IMPORTING THE LIBRARIES

In [29]:

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

In [4]:

```
df = pd.read_csv("C:/Users/91944/OneDrive/Documents/AI smary/Titanic-Dataset.
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	(
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	

In [7]:
```
df.tail()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	Nal
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B4
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	Nal
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C14
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	Nal

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [9]:
```
df.describe()
```

Out[9]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Null values

In [10]:
```
df.isnull().any()
```

Out[10]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [16]:
```
mean = df["Age"].mean()
df["Age"] = df["Age"].fillna(mean)
df["Age"].tail()
```

Out[16]:
```
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, dtype: float64
```

In [17]:
```
df["Age"].isnull().sum()
```

Out[17]:
```
0
```

In [18]:
```
E_mode = df["Embarked"].mode()
df["Embarked"] = df["Embarked"].fillna(E_mode[0])
df["Embarked"].isnull().sum()
```

Out[18]:
```
0
```

In [20]:
```
Cabin_mode=df["Cabin"].mode()
df["Cabin"]
```

Out[20]:
```
0       NaN
1       C85
2       NaN
3      C123
4       NaN
       ...
886     NaN
887     B42
888     NaN
889    C148
890     NaN
Name: Cabin, Length: 891, dtype: object
```

In [21]:
```
Cabin_mode
```

Out[21]:
```
0        B96 B98
1    C23 C25 C27
2             G6
dtype: object
```

In [24]:

```
df["Cabin"] = df["Cabin"].fillna(Cabin_mode[2])
df["Cabin"].isnull().sum()
df["Cabin"]
```

Out[24]:

```
0        G6
1       C85
2        G6
3      C123
4        G6
       ... 
886      G6
887     B42
888      G6
889    C148
890      G6
Name: Cabin, Length: 891, dtype: object
```

In [25]:

```
df.isnull().sum()
```

Out[25]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

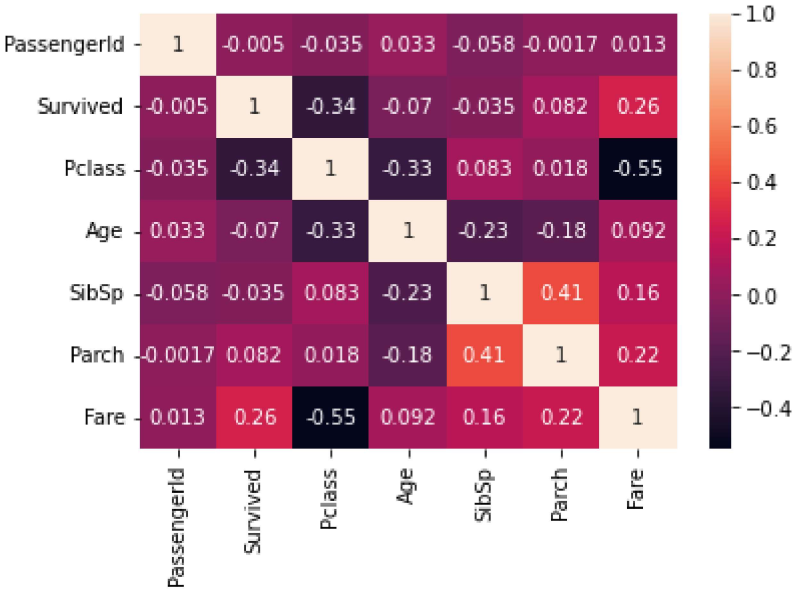
Data Vizualization

In [27]:

```
corr = df.corr()
```

```
In [30]: sns.heatmap(corr, annot = True)
```

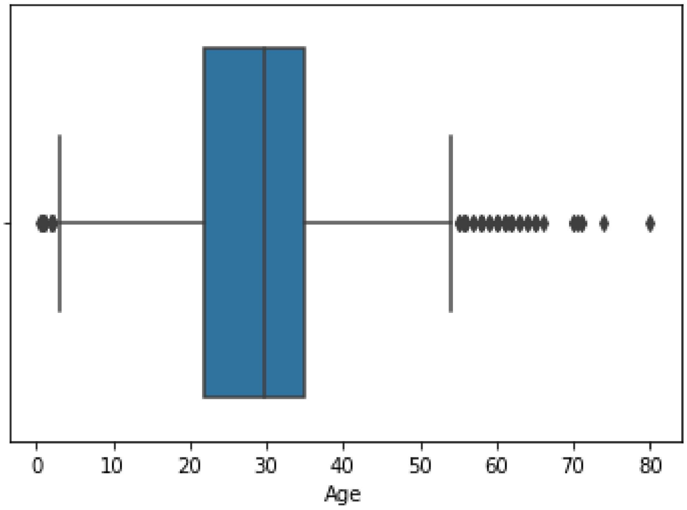
```
Out[30]: <AxesSubplot:>
```



```
In [31]: sns.boxplot(df["Age"])
```

```
C:\Users\91944\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[31]: <AxesSubplot:xlabel='Age'>
```



In [32]:

```
Age_q1 = df.Age.quantile(0.25)
Age_q2 = df.Age.quantile(0.75)
print(Age_q1)
print(Age_q2)
```

```
22.0
35.0
```

In [34]:

```
IQR_Age=Age_q2-Age_q1
IQR_Age
```

Out[34]:

```
13.0
```

In [35]:

```
upperlimit_Age=Age_q2+1.5*IQR_Age
upperlimit_Age
```

Out[35]:

```
54.5
```

In [36]:

```
lower_limit_Age = Age_q1-1.5*IQR_Age
lower_limit_Age
```

Out[36]:

```
2.5
```

In [38]:

```
median_Age=df["Age"].median()
median_Age
```

Out[38]:

```
29.69911764705882
```

In [41]:

```
df["Age"]=np.where(df["Age"]>upperlimit_Age,median_Age,df["Age"])
(df["Age"]>54.5).sum()
```

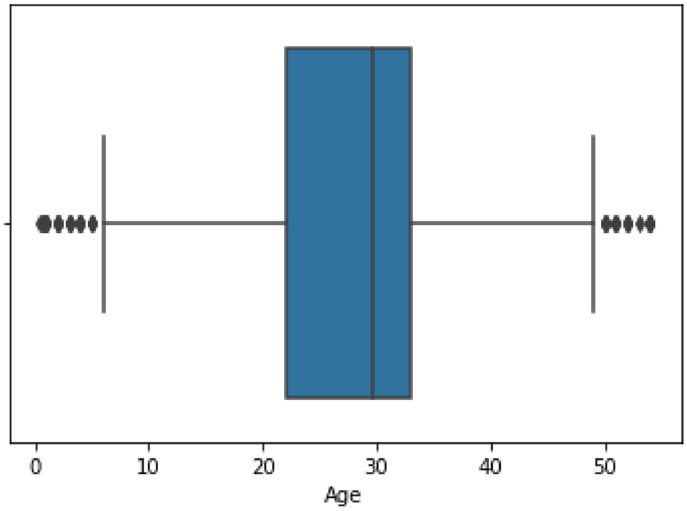
Out[41]:

```
0
```

In [43]:
```
sns.boxplot(df["Age"])
```

```
C:\Users\91944\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

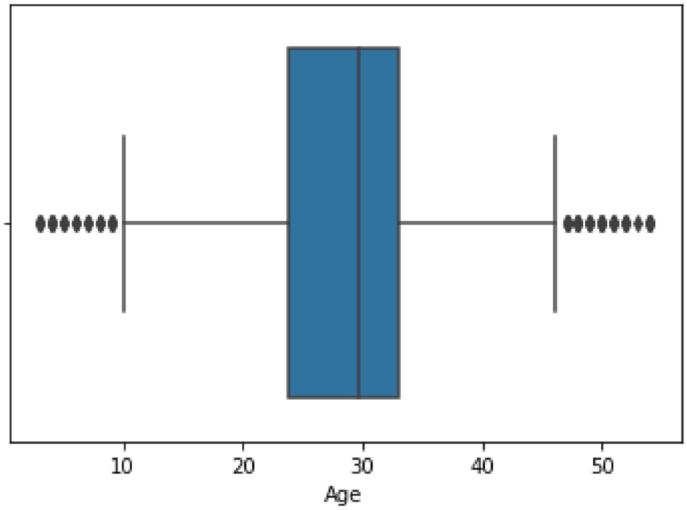
Out[43]:
```
<AxesSubplot:xlabel='Age'>
```



In [45]:
```
df["Age"]=np.where(df["Age"]<lower_limit_Age,median_Age,df["Age"])
sns.boxplot(df["Age"])
```

```
C:\Users\91944\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

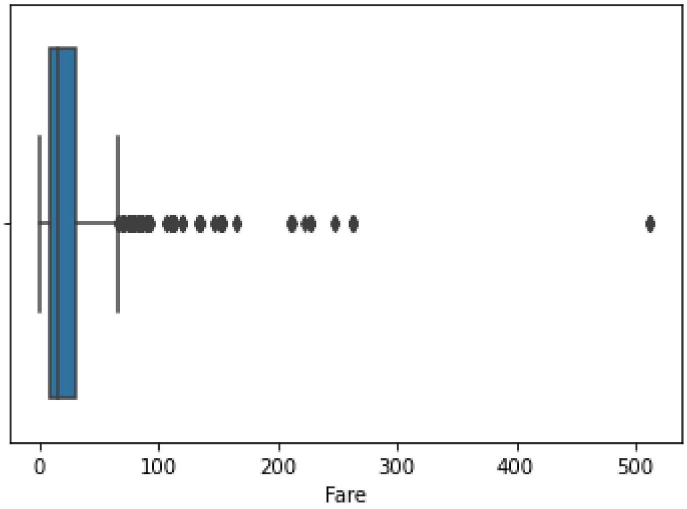
Out[45]:
```
<AxesSubplot:xlabel='Age'>
```



In [46]:
```
sns.boxplot(df["Fare"])
```

```
C:\Users\91944\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[46]:
```
<AxesSubplot:xlabel='Fare'>
```



In [48]:
```
Fare_q1 = df.Fare.quantile(0.25)
Fare_q2 = df.Fare.quantile(0.75)
print(Fare_q1)
print(Fare_q2)
```

```
7.9104
31.0
```

In [49]:
```
IQR_Fare=Fare_q2-Fare_q1
IQR_Fare
```

Out[49]:
```
23.0896
```

In [50]:
```
upperlimit_Fare=Fare_q2+1.5*IQR_Fare
upperlimit_Fare
```

Out[50]:
```
65.6344
```

In [51]:
```
lower_limit_Fare = Fare_q1-1.5*IQR_Fare
lower_limit_Fare
```

Out[51]:
```
-26.724
```

In [52]:

```
median_Fare=df["Fare"].median()
median_Fare
```

Out[52]:

```
14.4542
```

In [54]:

```
df['Fare'] = np.where(
 (df['Fare'] > upperlimit_Fare),
 median_Fare,
 df['Fare']
)
```

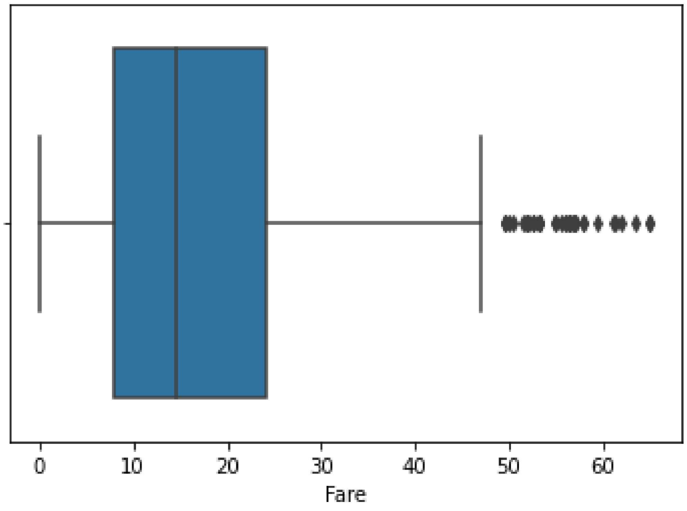
In [55]:

```
sns.boxplot(df["Fare"])
```

```
C:\Users\91944\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[55]:

```
<AxesSubplot:xlabel='Fare'>
```



In [56]:

```
(df["Fare"]>65).sum()
```

Out[56]:

```
0
```

In [58]:

```
df.drop(['Name'],axis=1,inplace=True)
```

In [59]:
```
df
```

Out[59]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	male	22.000000	1	0	A/5 21171	7.2500	G
1	2	1	1	female	38.000000	1	0	PC 17599	14.4542	C8
2	3	1	3	female	26.000000	0	0	STON/O2. 3101282	7.9250	G
3	4	1	1	female	35.000000	1	0	113803	53.1000	C12
4	5	0	3	male	35.000000	0	0	373450	8.0500	G
...	...	...	...	...	...	...	...	...	...	..
886	887	0	2	male	27.000000	0	0	211536	13.0000	G
887	888	1	1	female	19.000000	0	0	112053	30.0000	B4
888	889	0	3	female	29.699118	1	2	W./C. 6607	23.4500	G
889	890	1	1	male	26.000000	0	0	111369	30.0000	C14
890	891	0	3	male	32.000000	0	0	370376	7.7500	G

891 rows × 11 columns

In [60]:
```
df.drop(['Ticket'],axis=1,inplace=True)
df
```

Out[60]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarke
0	1	0	3	male	22.000000	1	0	7.2500	G6	
1	2	1	1	female	38.000000	1	0	14.4542	C85	
2	3	1	3	female	26.000000	0	0	7.9250	G6	
3	4	1	1	female	35.000000	1	0	53.1000	C123	
4	5	0	3	male	35.000000	0	0	8.0500	G6	
...	...	...	...	...	...	...	...	...	...	.
886	887	0	2	male	27.000000	0	0	13.0000	G6	
887	888	1	1	female	19.000000	0	0	30.0000	B42	
888	889	0	3	female	29.699118	1	2	23.4500	G6	
889	890	1	1	male	26.000000	0	0	30.0000	C148	
890	891	0	3	male	32.000000	0	0	7.7500	G6	

891 rows × 10 columns

In [61]:
```
df.drop(["PassengerId"],axis=1,inplace=True)
df
```

Out[61]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	male	22.000000	1	0	7.2500	G6	S
1	1	1	female	38.000000	1	0	14.4542	C85	C
2	1	3	female	26.000000	0	0	7.9250	G6	S
3	1	1	female	35.000000	1	0	53.1000	C123	S
4	0	3	male	35.000000	0	0	8.0500	G6	S
...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.000000	0	0	13.0000	G6	S
887	1	1	female	19.000000	0	0	30.0000	B42	S
888	0	3	female	29.699118	1	2	23.4500	G6	S
889	1	1	male	26.000000	0	0	30.0000	C148	C
890	0	3	male	32.000000	0	0	7.7500	G6	Q

891 rows × 9 columns

In [62]:
```
df.drop(["Cabin"],axis=1,inplace=True)
df
```

Out[62]:

	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	14.4542	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

891 rows × 8 columns

Splitting the data

In [63]:

```
y=df["Survived"]
y.head()
```

Out[63]:

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

Encoding

In [64]:

```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()
df["Sex"]=le.fit_transform(df["Sex"])
df["Sex"]
```

Out[64]:

```
0      1
1      0
2      0
3      0
4      1
      ..
886    1
887    0
888    0
889    1
890    1
Name: Sex, Length: 891, dtype: int32
```

In [67]:

```
df["Embarked"]=le.fit_transform(df["Embarked"])
```

In [68]:

```
df.head()
```

Out[68]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	1	22.0	1	0	7.2500	2
1	1	1	0	38.0	1	0	14.4542	0
2	1	3	0	26.0	0	0	7.9250	2
3	1	1	0	35.0	1	0	53.1000	2
4	0	3	1	35.0	0	0	8.0500	2

In [70]:
```
df["Pclass"].nunique()
df["Pclass"].unique()
```

Out[70]:
```
array([3, 1, 2], dtype=int64)
```

In [72]:
```
df["Sex"].unique()
```

Out[72]:
```
array([1, 0])
```

In [73]:
```
df["Embarked"].unique()
```

Out[73]:
```
array([2, 0, 1])
```

Test and Train Data

In [75]:
```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(df,y,test_size=0.3,random_stat
```

In [76]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[76]:
```
((623, 8), (268, 8), (623,), (268,))
```

Feature Scaling

In [77]:
```
from sklearn.preprocessing import StandardScaler

sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_train
```

Out[77]:
```
array([[ 1.25474307, -1.5325562 ,  0.72592065, ..., -0.47299765,
         0.67925137,  0.56710989],
       [ 1.25474307, -1.5325562 , -1.37756104, ..., -0.47299765,
        -0.26059483, -2.03075381],
       [-0.79697591,  0.84844757,  0.72592065, ...,  1.93253327,
         2.26045064,  0.56710989],
       ...,
       [-0.79697591,  0.84844757,  0.72592065, ..., -0.47299765,
        -0.78281017, -0.73182196],
       [ 1.25474307,  0.84844757, -1.37756104, ..., -0.47299765,
        -0.03170555,  0.56710989],
       [-0.79697591, -0.34205431,  0.72592065, ...,  0.72976781,
         1.64661898,  0.56710989]])
```

In [80]:

```
x_test=sc.fit_transform(x_test)

x_test
```

Out[80]:

```
array([[-0.77151675,  0.77963055,  0.76537495, ..., -0.47809977,
        -0.15813988, -1.76531134],
       [-0.77151675,  0.77963055,  0.76537495, ..., -0.47809977,
        -0.72165412,  0.63014911],
       [-0.77151675,  0.77963055,  0.76537495, ...,  0.87064484,
         1.03823178, -0.56758111],
       ...,
       [-0.77151675,  0.77963055,  0.76537495, ..., -0.47809977,
        -0.15847431, -1.76531134],
       [ 1.29614814,  0.77963055, -1.30654916, ..., -0.47809977,
        -0.72607524,  0.63014911],
       [-0.77151675, -1.64991582,  0.76537495, ..., -0.47809977,
         0.92369033, -1.76531134]])
```