Question 4

Import required libraries

In [3]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from apyori import apriori
from mlxtend.frequent_patterns import fpgrowth
from sklearn.inspection import permutation_importance
from sklearn.metrics import classification_report, mean_absolute_error, mean_squared_error
RANDOM_STATE = 123
```

Loading Data

In [4]:

```
crawled_df = pd.read_csv('Q1_Mudah_PropAds.csv')
crawled_df.head()
```

Out[4]:

	list_title	url	price	area	category	prop_type	prop_title1	p
0	New Luxury Freehold Residence 4min Walk to Mid...	https://www.mudah.my/New+Luxury+Freehold+Resid...	597000	Mid Valley City	Apartments	Condo / Services residence / Penthouse / Townh...	Freehold	
1	Sri Putramas 1 1100sqft Jalan Kuching Below Ma...	https://www.mudah.my/Sri+Putramas+1+1100sqft+J...	405000	Jalan Kuching	Apartments	Condo / Services residence / Penthouse / Townh...	Freehold	
2	0% DOWNPAYMENT Arena Green 750SF Bukit Jalil [...	https://www.mudah.my/0+DOWNPAYMENT+Arena+Green...	320000	Bukit Jalil	Apartments	Condo / Services residence / Penthouse / Townh...	Freehold	
3	[Duplex Penthouse] Silk Residence Duplex Doubl...	https://www.mudah.my/+Duplex+Penthouse+Silk+Re...	900000	Cheras	Apartments	Condo / Services residence / Penthouse / Townh...	Freehold	
4	BELOW MARKET!! Menara D'Sara Condo Sri Damansa...	https://www.mudah.my/BELOW+MARKET+Menara+D+Sar...	380000	Sri Damansara	Apartments	Condo / Services residence / Penthouse / Townh...	Freehold	

Creating List from Dataset

For Apriori, we will focus on facilities and use that as our items set

In [5]:

```
facilities_list = [row for row in crawled_df['facilities'].apply(lambda x: list(str(x).strip(',').split(', ')))]
print('Example List:\n',facilities_list[:6])
```

```
Example List:
 [['nan'], ['Swimming Pool', 'Gymnasium', 'Tennis Court', 'Squash Court', 'Mini Market', 'Playground', 'Jo
gging Track', '24 Hour Security', 'Balcony/Patio', 'Cable TV'], ['Mini Market', 'Playground', 'Jogging Tra
ck', '24 Hour Security', 'Balcony/Patio', 'Cable TV'], ['Swimming Pool', 'Gymnasium', 'Mini Market', 'Play
ground', '24 Hour Security'], ['Swimming Pool', 'Gymnasium', 'Tennis Court', 'Squash Court', 'Mini Marke
t', 'Playground', 'Jogging Track', '24 Hour Security', 'Balcony/Patio', 'Cable TV'], ['Squash Court', 'Min
i Market', 'Playground']]
```

Convert list to dataframe with boolean values

In [34]:

```
unique_items = np.unique([item for sets in facilities_list for item in sets if item != ''])
DF_dict = {}
for key in unique_items:
    DF_dict[key] = []
for transaction in facilities_list:
    for key in unique_items:
        value = any([True for item in transaction if item.find(key)!=-1])
        DF_dict[key].append(value)

bool_DF = pd.DataFrame.from_dict(DF_dict).rename(columns={'nan':'No Facility'})
bool_DF.head()
```

Out[34]:

	24 Hour Security	Balcony/Patio	Cable TV	Gymnasium	Jogging Track	Mini Market	Playground	Squash Court	Swimming Pool	Tennis Court	No Facility
0	False	False	False	False	False	False	False	False	False	False	True
1	True	True	True	True	True	True	True	True	True	True	False
2	True	True	True	False	True	True	True	False	False	False	False
3	True	False	False	True	False	True	True	False	True	False	False
4	True	True	True	True	True	True	True	True	True	True	False

Find frequently occurring itemsets using Apriori Algorithm

In [35]:

```
apriori_result = apriori(facilities_list, min_support=0.05)
```

In [36]:

```
apriori_dict ={'rules':[], 'support':[]}
for item in list(apriori_result):
    apriori_dict['rules'].append(','.join(item.items))
    apriori_dict['support'].append(item.support)
apriori_sortby_support = pd.DataFrame.from_dict(apriori_dict).sort_values('support', ascending=False).reset_index(drop
=True)
```

In [37]:

```
apriori_sortby_support
```

Out[37]:

	rules	support
0	24 Hour Security	0.6750
1	Playground	0.6400
2	Playground,24 Hour Security	0.6000
3	Balcony/Patio	0.5350
4	Mini Market	0.5325
...	...	...
1019	Playground,24 Hour Security,Gymnasium,Cable TV...	0.1725
1020	Jogging Track,24 Hour Security,Gymnasium,Cable...	0.1725
1021	Playground,Jogging Track,24 Hour Security,Gymn...	0.1725
1022	Jogging Track,Gymnasium,Squash Court,Cable TV,...	0.1725
1023	Playground,Jogging Track,24 Hour Security,Gymn...	0.1725

1024 rows × 2 columns

Find frequently occurring itemsets using FP_Growth

In [39]:

```
fpgrowth(bool_DF, min_support=0.05, use_colnames=True).sort_values('support', ascending=False)
```

Out[39]:

	support	itemsets
1	0.6750	(24 Hour Security)
2	0.6400	(Playground)
11	0.6000	(Playground, 24 Hour Security)
3	0.5350	(Balcony/Patio)
4	0.5325	(Mini Market)
...	...	...
903	0.1725	(Gymnasium, Tennis Court, Squash Court, Cable TV)
966	0.1725	(Playground, Jogging Track, Gymnasium, Squash ...
909	0.1725	(Jogging Track, Gymnasium, Squash Court, Cable...
963	0.1725	(Playground, Jogging Track, 24 Hour Security, ...
1023	0.1725	(Playground, Jogging Track, 24 Hour Security, ...

1024 rows × 2 columns

Mine the Association Rules

Association Rules can be retrieved using apriori like what was done earlier

In [42]:

```
apriori_result = apriori(facilities_list, min_support=0.05)
```

In [43]:

```
apriori_dict ={'rules':[], 'support':[], 'items_base':[], 'items_add':[], 'confidence':[], 'lift':[]}
for item in list(apriori_result):
    for statistic in item.ordered_statistics:
        apriori_dict['rules'].append(','.join(item.items))
        apriori_dict['support'].append(item.support)
        apriori_dict['items_base'].append(','.join(statistic[0]))
        apriori_dict['items_add'].append(','.join(statistic[1]))
        apriori_dict['confidence'].append(statistic[2])
        apriori_dict['lift'].append(statistic[3])
apriori_sortby_lift = pd.DataFrame.from_dict(apriori_dict).sort_values('lift', ascending=False).reset_index(drop=True)
```

In [44]:

```
apriori_sortby_lift
```

Out[44]:

	rules	support	items_base	items_add	confidence	lift
0	Playground,Jogging Track,24 Hour Security,Gymn...	0.1725	Playground,Jogging Track,24 Hour Security,Squa...	Gymnasium,Tennis Court,Cable TV	0.920000	4.717949
1	Playground,Jogging Track,24 Hour Security,Gymn...	0.1725	Playground,Jogging Track,Squash Court,Mini Mar...	Gymnasium,Cable TV,Tennis Court,24 Hour Security	0.920000	4.717949
2	Playground,Jogging Track,Gymnasium,Cable TV,Sq...	0.1725	Gymnasium,Tennis Court,Cable TV	Playground,Jogging Track,Squash Court,Swimming...	0.884615	4.717949
3	Jogging Track,24 Hour Security,Gymnasium,Cable...	0.1725	Jogging Track,Squash Court,Swimming Pool,Mini ...	Gymnasium,Cable TV,Tennis Court,24 Hour Security	0.920000	4.717949
4	Playground,Jogging Track,Gymnasium,Cable TV,Sq...	0.1725	Gymnasium,Tennis Court,Cable TV,Swimming Pool	Playground,Jogging Track,Squash Court,Mini Mar...	0.884615	4.717949
...	...	...	...	...	...	...
58021	Playground,Jogging Track,24 Hour Security,Gymn...	0.1925		Playground,Jogging Track,24 Hour Security,Gymn...	0.192500	1.000000
58022	Jogging Track,24 Hour Security,Gymnasium,Squas...	0.1850		Jogging Track,24 Hour Security,Balcony/Patio,G...	0.185000	1.000000
58023	Playground,Jogging Track,24 Hour Security,Gymn...	0.3450		Playground,Jogging Track,24 Hour Security,Gymn...	0.345000	1.000000
58024	Playground,Jogging Track,24 Hour Security,Gymn...	0.2150		Playground,Tennis Court,Jogging Track,24 Hour ...	0.215000	1.000000
58025	24 Hour Security	0.6750		24 Hour Security	0.675000	1.000000

58026 rows × 6 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]: