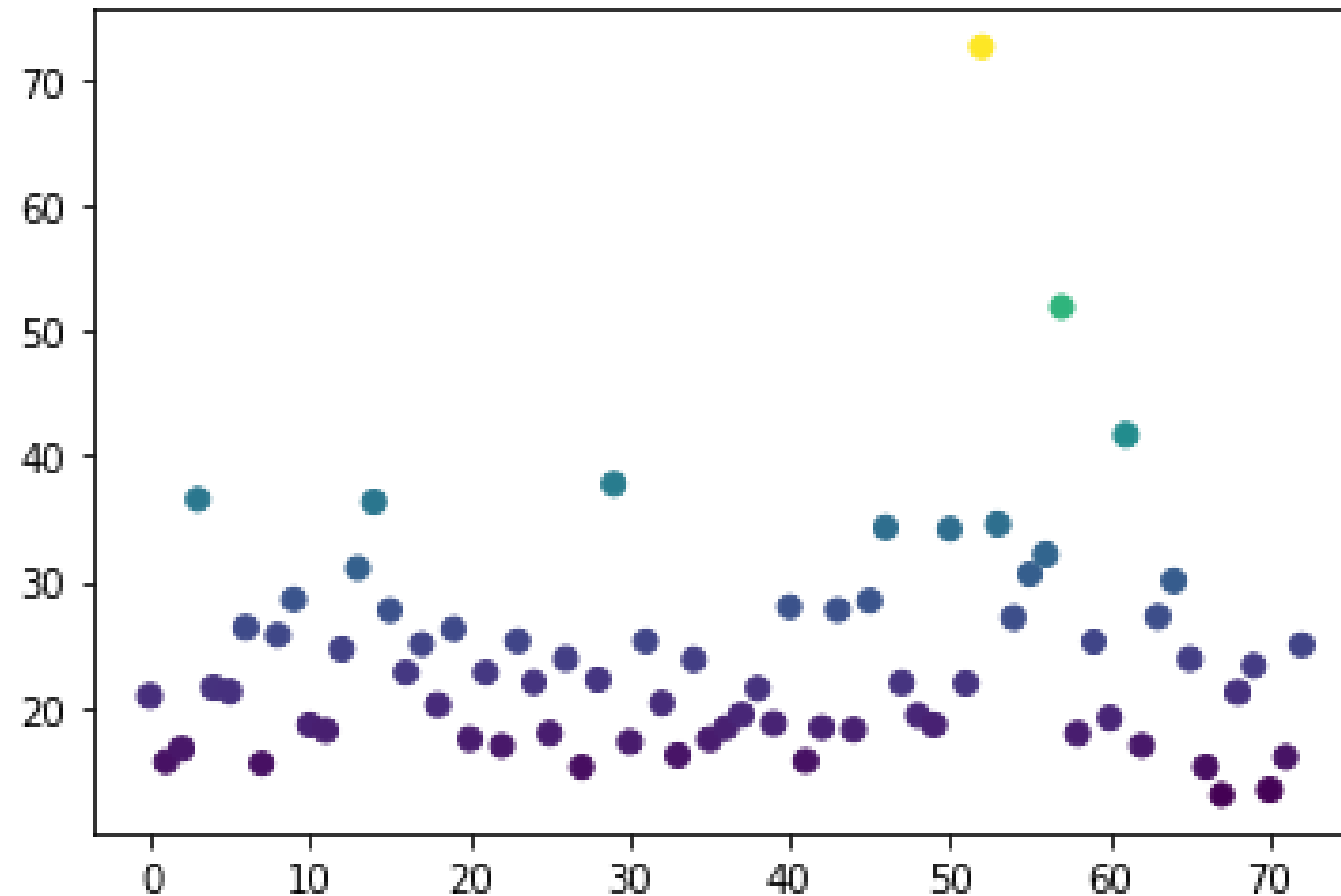
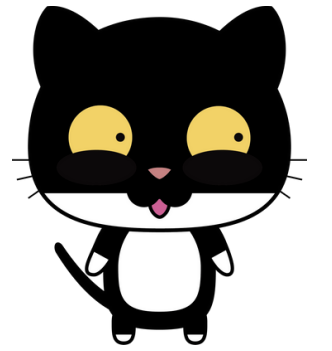
น้ำหนักเฉลี่ยของแต่ละ category



จากกราฟ จะเห็นว่า category
ส่วนใหญ่มีน้ำหนักของสินค้าเฉลี่ยไม่เกิน 4000 g



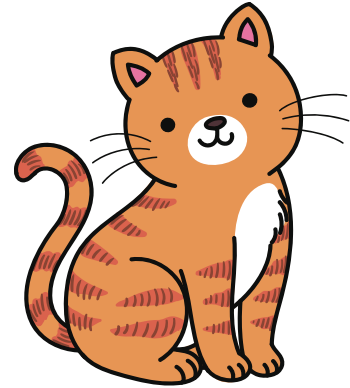
ความกว้างเฉลี่ยของแต่ละ category



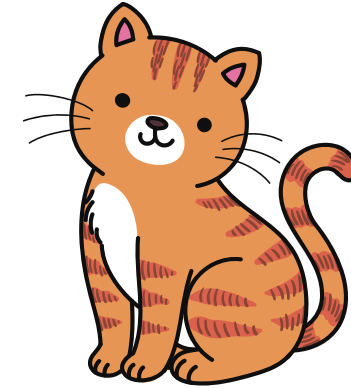
จากกราฟ จะเห็นว่า category
ส่วนใหญ่มีความกว้างของสินค้าเฉลี่ยไม่เกิน 30 cm

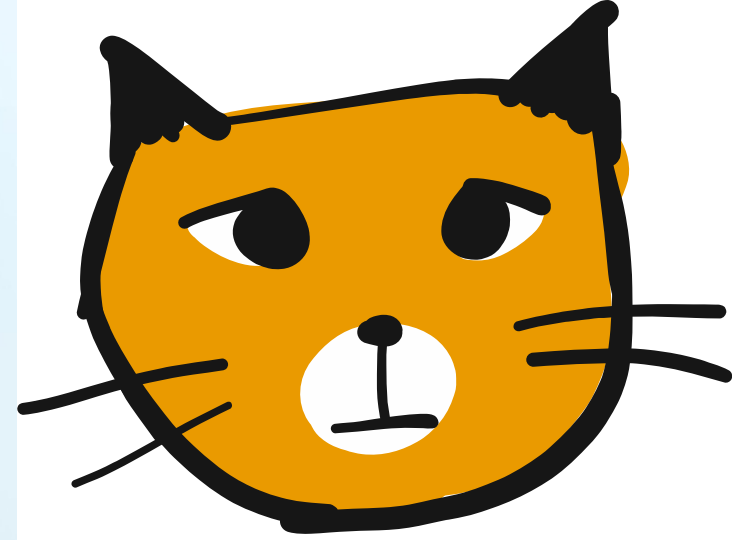


Created By



แม่จ๋า Team





- **กรกนก สังขพันธ์** **603021852-8**
- **วานิดา สมเด็จพระ** **623020536-5**
- **คำแสน แก้วพิภพ** **623020761-8**
- **จิตรกร จันทะสี** **623021044-1**
- **อาริตียา ธรรมวงษา** **623021058-0**

