In [45]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.impute import SimpleImputer
from sklearn.metrics import mean_squared_error
```

In [2]:

```
data = pd.read_csv("titanic.csv")
```

In [3]:

```
data.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN

In [4]:

```
data.shape
```

Out[4]:

```
(418, 12)
```

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [6]:

```
data.describe()
```

Out[6]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [7]:

```
data.head()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN

In [8]:

```
data = data.drop(["PassengerId" , "Name" , "Ticket"],axis = 1)
```

In [9]:

```
data.head()
```

Out[9]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	male	34.5	0	0	7.8292	NaN	Q
1	1	3	female	47.0	1	0	7.0000	NaN	S
2	0	2	male	62.0	0	0	9.6875	NaN	Q
3	0	3	male	27.0	0	0	8.6625	NaN	S
4	1	3	female	22.0	1	1	12.2875	NaN	S

In [10]:

```
data.isna().sum()
```

Out[10]:

```
Survived      0
Pclass        0
Sex           0
Age          86
SibSp         0
Parch         0
Fare          1
Cabin       327
Embarked      0
dtype: int64
```

In [11]:

```
imputer = SimpleImputer(missing_values = np.nan , strategy = "mean" )
imputer.fit(data.iloc[:,3:4])
data.iloc[:,3:4] = imputer.fit_transform(data.iloc[:,3:4].values)
```

In [12]:

```
data.isna().sum()
```

Out[12]:

```
Survived      0
Pclass        0
Sex           0
Age           0
SibSp         0
Parch         0
Fare          1
Cabin       327
Embarked      0
dtype: int64
```

In [13]:

```
data.dropna()
```

Out[13]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
12	1	1	female	23.0	1	0	82.2667	B45	S
14	1	1	female	47.0	1	0	61.1750	E31	S
24	1	1	female	48.0	1	3	262.3750	B57 B59 B63 B66	C
26	1	1	female	22.0	0	1	61.9792	B36	C
28	0	1	male	41.0	0	0	30.5000	A21	S
...	...	...	...	...	...	...	...	...	...
404	0	1	male	43.0	1	0	27.7208	D40	C
405	0	2	male	20.0	0	0	13.8625	D38	C
407	0	1	male	50.0	1	1	211.5000	C80	C
411	1	1	female	37.0	1	0	90.0000	C78	Q
414	1	1	female	39.0	0	0	108.9000	C105	C

91 rows × 9 columns

In [14]:

```
data.isna().sum()
```

Out[14]:

```
Survived      0
Pclass        0
Sex           0
Age           0
SibSp         0
Parch         0
Fare          1
Cabin       327
Embarked      0
dtype: int64
```

In [15]:

```
data.head()
```

Out[15]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	male	34.5	0	0	7.8292	NaN	Q
1	1	3	female	47.0	1	0	7.0000	NaN	S
2	0	2	male	62.0	0	0	9.6875	NaN	Q
3	0	3	male	27.0	0	0	8.6625	NaN	S
4	1	3	female	22.0	1	1	12.2875	NaN	S

In [16]:

```
data = data.drop(["Cabin" , "Embarked" ],axis = 1)
```

In [17]:

```
data
```

Out[17]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	3	male	34.50000	0	0	7.8292
1	1	3	female	47.00000	1	0	7.0000
2	0	2	male	62.00000	0	0	9.6875
3	0	3	male	27.00000	0	0	8.6625
4	1	3	female	22.00000	1	1	12.2875
...	...	...	...	...	...	...	...
413	0	3	male	30.27259	0	0	8.0500
414	1	1	female	39.00000	0	0	108.9000
415	0	3	male	38.50000	0	0	7.2500
416	0	3	male	30.27259	0	0	8.0500
417	0	3	male	30.27259	1	1	22.3583

418 rows × 7 columns

In [18]:

```
sex = pd.get_dummies(data["Sex"] , drop_first = True)
sex = sex.astype(int)
data = pd.concat([sex , data] , axis = 1)
```

In [19]:

```
data
```

Out[19]:

	male	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	1	0	3	male	34.50000	0	0	7.8292
1	0	1	3	female	47.00000	1	0	7.0000
2	1	0	2	male	62.00000	0	0	9.6875
3	1	0	3	male	27.00000	0	0	8.6625
4	0	1	3	female	22.00000	1	1	12.2875
...	...	...	...	...	...	...	...	...
413	1	0	3	male	30.27259	0	0	8.0500
414	0	1	1	female	39.00000	0	0	108.9000
415	1	0	3	male	38.50000	0	0	7.2500
416	1	0	3	male	30.27259	0	0	8.0500
417	1	0	3	male	30.27259	1	1	22.3583

418 rows × 8 columns

In [20]:

```
data = data.drop("Sex" , axis = 1)
```

In [21]:

```
data.head()
```

Out[21]:

	male	Survived	Pclass	Age	SibSp	Parch	Fare
0	1	0	3	34.5	0	0	7.8292
1	0	1	3	47.0	1	0	7.0000
2	1	0	2	62.0	0	0	9.6875
3	1	0	3	27.0	0	0	8.6625
4	0	1	3	22.0	1	1	12.2875

In [22]:

```
data.isna().sum()
```

Out[22]:

```
male        0
Survived    0
Pclass      0
Age         0
SibSp       0
Parch       0
Fare        1
dtype: int64
```

In [23]:

```
data = data.dropna()
```

In [24]:

```
data.isna().sum()
```

Out[24]:

```
male        0
Survived    0
Pclass      0
Age         0
SibSp       0
Parch       0
Fare        0
dtype: int64
```

In [25]:

```
x = data.drop("Survived" , axis = 1)
```

In [ ]:

```

```

In [26]:

```
y = data[["Survived"]]
```

In [27]:

```
x
```

Out[27]:

	male	Pclass	Age	SibSp	Parch	Fare
0	1	3	34.50000	0	0	7.8292
1	0	3	47.00000	1	0	7.0000
2	1	2	62.00000	0	0	9.6875
3	1	3	27.00000	0	0	8.6625
4	0	3	22.00000	1	1	12.2875
...	...	...	...	...	...	...
413	1	3	30.27259	0	0	8.0500
414	0	1	39.00000	0	0	108.9000
415	1	3	38.50000	0	0	7.2500
416	1	3	30.27259	0	0	8.0500
417	1	3	30.27259	1	1	22.3583

417 rows × 6 columns

In [28]:

```
y
```

Out[28]:

	Survived
0	0
1	1
2	0
3	0
4	1
...	...
413	0
414	1
415	0
416	0
417	0

417 rows × 1 columns

In [29]:

```
x_train , x_test ,y_train , y_test = train_test_split(x , y , test_size = 0.2 , ran
```

In [30]:

```
x_train
```

Out[30]:

	male	Pclass	Age	SibSp	Parch	Fare
52	0	2	20.00000	2	1	23.0000
316	1	1	57.00000	1	0	146.5208
296	0	2	1.00000	1	2	41.5792
275	0	2	20.00000	1	0	26.0000
64	1	1	13.00000	2	2	262.3750
...	...	...	...	...	...	...
370	1	2	21.00000	1	0	11.5000
321	1	3	25.00000	0	0	7.2292
15	0	2	24.00000	1	0	27.7208
125	0	3	17.00000	0	1	16.1000
266	1	1	30.27259	0	0	0.0000

333 rows × 6 columns

In [31]:

```
x_train.shape
```

Out[31]:

```
(333, 6)
```

In [32]:

```
y_train
```

Out[32]:

	Survived
52	1
316	0
296	1
275	1
64	0
...	...
370	0
321	0
15	1
125	1
266	0

333 rows × 1 columns

In [33]:

```
y_train.shape
```

Out[33]:

```
(333, 1)
```

In [34]:

```
x_test
```

Out[34]:

	male	Pclass	Age	SibSp	Parch	Fare
229	1	2	36.00000	0	0	13.0000
317	1	2	19.00000	0	0	10.5000
282	0	3	30.27259	0	0	7.7500
78	1	2	30.00000	0	0	13.0000
1	0	3	47.00000	1	0	7.0000
...	...	...	...	...	...	...
362	0	2	31.00000	0	0	21.0000
162	0	2	26.00000	0	0	13.5000
319	1	2	22.00000	2	0	31.5000
303	1	3	24.00000	0	0	8.6625
279	1	2	22.00000	0	0	10.5000

84 rows × 6 columns

In [35]:
```
x_test.shape
```

Out[35]:
```
(84, 6)
```

In [36]:
```
y_test
```

Out[36]:

	Survived
229	0
317	0
282	1
78	0
1	1
...	...
362	1
162	1
319	0
303	0
279	0

84 rows × 1 columns

In [37]:
```
y_test.shape
```

Out[37]:
```
(84, 1)
```

In [38]:
```
model = SVR()
```

In [39]:
```
model
```

Out[39]:

```
▾ SVR
SVR()
```

In [40]:

```
model.fit(x_train , y_train)
```

```
C:\Users\godde\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1184: DataC
onversionWarning: A column-vector y was passed when a 1d array was expected. Pleas
e change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[40]:

```
▾ SVR
SVR()
```

In [41]:

```
y_prediction = model.predict(x_test)
```

In [42]:

```
y_prediction[10]
```

Out[42]:

```
0.16017185363186337
```

In [ ]:

In [43]:

```
data
```

Out[43]:

	male	Survived	Pclass	Age	SibSp	Parch	Fare
0	1	0	3	34.50000	0	0	7.8292
1	0	1	3	47.00000	1	0	7.0000
2	1	0	2	62.00000	0	0	9.6875
3	1	0	3	27.00000	0	0	8.6625
4	0	1	3	22.00000	1	1	12.2875
...	...	...	...	...	...	...	...
413	1	0	3	30.27259	0	0	8.0500
414	0	1	1	39.00000	0	0	108.9000
415	1	0	3	38.50000	0	0	7.2500
416	1	0	3	30.27259	0	0	8.0500
417	1	0	3	30.27259	1	1	22.3583

417 rows × 7 columns

In [46]:

```
error = mean_squared_error(y_test , y_prediction)
```

In [47]:

```
error
```

Out[47]:

```
0.32178087547574874
```

In [ ]:

In [ ]:

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```