In [117]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
import numpy as np
```

In [201]:
```
df = pd.read_csv('plot.csv')
df
```

Out[201]:

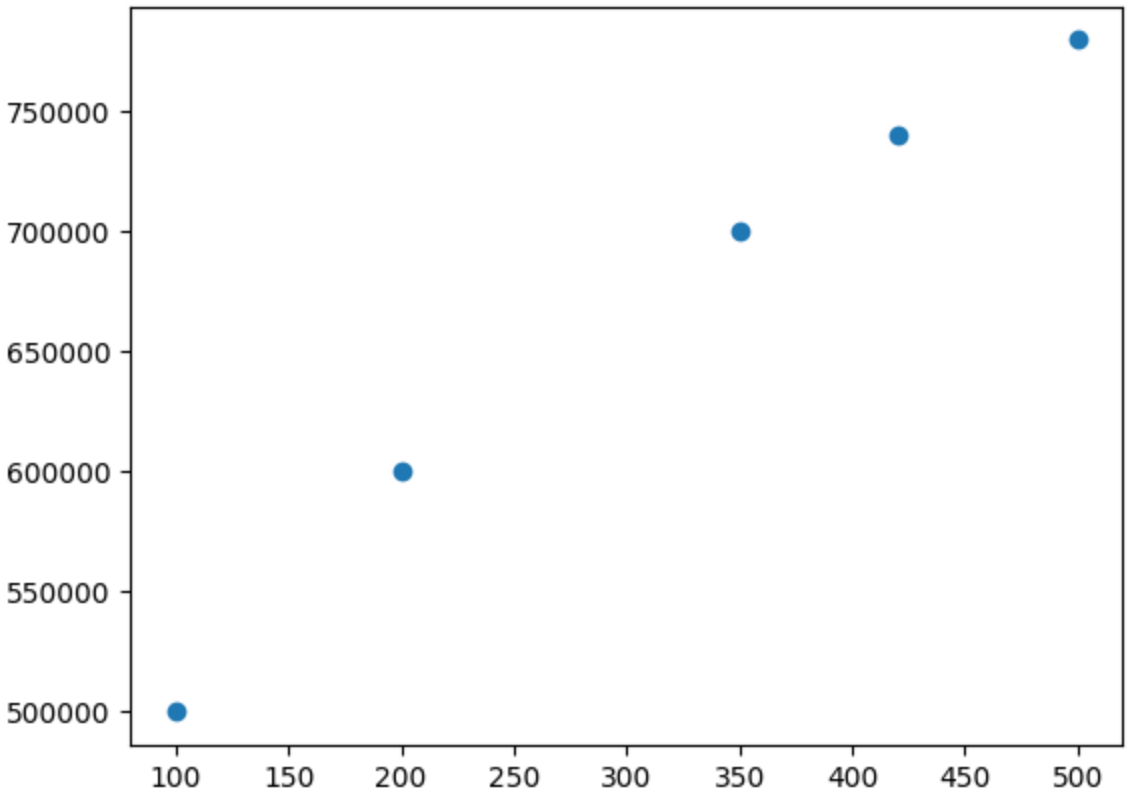
	size	rate
0	100	50000
1	150	52000
2	200	56000
3	250	61000
4	300	65000
5	350	70000
6	400	76000

In [202]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   size    7 non-null      int64
 1   rate    7 non-null      int64
dtypes: int64(2)
memory usage: 244.0 bytes
```

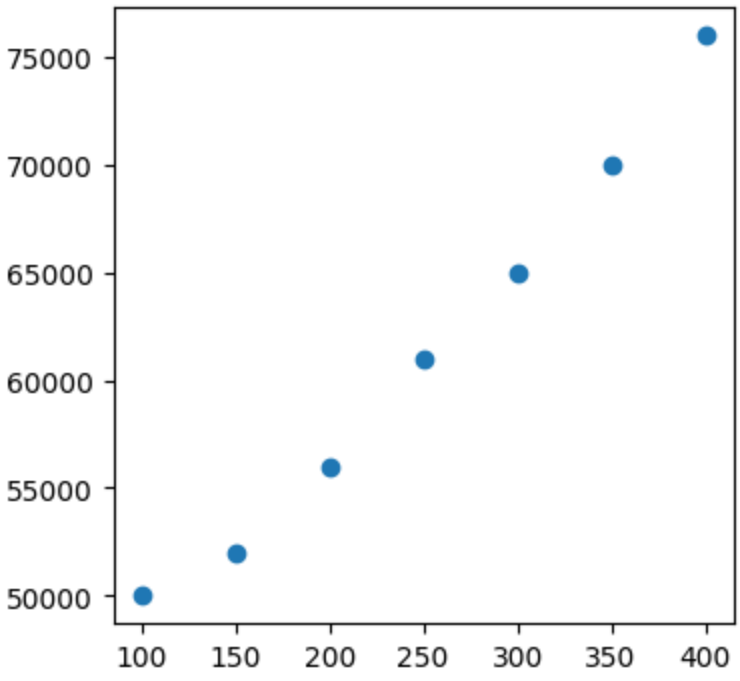
In [203]:
```
x =[100,200,350,420,500]
y =[500000,600000,700000,740000,780000]
plt.scatter(x,y)
```

Out[203]: <matplotlib.collections.PathCollection at 0x1e83d982ed0>



In [204]:
```
plt.figure(figsize=(4,4))
plt.scatter(df['size'], df['rate'])
```

Out[204]: <matplotlib.collections.PathCollection at 0x1e83e025050>



In [205]:
```
df['rate']
```

Out[205]:
```
0    50000
1    52000
2    56000
3    61000
4    65000
5    70000
6    76000
Name: rate, dtype: int64
```

In [206]:
```
df.describe()
```

Out[206]:

	size	rate
count	7.000000	7.000000
mean	250.000000	61428.571429
std	108.012345	9554.355776
min	100.000000	50000.000000
25%	175.000000	54000.000000
50%	250.000000	61000.000000
75%	325.000000	67500.000000
max	400.000000	76000.000000

In [207]:
```
df['size']
```

Out[207]:
```
0    100
1    150
2    200
3    250
4    300
5    350
6    400
Name: size, dtype: int64
```

In [211]:
```
# Machine learning
a = linear_model.LinearRegression()
a.fit(df[['size']], df['rate'])
```

Out[211]:
```
▾ LinearRegression
LinearRegression()
```

In [210]:
```
a
```

Out[210]:
```
▾ LinearRegression
LinearRegression()
```

In [209]:
```
a.predict([[2]])
```

```
C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Lib\site-packa
ges\sklearn\base.py:465: UserWarning: X does not have valid feature names, bu
t LinearRegression was fitted with feature names
  warnings.warn(
```

Out[209]:
```
array([39640.])
```

In [212]:
```
df1= df.drop('rate', axis=1)
df1
```

Out[212]:

	size
0	100
1	150
2	200
3	250
4	300
5	350
6	400

In [213]:
```
df
```

Out[213]:

	size	rate
0	100	50000
1	150	52000
2	200	56000
3	250	61000
4	300	65000
5	350	70000
6	400	76000

In [214]:
```
df1
```

Out[214]:

	size
0	100
1	150
2	200
3	250
4	300
5	350
6	400

In [215]:
```
df2 = df['rate']
df2
```

Out[215]:
```
0    50000
1    52000
2    56000
3    61000
4    65000
5    70000
6    76000
Name: rate, dtype: int64
```

In [216]:
```
df['rate']
```

Out[216]:
```
0    50000
1    52000
2    56000
3    61000
4    65000
5    70000
6    76000
Name: rate, dtype: int64
```

In [217]:
```
b = linear_model.LinearRegression()
b.fit(df1,df2)
```

Out[217]:
```
LinearRegression()
```

In [218]:
```
b.predict([[100]])
```

```
C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Lib\site-packa
ges\sklearn\base.py:465: UserWarning: X does not have valid feature names, bu
t LinearRegression was fitted with feature names
  warnings.warn(
```

Out[218]:
```
array([48250.])
```

In [220]:
```
# #features
a.coef_
```

Out[220]:
```
array([87.85714286])
```

In [221]:
```
a.intercept_
```

Out[221]:
```
39464.28571428571
```

In [222]:
```
c =pd.read_csv('area.csv')
c
```

Out[222]:

	size
0	333
1	458
2	657
3	543
4	1287
5	4329
6	2309
7	1859
8	969
9	1431
10	1440

In [223]:
```
b.predict(c)
```

Out[223]:
```
array([ 68720.71428571,  79702.85714286,  97186.42857143,  87170.71428571,
       152536.42857143, 419797.85714286, 242326.42857143, 202790.71428571,
       124597.85714286, 165187.85714286, 165978.57142857, 141817.85714286,
       117130.        ,  93847.85714286])
```

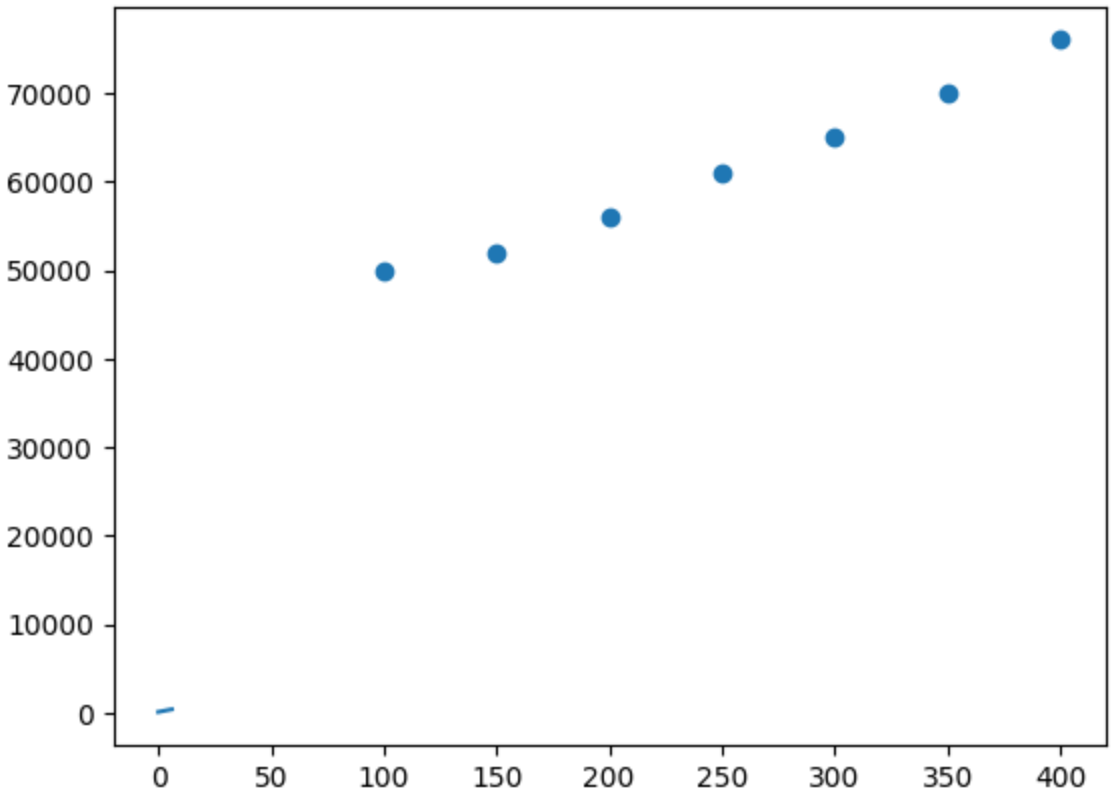
In [224]:
```
d = b.predict(c)
```

In [225]:
```
c['estimations']= d
```

In [226]:
```
c.to_csv('prediction.csv', index=False)
```

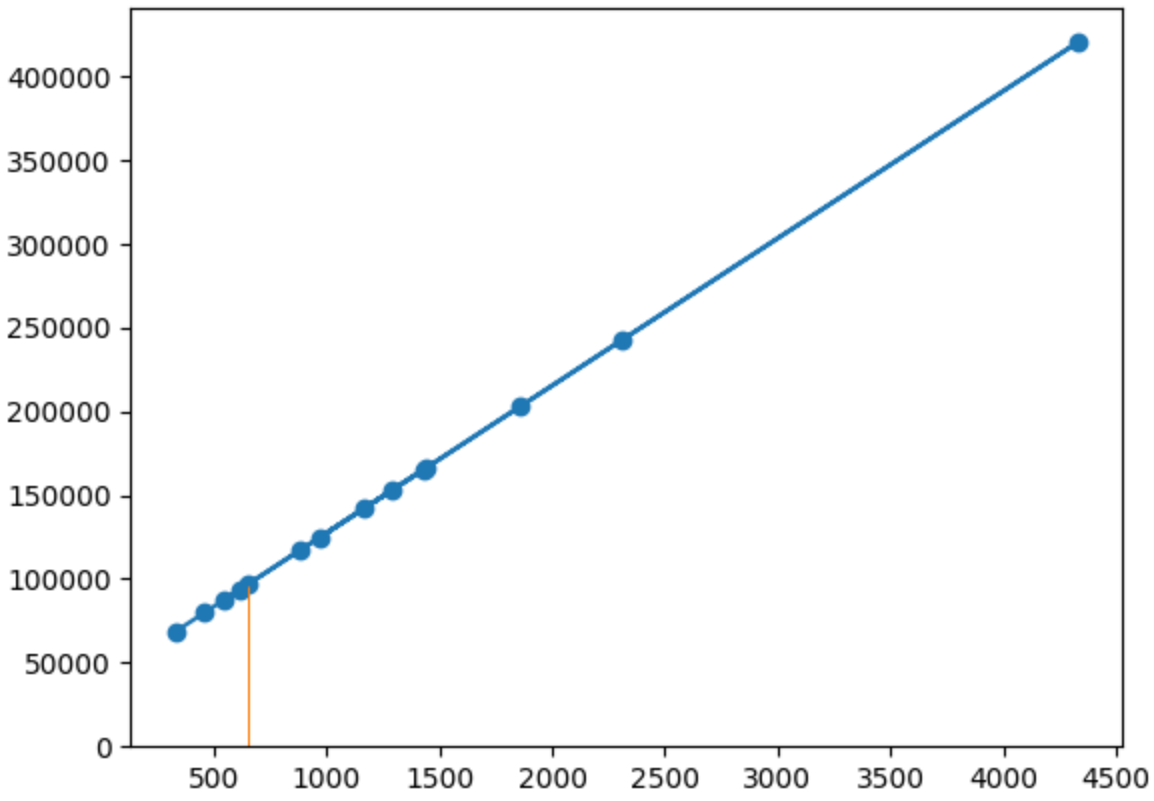
In [227]:
```
plt.scatter(df['size'], df['rate'])
plt.plot(df[['size']])
```

Out[227]:
```
[<matplotlib.lines.Line2D at 0x1e83f0ca750>]
```



In [228]:
```
df_f = pd.read_csv('prediction.csv')
plt.scatter(df_f['size'], df_f['estimations'])
plt.plot(df_f['size'], df_f['estimations'])
plt.bar(df_f['size'], df_f['estimations'])
```

Out[228]: <BarContainer object of 14 artists>



In [ ]: