In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression,Ridge,ElasticNet
```

In [2]:

```
df=pd.read_csv("C:\\Users\\swast\\Desktop\\tcs sustainathon\\E WASTE INDIA.csv")
```

In [3]:

```
df
```

Out[3]:

	year	ewaste(MT)
0	2007.0	332979.0
1	2009.0	402905.0
2	2011.0	487515.0
3	2013.0	589893.0
4	2015.0	713770.0
5	2017.0	863662.0
6	2019.0	1045031.0

In [4]:

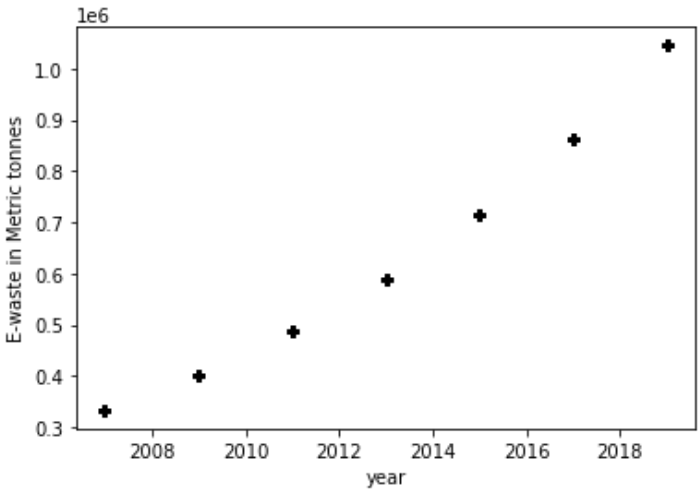
```
x=list(df.iloc[:,0])
y=list(df.iloc[:,1])
x_train=df.iloc[:,0].values.reshape(-1,1)
y_train=df.iloc[:,1].values.reshape(-1,1)
```

In [5]:

```
plt.scatter(x,y,marker='P',color='BLACK')
plt.xlabel("year")
plt.ylabel("E-waste in Metric tonnes")
```

Out[5]:

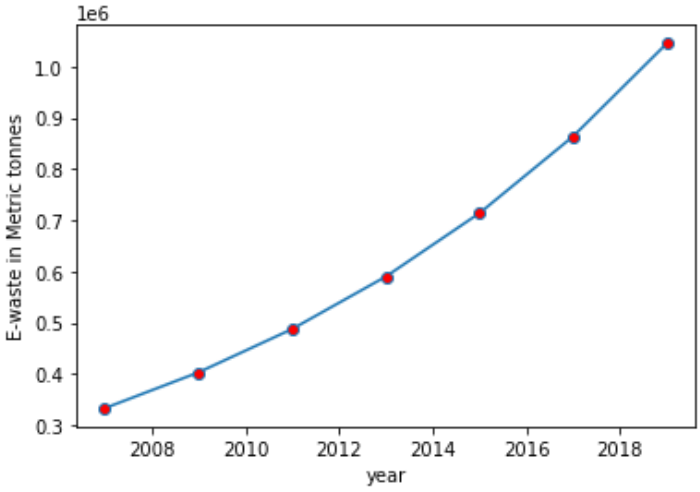
```
Text(0, 0.5, 'E-waste in Metric tonnes')
```



In [6]:

```
plt.plot(x,y,marker='o',mfc='red')
```

```
plt.xlabel("year")
plt.ylabel("E-waste in Metric tonnes")
plt.show()
```

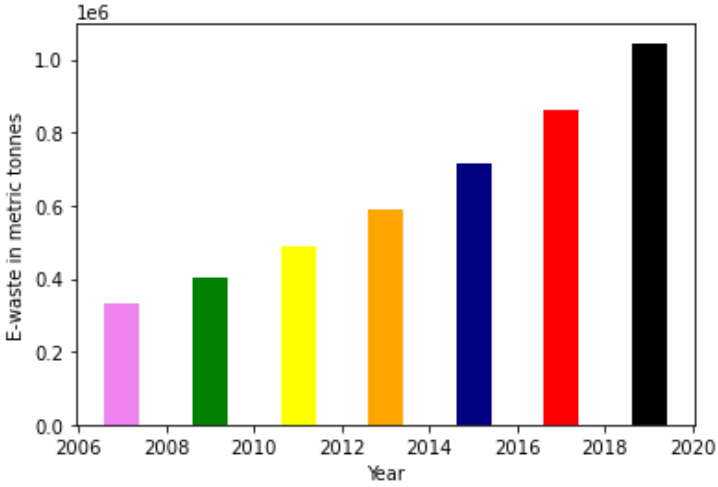


In [7]:

```
plt.bar(x,y,color=['violet','green','yellow','orange','navy','red','black'])
plt.xlabel('Year')
plt.ylabel('E-waste in metric tonnes')
```

Out[7]:

```
Text(0, 0.5, 'E-waste in metric tonnes')
```



In [9]:

```
model=np.polyfit(x,y,3)
lin=LinearRegression()
Rig=Ridge()
eln=ElasticNet()
predict = np.poly1d(model)
model
```

Out[9]:

```
array([ 8.69444424e+01, -5.22299681e+05,  1.04589400e+09, -6.98144682e+11])
```

In [ ]:

```

```

In [10]:

```
lin.fit(x_train,y_train)
Rig.fit(x_train,y_train)
eln.fit(x_train,y_train)
```

Out[10]:

```
ElasticNet()
```

In [12]:

```
print(lin.predict([[2021]]))
print(Rig.predict([[2021]]))
print(eln.predict([[2021]]))
print(predict(2021))
print(predict(2030))
```

```
[[1102811.42857143]]
[[1098659.81668773]]
[1088595.06060606]
1260226.142211914
2769238.771118164
```

INFERENCE :

the poly fit function seems to give us a much accurate result compared to the machine learning linear regression techniques which could have been predicted by observing the scatter plot. so the estimated E-waste seems to be 12,60,226 metric tonnes in 2021 which seems to exponentially grow to about 28,00,0000 metric tonnes per year by 2030.

-- There's 80 times as much gold in one ton of cellphones as there is in a gold mine, says Federico Magalini, an expert on electronic waste. That means there's enormous potential for recycling.For , one ton of mobile phones, there is usually about 350 grams of gold

```
    ---https://www.theverge.com/2018/4/23/17270960/electronic-waste-urban-mining-mat
   erials-recycling
```

In [ ]:

```
t=(2769238.771118164*1.10231)       #tonnes to tons
cost=670000*t                       # as of 07-09-2021 price of 200gm of gold in india in
rupees is about 670000
cost
```

considering an average of 200 gm of gold to be retrieved by recycling 1 ton of E-waste by the year 2030(not considering inflation and price change of gold) generates a total sum of about 27,850 million dollars by our calculation . hence large revenues could be collected by recycling E-waste and indeed recovering the metal scrap and gold in it . unfortunately conservative mindsets and lack of proper infrastructure for efficient recycling of E-waste make the process of E-waste recycling unpopular in India.

In [26]:

```
lst=list(range(2021,2032))
lst
```

Out[26]:

```
[2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031]
```

In [28]:

```
lst2=[]
for i in lst:
    lst2.append(predict(i))
lst2
```

Out[28]:

```
[1260226.142211914,
 1382182.8712158203,
 1514350.2133789062,
 1657249.8350830078,
```

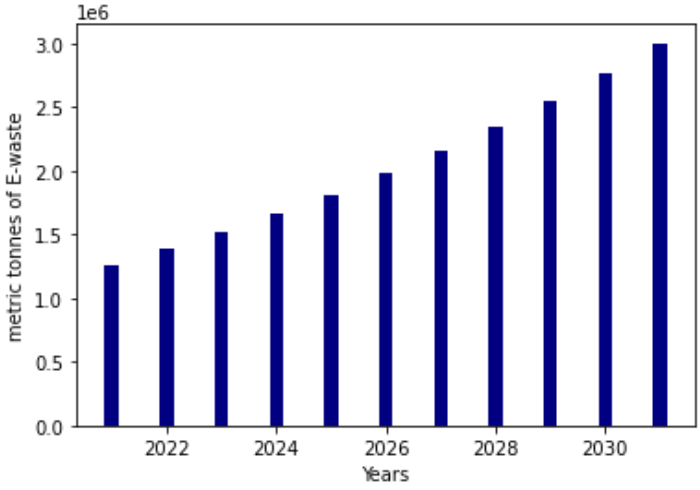
```
1811403.4029541016,
1977332.583984375,
2155559.0446777344,
2346604.451538086,
2550990.471435547,
2769238.771118164,
3001871.016845703]
```

In [30]:

```
plt.bar(lst,lst2,width=0.25,color="navy")
plt.xlabel("Years")
plt.ylabel("metric tonnes of E-waste")
```

Out[30]:

```
Text(0, 0.5, 'metric tonnes of E-waste')
```



above is the predicted graph of e-waste growth which has the potential to increase the revenue by a significnt percent and indeed also contribute to India's GDP by a good decimal percentage

In [ ]:

```
```