In [124…

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
df = pd.read_csv(r"C:\Users\LEN\Downloads\machines_data.csv")
```

In [125…

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19 entries, 0 to 18
Data columns (total 12 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Machine                      19 non-null     object 
 1   Price(INR)                   19 non-null     object 
 2   Size of Area                 19 non-null     int64  
 3   capacity                     19 non-null     int64  
 4   Maintenance                  19 non-null     object 
 5   Maneuverability              19 non-null     object 
 6   Power source                 19 non-null     object 
 7   Bulkyness                    19 non-null     int64  
 8   Noise level                  19 non-null     int64  
 9   Filtration system            19 non-null     object 
 10  Propel Speed                 19 non-null     float64
 11  Performance(Productivity)    19 non-null     int64  
dtypes: float64(1), int64(5), object(6)
memory usage: 1.9+ KB
```

In [126…

```python
df.head(20)
```

Out[126…

	Machine	Price(INR)	Size of Area	capacity	Maintenance	Maneuverability	Power source	Bulkyness	Noise level
0	Tennant	3,00,000	5766	3520	easy	easy	LPG,DIESEL	940	50
1	Lavour Free Evo 50E	2,25,000	8930	89	easy	easy	BATTERY	320	56
2	Cyclone FP240	94,500	7730	285	easy	easy	BATTERY	200	70
3	GCT 32L	1,06,000	2280	42	easy	easy	BATTERY	90	50
4	ruby 50	1,00,000	2500	80	hard	hard	DIESEL	3200	80
5	S30 ride on sweeper	5,00,000	26010	395	easy	easy	LPG,GAS	1620	80
6	karcher KM 125/130 R Bp pack	3,00,000	13600	130	easy	easy	BATTERY	840	40
7	Nilfisk SW250	50,000	6000	38	easy	easy	BATTERY	400	80

	Machine	Price(INR)	Size of Area	capacity	Maintenance	Maneuverability	Power source	Bulkyness	Noise level
	walk behind sweeper								
8	IPC Eagle S3 walk behind sweeper	3,75,000	27000	64	easy	easy	LPG,DIESEL	600	64
9	Hako-Jonas 1450v ride on	4,00,000	16200	27	hard	easy	LPG,DIESEL	1080	40
10	Dulevo 200 Quattro Sweeper	5,00,000	19000	2500	hard	hard	DIESEL	4000	40
11	Power Boss Nautilus High Dump	2,00,000	33017	90	easy	easy	BATTERY	4800	70
12	RCM cruises sweeper	2,50,000	10000	120	hard	hard	BATTERY	800	80
13	Factory cat XR sweeper	1,50,000	25000	78	easy	easy	BATTERY	1950	50
14	Comac C130 BSB sweeper	1,20,000	25000	300	easy	easy	DIESEL	810	70
15	diversy TASKI Swingo XP sweeper	1,75,000	2250	113	easy	easy	BATTERY	905	70
16	RCM Atom plus sweeper	4,93,638	7500	95	easy	hard	PETROL	600	40
17	City Master 1600 sweeper	4,49,089	1300	180	easy	easy	DIESEL	2490	40
18	Nilfisk SW 8000 ride on sweeper	3,00,000	2000	200	easy	easy	DIESEL	465	50

In [127…

```python
print(df.columns)
```

```
Index(['Machine ', 'Price(INR)', 'Size of Area', 'capacity', 'Maintenance ',
       'Maneuverability', 'Power source ', 'Bulkyness', 'Noise level',
       'Filtration system', 'Propel Speed', 'Performance(Productivity)'],
      dtype='object')
```

In [128…

```python
# changing data type to categorical value
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
df['Power source '] = label_encoder.fit_transform(df['Power source '])
df['Maintenance '] = label_encoder.fit_transform(df['Maintenance '])
df['Filtration system'] = label_encoder.fit_transform(df['Filtration system'])
df['Maneuverability'] = label_encoder.fit_transform(df['Maneuverability'])

df.head()
```

Out[128…

	Machine	Price(INR)	Size of Area	capacity	Maintenance	Maneuverability	Power source	Bulkyness	Noise level	Filtrat syst
0	Tennant	3,00,000	5766	3520	0	2	2	940	50	
1	Lavour Free Evo 50E	2,25,000	8930	89	0	2	0	320	56	
2	Cyclone FP240	94,500	7730	285	0	2	0	200	70	
3	GCT 32L	1,06,000	2280	42	0	2	0	90	50	
4	ruby 50	1,00,000	2500	80	1	3	1	3200	80	

In [129…

```python
df.head(20)
```

Out[129…

	Machine	Price(INR)	Size of Area	capacity	Maintenance	Maneuverability	Power source	Bulkyness	Noise level	Filtr sy
0	Tennant	3,00,000	5766	3520	0	2	2	940	50	
1	Lavour Free Evo 50E	2,25,000	8930	89	0	2	0	320	56	
2	Cyclone FP240	94,500	7730	285	0	2	0	200	70	
3	GCT 32L	1,06,000	2280	42	0	2	0	90	50	
4	ruby 50	1,00,000	2500	80	1	3	1	3200	80	
5	S30 ride on sweeper	5,00,000	26010	395	0	1	3	1620	80	

	Machine	Price(INR)	Size of Area	capacity	Maintenance	Maneuverability	Power source	Bulkyness	Noise level	Filt sy
6	karcher KM 125/130 R Bp pack	3,00,000	13600	130	0	1	0	840	40	
7	Nilfisk SW250 walk behind sweeper	50,000	6000	38	0	1	0	400	80	
8	IPC Eagle S3 walk behind sweeper	3,75,000	27000	64	0	1	2	600	64	
9	Hako-Jonas 1450v ride on	4,00,000	16200	27	1	1	2	1080	40	
10	Dulevo 200 Quattro Sweeper	5,00,000	19000	2500	1	3	1	4000	40	
11	Power Boss Nautilus High Dump	2,00,000	33017	90	0	1	0	4800	70	
12	RCM cruises sweeper	2,50,000	10000	120	1	3	0	800	80	
13	Factory cat XR sweeper	1,50,000	25000	78	0	2	0	1950	50	
14	Comac C130 BSB sweeper	1,20,000	25000	300	0	1	1	810	70	
15	diversy TASKI Swingo XP sweeper	1,75,000	2250	113	0	1	0	905	70	
16	RCM Atom plus sweeper	4,93,638	7500	95	0	3	4	600	40	

	Machine	Price(INR)	Size of Area	capacity	Maintenance	Maneuverability	Power source	Bulkyness	Noise level	Filt sy
17	City Master 1600 sweeper	4,49,089	1300	180	0	1	1	2490	40	
18	Nilfisk SW 8000 ride on sweeper	3,00,000	2000	200	0	0	1	465	50	

In [130…

```
df['Price(INR)'] = df['Price(INR)'].replace(',', '', regex=True).astype(float)
```

In [131…

```
criteria = ['Price', 'Size of Area', 'Capacity', 'Maintenance', 'Maneuverability', 'Pow
weights = [1/11] * 11
```

In [133…

```
# Import necessary libraries
import pandas as pd
import numpy as np

# Define the decision matrix
X = df.iloc[:, 1:].values

# Define the weight vector and the impact vector

w = np.array([0.2, 0.1, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05])
impacts = np.array([1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1])

# Step 1 - Normalize the decision matrix
X_norm = X / np.linalg.norm(X, axis=0)

# Step 2 - Calculate the weighted normalized decision matrix
X_weighted = X_norm * w

# Step 3 - Determine the ideal and anti-ideal solutions
ideal = np.max(X_weighted, axis=0)
anti_ideal = np.min(X_weighted, axis=0)

# Step 4 - Calculate the distance to the ideal and anti-ideal solutions
d_plus = np.sqrt(np.sum((X_weighted - ideal) ** 2, axis=1))
d_minus = np.sqrt(np.sum((X_weighted - anti_ideal) ** 2, axis=1))

# Step 5 - Calculate the closeness coefficient
cc = d_minus / (d_plus + d_minus)

# Step 6 - Calculate the relative closeness coefficient
rcc = cc / np.sum(cc)

# Step 7 - Rank the alternatives based on the relative closeness coefficient
df['Rank'] = rcc
df = df.sort_values('Rank', ascending=False)
```

In [134…

```python
best_machine = df.sort_values('Rank', ascending=False).iloc[0]
print('The best floor cleaning machine is:')
print(best_machine)
```

```
The best floor cleaning machine is:
Machine                      Dulevo 200 Quattro Sweeper
Price(INR)                                     500000.0
Size of Area                                      19000
capacity                                           2500
Maintenance                                           1
Maneuverability                                       3
Power source                                          1
Bulkyness                                          4000
Noise level                                          40
Filtration system                                     4
Propel Speed                                        8.0
Performance(Productivity)                         78000
Rank                                           0.116578
Name: 10, dtype: object
```

In [136…

```python
# ranks of all machines

df_sorted = df.sort_values(by=['Rank'], ascending=False)
print(df_sorted[['Machine ', 'Rank']])
```

```
                             Machine       Rank
10         Dulevo 200 Quattro Sweeper  0.116578
0                            Tennant   0.084652
5                 S30 ride on sweeper  0.073243
9            Hako-Jonas 1450v ride on  0.072934
16              RCM Atom plus sweeper  0.065521
8    IPC Eagle S3 walk behind sweeper  0.062230
12                RCM cruises sweeper  0.059985
17           City Master 1600 sweeper  0.059196
11      Power Boss Nautilus High Dump  0.057601
4                             ruby 50  0.053479
18    Nilfisk SW 8000 ride on sweeper  0.047023
6        karcher KM 125/130 R Bp pack  0.045748
13             Factory cat XR sweeper  0.042817
14             Comac C130 BSB sweeper  0.038974
1                 Lavour Free Evo 50E  0.035378
15    diversy TASKI Swingo XP sweeper  0.028024
2                      Cyclone FP240   0.021181
3                             GCT 32L  0.020934
7   Nilfisk SW250 walk behind sweeper  0.014500
```

In [ ]:

```

```