ML Day2

March 18, 2023

1 Handling Missing values

[1]:
```
import seaborn as sns
```

[2]:
```
df=sns.load_dataset("titanic")
```

[3]:
```
df
```

[3]:
```
     survived  pclass     sex   age  sibsp  parch     fare embarked   class  \
0           0       3    male  22.0      1      0   7.2500        S   Third
1           1       1  female  38.0      1      0  71.2833        C   First
2           1       3  female  26.0      0      0   7.9250        S   Third
3           1       1  female  35.0      1      0  53.1000        S   First
4           0       3    male  35.0      0      0   8.0500        S   Third
..        …       …       …     …      …      …        …        …       …
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
888         0       3  female   NaN      1      2  23.4500        S   Third
889         1       1    male  26.0      0      0  30.0000        C   First
890         0       3    male  32.0      0      0   7.7500        Q   Third

       who  adult_male deck  embark_town alive  alone
0      man        True  NaN  Southampton    no  False
1    woman       False    C    Cherbourg   yes  False
2    woman       False  NaN  Southampton   yes   True
3    woman       False    C  Southampton   yes  False
4      man        True  NaN  Southampton    no   True
..     …           …    …          …     …      …
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]
```

[4]:
```
df.isna()
```

```
[4]:      survived  pclass    sex    age  sibsp  parch   fare  embarked  class  \
0       False   False  False  False  False  False  False     False  False
1       False   False  False  False  False  False  False     False  False
2       False   False  False  False  False  False  False     False  False
3       False   False  False  False  False  False  False     False  False
4       False   False  False  False  False  False  False     False  False
..        …       …     …      …     …     …     …         …      …
886     False   False  False  False  False  False  False     False  False
887     False   False  False  False  False  False  False     False  False
888     False   False  False   True  False  False  False     False  False
889     False   False  False  False  False  False  False     False  False
890     False   False  False  False  False  False  False     False  False

       who  adult_male   deck  embark_town  alive  alone
0    False       False   True        False  False  False
1    False       False  False        False  False  False
2    False       False   True        False  False  False
3    False       False  False        False  False  False
4    False       False   True        False  False  False
..     …           …     …            …      …      …
886  False       False   True        False  False  False
887  False       False  False        False  False  False
888  False       False   True        False  False  False
889  False       False  False        False  False  False
890  False       False   True        False  False  False

[891 rows x 15 columns]
```

```
[5]: df.isna().sum()
```

```
[5]: survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

[6]:
```
df.drop(["deck"],axis=1,inplace=True)
```

[7]:
```
df
```

[7]:
```
     survived  pclass     sex   age  sibsp  parch     fare embarked   class  \
0           0       3    male  22.0      1      0   7.2500        S   Third
1           1       1  female  38.0      1      0  71.2833        C   First
2           1       3  female  26.0      0      0   7.9250        S   Third
3           1       1  female  35.0      1      0  53.1000        S   First
4           0       3    male  35.0      0      0   8.0500        S   Third
..        …       …       …     …      …      …        …        …       …
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
888         0       3  female   NaN      1      2  23.4500        S   Third
889         1       1    male  26.0      0      0  30.0000        C   First
890         0       3    male  32.0      0      0   7.7500        Q   Third

       who  adult_male  embark_town alive  alone
0      man        True  Southampton    no  False
1    woman       False    Cherbourg   yes  False
2    woman       False  Southampton   yes   True
3    woman       False  Southampton   yes  False
4      man        True  Southampton    no   True
..     …           …          …     …      …
886    man        True  Southampton    no   True
887  woman       False  Southampton   yes   True
888  woman       False  Southampton    no  False
889    man        True    Cherbourg   yes   True
890    man        True   Queenstown    no   True

[891 rows x 14 columns]
```

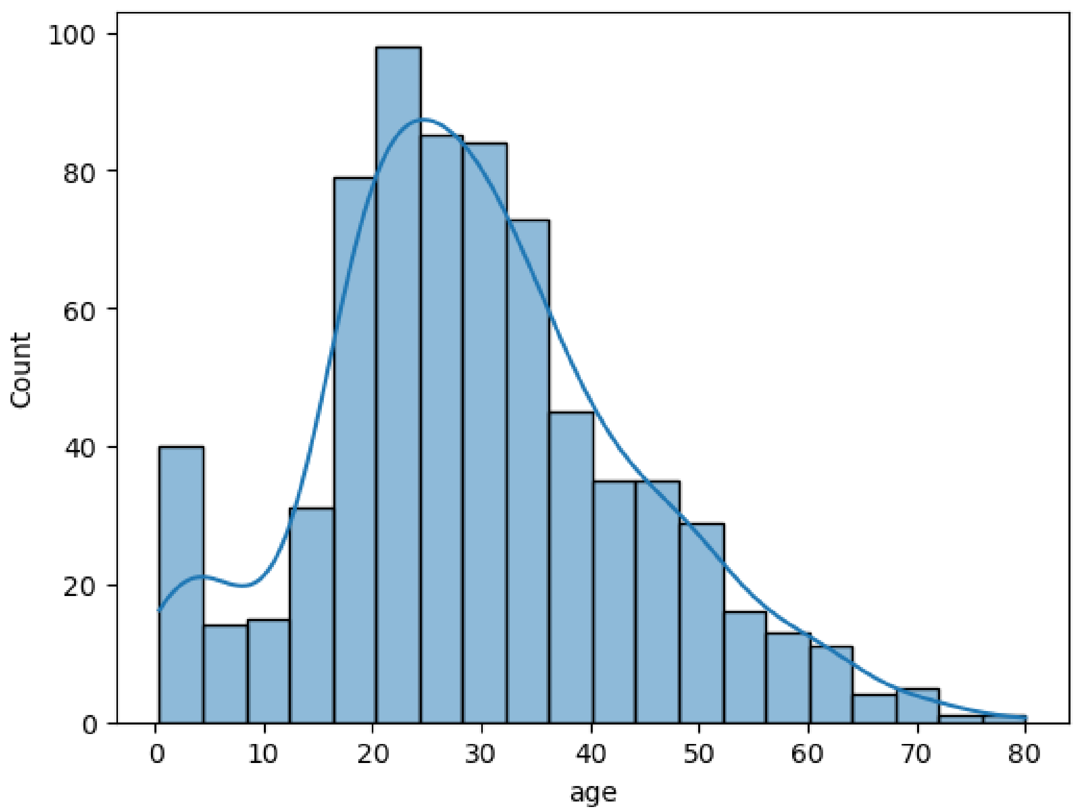
[8]:
```
df.isna().sum()
```

[8]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         0
class            0
who              0
adult_male       0
embark_town      2
alive            0
```

```
alone             0
dtype: int64
```

[9]:
```
sns.histplot(df["age"],kde=True)
```

[9]:
```
<AxesSubplot: xlabel='age', ylabel='Count'>
```



[10]:
```
df["age"].fillna(df["age"].mean(),inplace=True)
```

[11]:
```
df
```

[11]:
```
     survived  pclass     sex        age  sibsp  parch     fare embarked  \
0           0       3    male  22.000000      1      0   7.2500        S
1           1       1  female  38.000000      1      0  71.2833        C
2           1       3  female  26.000000      0      0   7.9250        S
3           1       1  female  35.000000      1      0  53.1000        S
4           0       3    male  35.000000      0      0   8.0500        S
..        …       …     …        …     …      …       …        …
886         0       2    male  27.000000      0      0  13.0000        S
887         1       1  female  19.000000      0      0  30.0000        S
888         0       3  female  29.699118      1      2  23.4500        S
```

```
889         1       1    male  26.000000      0      0  30.0000        C
890         0       3    male  32.000000      0      0   7.7500        Q

      class    who  adult_male  embark_town alive  alone
0     Third    man        True  Southampton    no  False
1     First  woman       False    Cherbourg   yes  False
2     Third  woman       False  Southampton   yes   True
3     First  woman       False  Southampton   yes  False
4     Third    man        True  Southampton    no   True
..      …      …         …          …    …     …
886  Second    man        True  Southampton    no   True
887   First  woman       False  Southampton   yes   True
888   Third  woman       False  Southampton    no  False
889   First    man        True    Cherbourg   yes   True
890   Third    man        True   Queenstown    no   True

[891 rows x 14 columns]
```

[12]:
```
df.isna().sum()
```

[12]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       2
class          0
who            0
adult_male     0
embark_town    2
alive          0
alone          0
dtype: int64
```

[13]:
```
median=df["embarked"].notna().mode()[0]
```

[14]:
```
df["embarked"].fillna(median,inplace=True)
```

[15]:
```
df.isna().sum()
```

[15]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
```

```
parch          0
fare           0
embarked       0
class          0
who            0
adult_male     0
embark_town    2
alive          0
alone          0
dtype: int64
```

[16]:
```
df.embark_town.fillna("Missing",inplace=True)
```

[17]:
```
df.isna().sum()
```

[17]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       0
class          0
who            0
adult_male     0
embark_town    0
alive          0
alone          0
dtype: int64
```

[18]:
```
df
```

[18]:
```
     survived  pclass     sex        age  sibsp  parch     fare embarked  \
0           0       3    male  22.000000      1      0   7.2500        S
1           1       1  female  38.000000      1      0  71.2833        C
2           1       3  female  26.000000      0      0   7.9250        S
3           1       1  female  35.000000      1      0  53.1000        S
4           0       3    male  35.000000      0      0   8.0500        S
..        …       …       …          …      …      …        …
886         0       2    male  27.000000      0      0  13.0000        S
887         1       1  female  19.000000      0      0  30.0000        S
888         0       3  female  29.699118      1      2  23.4500        S
889         1       1    male  26.000000      0      0  30.0000        C
890         0       3    male  32.000000      0      0   7.7500        Q

      class    who  adult_male  embark_town alive  alone
```

```
0     Third    man        True  Southampton    no  False
1     First  woman       False    Cherbourg   yes  False
2     Third  woman       False  Southampton   yes   True
3     First  woman       False  Southampton   yes  False
4     Third    man        True  Southampton    no   True
..      …      …         …           …     …     …
886  Second    man        True  Southampton    no   True
887   First  woman       False  Southampton   yes   True
888   Third  woman       False  Southampton    no  False
889   First    man        True    Cherbourg   yes   True
890   Third    man        True   Queenstown    no   True

[891 rows x 14 columns]
```

2 Handling imbalanced Data

2.1 Making Data

[19]:
```python
import numpy as np
import pandas as pd

# Set the random seed for reproducibility
np.random.seed(123)

# Create a dataframe with two classes
n_samples = 1000
class_0_ratio = 0.9
n_class_0 = int(n_samples * class_0_ratio)
n_class_1 = n_samples - n_class_0

## CREATE MY DATAFRAME WITH IMBALANCED DATASET
class_0 = pd.DataFrame({
    'feature_1': np.random.normal(loc=0, scale=1, size=n_class_0),
    'feature_2': np.random.normal(loc=0, scale=1, size=n_class_0),
    'target': [0] * n_class_0
})

class_1 = pd.DataFrame({
    'feature_1': np.random.normal(loc=2, scale=1, size=n_class_1),
    'feature_2': np.random.normal(loc=2, scale=1, size=n_class_1),
    'target': [1] * n_class_1
})

df=pd.concat([class_0,class_1]).reset_index(drop=True)
```

[20]:
```python
df
```

[20]:
```
     feature_1  feature_2  target
0    -1.085631   0.551302       0
1     0.997345   0.419589       0
2     0.282978   1.815652       0
3    -1.506295  -0.252750       0
4    -0.578600  -0.292004       0
..         …          …       …
995   1.376371   2.845701       1
996   2.239810   0.880077       1
997   1.131760   1.640703       1
998   2.902006   0.390305       1
999   2.697490   2.013570       1

[1000 rows x 3 columns]
```

[21]:
```python
df["target"].value_counts()
```

[21]:
```
0    900
1    100
Name: target, dtype: int64
```

[22]:
```python
major=df[df["target"]==0]
minor=df[df["target"]==1]
```

[23]:
```python
major.shape,minor.shape
```

[23]:
```
((900, 3), (100, 3))
```

[24]:
```python
from sklearn.utils import resample
minor2=resample(minor,replace=True,n_samples=len(major),random_state=10)
```

[25]:
```python
minor2
```

[25]:
```
     feature_1  feature_2  target
909   3.239635   1.361938       1
915   3.519471  -0.233905       1
964   2.397060   0.740228       1
928   1.868135   1.026563       1
989   3.013493   2.047240       1
..         …          …       …
936   3.727988   3.468919       1
928   1.868135   1.026563       1
947   1.402209   2.775845       1
919   1.804892   2.842652       1
902   1.795683   1.803557       1

[900 rows x 3 columns]
```

[26]:
```
minor2.shape
```

[26]: (900, 3)

[27]:
```
major2=resample(major,replace=False,n_samples=len(minor),random_state=10)
```

[28]:
```
major2
```

[28]:
```
     feature_1  feature_2  target
437  -1.639397   0.273073       0
131  -1.100043   1.191189       0
633   0.600571   0.627744       0
195  -3.231055  -1.725890       0
230  -1.600441  -0.304086       0
..         …          …      …
235  -0.434167  -0.265576       0
192   0.199582  -0.096391       0
775   0.048109  -0.805562       0
718   0.301290   0.907483       0
769  -1.094273   0.639969       0

[100 rows x 3 columns]
```

[29]:
```
minor.shape,major2.shape
```

[29]: ((100, 3), (100, 3))

[30]:
```
major.shape,minor2.shape
```

[30]: ((900, 3), (900, 3))

3 SMOTE

[31]:
```
!pip install imblearn
```

```
Requirement already satisfied: imblearn in /opt/conda/lib/python3.10/site-
packages (0.0)
Requirement already satisfied: imbalanced-learn in
/opt/conda/lib/python3.10/site-packages (from imblearn) (0.10.1)
Requirement already satisfied: scipy>=1.3.2 in /opt/conda/lib/python3.10/site-
packages (from imbalanced-learn->imblearn) (1.9.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from imbalanced-learn->imblearn)
(3.1.0)
Requirement already satisfied: scikit-learn>=1.0.2 in
/opt/conda/lib/python3.10/site-packages (from imbalanced-learn->imblearn)
(1.2.0)
```

```
Requirement already satisfied: joblib>=1.1.1 in /opt/conda/lib/python3.10/site-
packages (from imbalanced-learn->imblearn) (1.2.0)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/lib/python3.10/site-
packages (from imbalanced-learn->imblearn) (1.23.5)
```

[32]:
```
from imblearn.over_sampling import SMOTE
```

[33]:
```
ans=SMOTE()
```

[34]:
```
df["target"].value_counts()
```

[34]:
```
0    900
1    100
Name: target, dtype: int64
```

[35]:
```
x,y=ans.fit_resample(df[["feature_1","feature_2"]],df["target"])
```

[36]:
```
y
```

[36]:
```
0       0
1       0
2       0
3       0
4       0
       ..
1795    1
1796    1
1797    1
1798    1
1799    1
Name: target, Length: 1800, dtype: int64
```

[37]:
```
x,y
```

[37]:
```
(      feature_1  feature_2
 0     -1.085631   0.551302
 1      0.997345   0.419589
 2      0.282978   1.815652
 3     -1.506295  -0.252750
 4     -0.578600  -0.292004
 …           …          …
 1795   1.090655   2.299404
 1796   1.894705   3.051485
 1797   2.391335   3.093234
 1798   2.392714   0.742123
 1799   2.276174   2.566088
```

```
[1800 rows x 2 columns],
0       0
1       0
2       0
3       0
4       0
       ..
1795    1
1796    1
1797    1
1798    1
1799    1
Name: target, Length: 1800, dtype: int64)
```

[38]:
```
import pandas as pd
```

[39]:
```
pd.DataFrame(x)
pd.DataFrame(y)
df=pd.concat([x,y],axis=1)
```

[40]:
```
df
```

[40]:
```
      feature_1  feature_2  target
0     -1.085631   0.551302       0
1      0.997345   0.419589       0
2      0.282978   1.815652       0
3     -1.506295  -0.252750       0
4     -0.578600  -0.292004       0
…           …         …      …
1795   1.090655   2.299404       1
1796   1.894705   3.051485       1
1797   2.391335   3.093234       1
1798   2.392714   0.742123       1
1799   2.276174   2.566088       1

[1800 rows x 3 columns]
```

[41]:
```
df["target"].value_counts()
```

[41]:
```
0    900
1    900
Name: target, dtype: int64
```

4 Interpolation

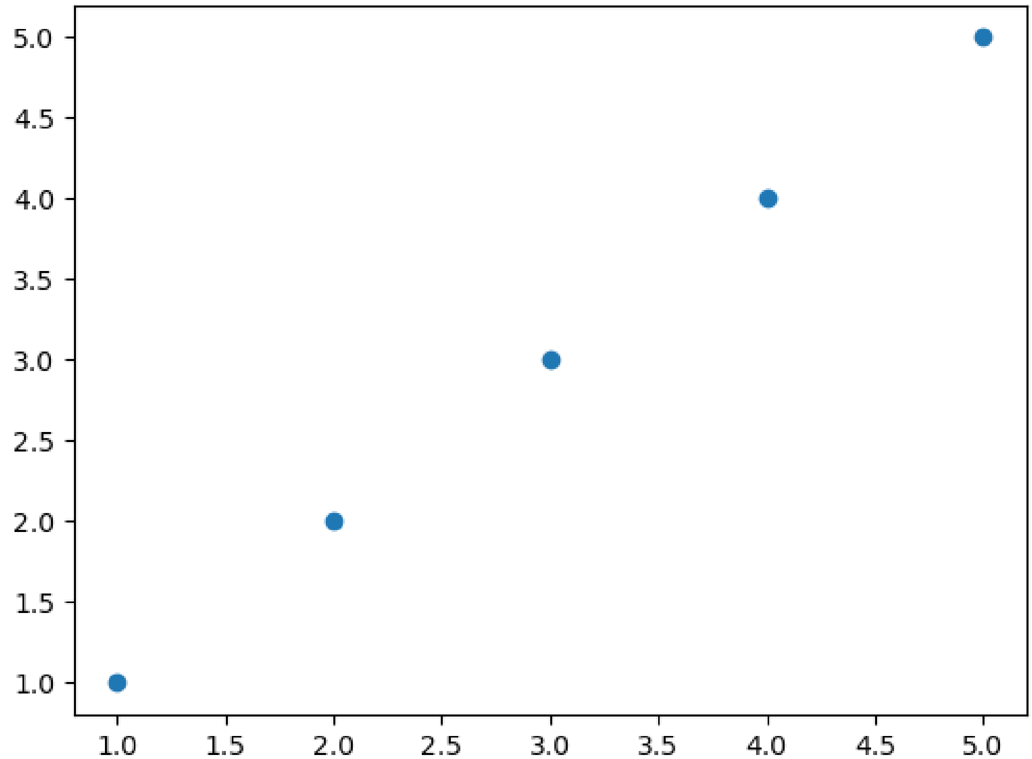
4.1 Linear Interpolate

[42]:
```
x=np.array([1,2,3,4,5])
y=np.array([1,2,3,4,5])
```

[43]:
```
import matplotlib.pyplot as plt
```

[44]:
```
plt.scatter(x,y)
```

[44]:
```
<matplotlib.collections.PathCollection at 0x7f6c8aa962c0>
```



[45]:
```
x_new=np.linspace(1,5,10)
```

[46]:
```
x_new
```

[46]:
```
array([1.        , 1.44444444, 1.88888889, 2.33333333, 2.77777778,
       3.22222222, 3.66666667, 4.11111111, 4.55555556, 5.        ])
```

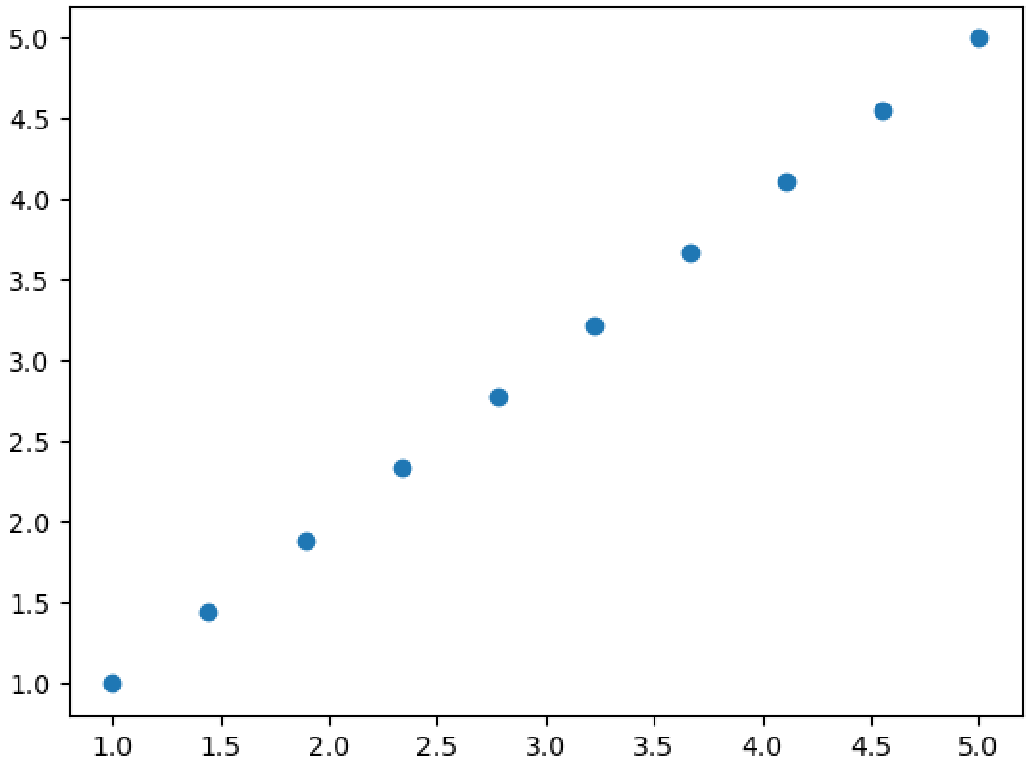
[47]:
```
y_new=np.interp(x_new,x,y)
```

[48]:
```
y_new
```

[48]:
```
array([1.        , 1.44444444, 1.88888889, 2.33333333, 2.77777778,
       3.22222222, 3.66666667, 4.11111111, 4.55555556, 5.        ])
```

[49]:
```
plt.scatter(x_new,y_new)
```

[49]:
```
<matplotlib.collections.PathCollection at 0x7f6c8278f340>
```

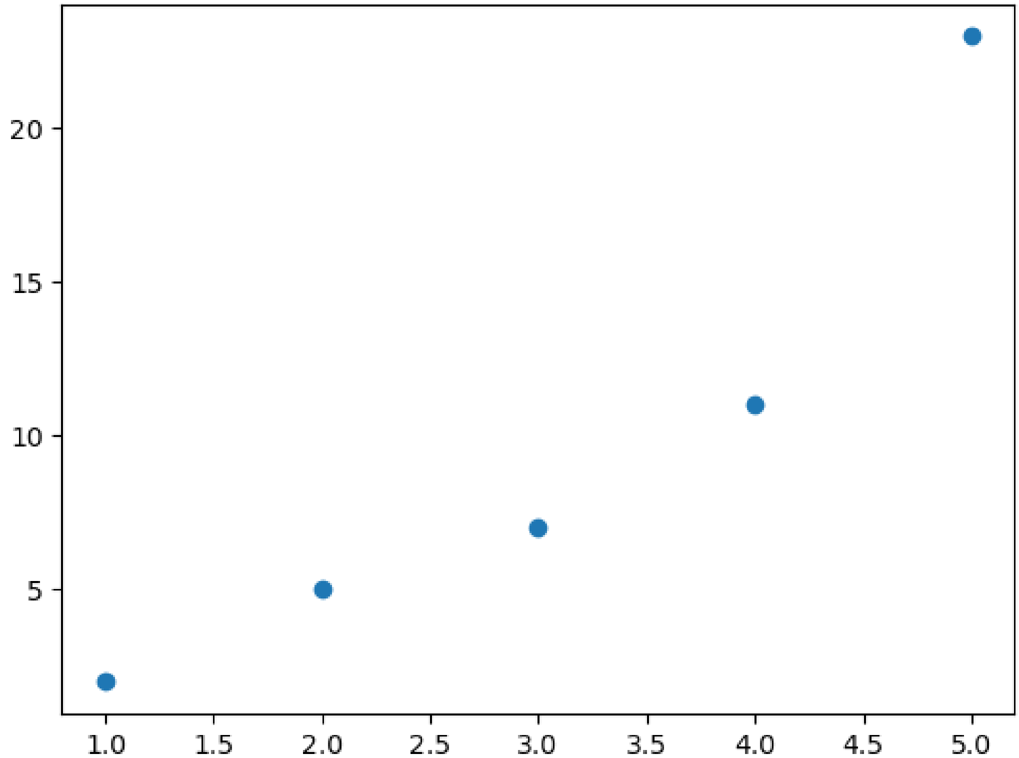


4.2 Cubic Interpolate

[50]:
```
a=np.array([1,2,3,4,5])
b=np.array([2,5,7,11,23])
```

[51]:
```
plt.scatter(a,b)
```

[51]:
```
<matplotlib.collections.PathCollection at 0x7f6c82617910>
```



[52]:
```
from scipy.interpolate import interp1d
```

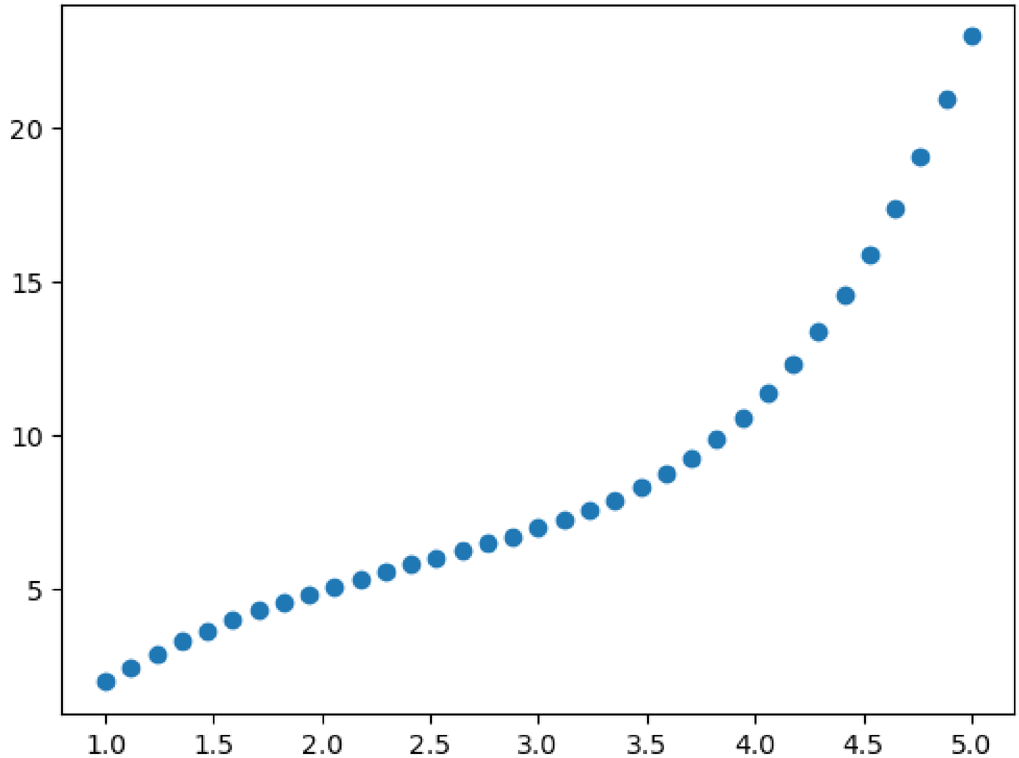
[53]:
```
f=interp1d(a,b,kind="cubic")
```

[54]:
```
a_new=np.linspace(1,5,35)
```

[55]:
```
b_new=f(a_new)
```

[56]:
```
plt.scatter(a_new,b_new)
```

[56]:
```
<matplotlib.collections.PathCollection at 0x7f6c826aa050>
```



5 Percentiles and Outliers

[57]:
```
marks=[11,23,56,80,90,90,100,100,100,109,112,132,148,179,230]
```

[58]:
```
minimum,q1,median,q3,maximum=np.quantile(marks,[0,0.25,0.50,0.75,1])
```

[59]:
```
minimum,q1,median,q3,maximum
```

[59]: (11.0, 85.0, 100.0, 122.0, 230.0)

[60]:
```
IQR=q3-q1
IQR
```

[60]: 37.0

[61]:
```
lower_fence=IQR-(1.5*q1)
higher_fence=IQR+(1.5*q3)
```

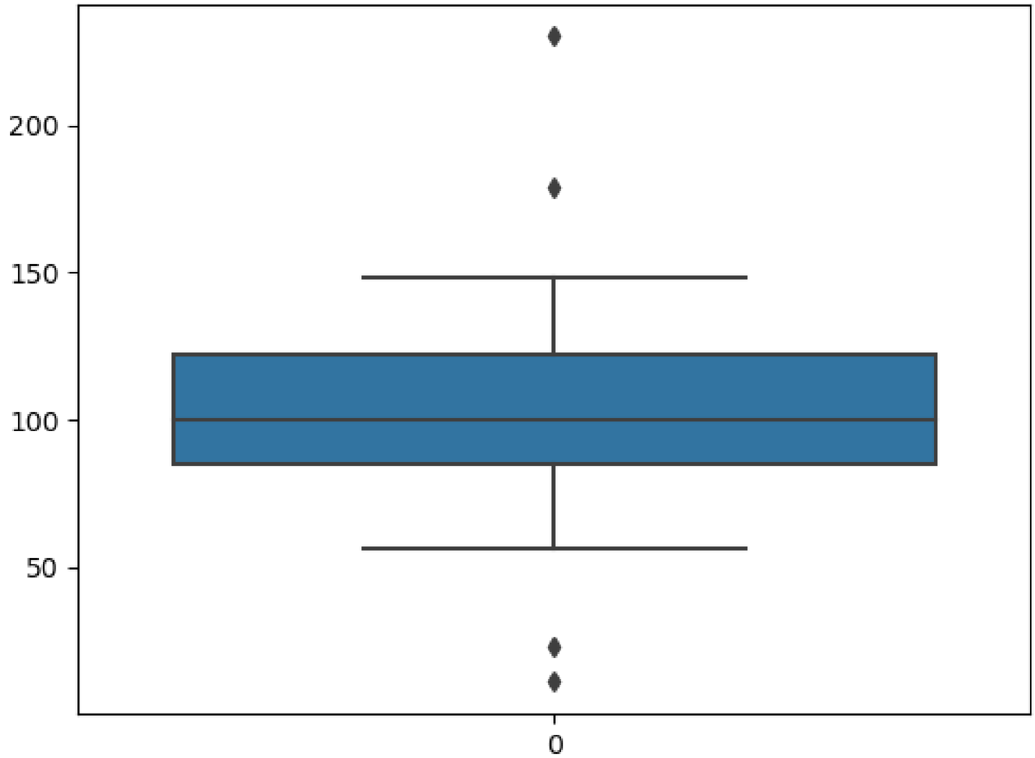
[62]:
```
lower_fence,higher_fence
```

[62]: (-90.5, 220.0)

[63]:

```
sns.boxplot(marks)
```

[63]: <AxesSubplot: >



[ ]: