In [ ]:
```
#Assuming we're working with a software company that's trying to predict a pro
#This score is dependent on four factors:• Average response time (in seconds)
#Number of features in the product.• Number of bugs reported by users.• Traini
```

In [2]:
```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
```

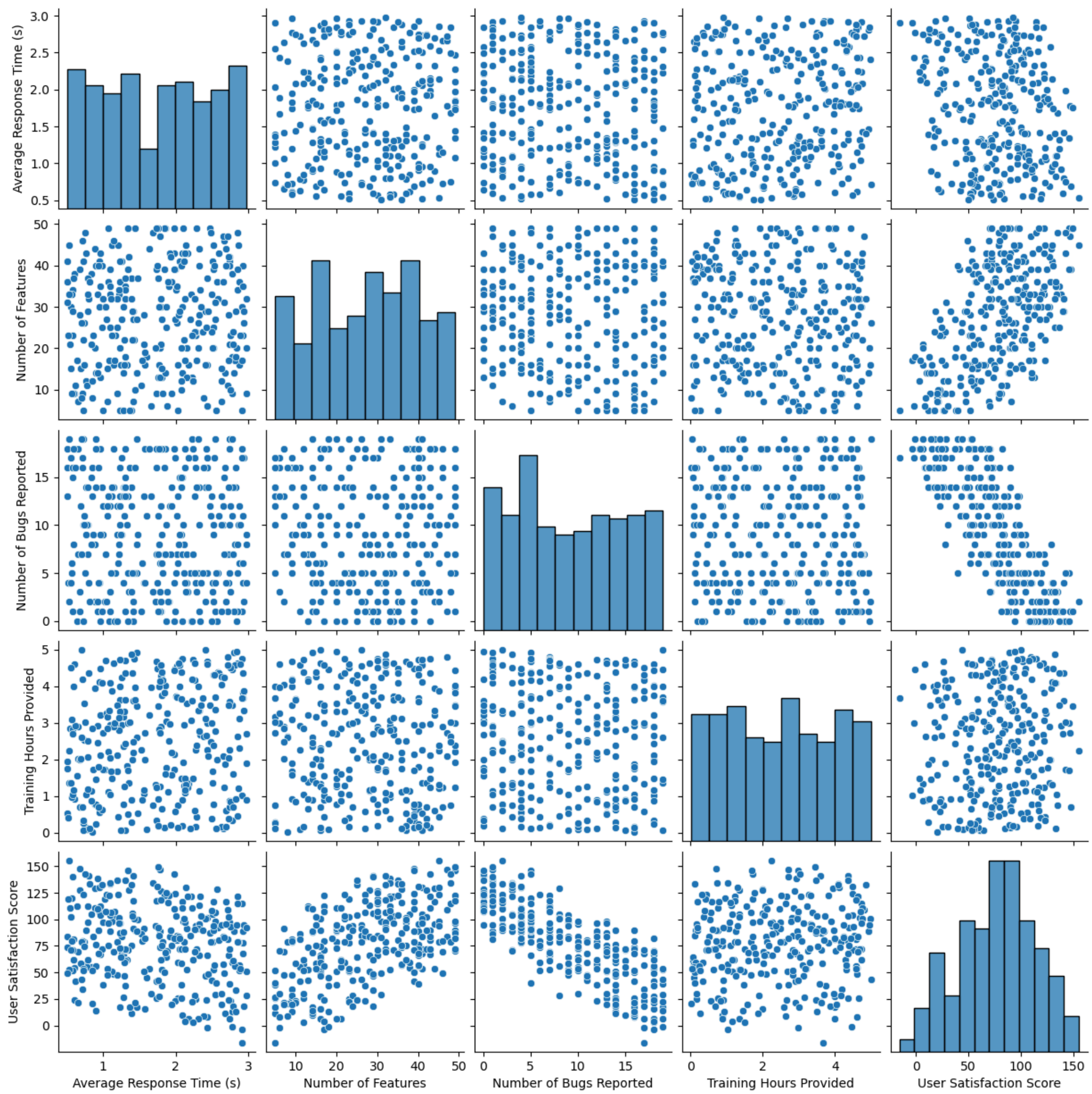
In [5]:
```
Synthetic_app_data = pd.read_csv('Synthetic_app_data (1).csv')
Synthetic_app_data.tail(5)
```

Out[5]:

	Average Response Time (s)	Number of Features	Number of Bugs Reported	Training Hours Provided	User Satisfaction Score
295	1.805608	15	6	4.373508	75.282835
296	2.424984	25	18	2.644686	24.424736
297	1.039553	30	7	4.695338	107.672922
298	2.057226	29	9	3.993916	82.359435
299	0.713369	26	19	4.989671	43.682683

In [6]:
```
sns.pairplot(Synthetic_app_data)
```

Out[6]: <seaborn.axisgrid.PairGrid at 0x1ab90f57cd0>



In [ ]:
```
#4. Clearly state your observations using a markdown cell.
# From the plot we could see that the independent variables "Number of Feature
```

In [37]:
```
#5.
indp_vars=Synthetic_app_data[['Number of Features','Number of Bugs Reported']]
dep_var=Synthetic_app_data['User Satisfaction Score']
```

In [38]:
```
#6.
train_x,test_x,train_y,test_y=train_test_split(indp_vars,dep_var,test_size=0.3
mlr_model=LinearRegression()
mlr_model.fit(train_x,train_y)
pred=mlr_model.predict(test_x)
pred
```

Out[38]:
```
array([117.03067483, 109.7687762 , 102.88465679,  87.64445919,
        38.28312758,  46.99740593, 126.12273239, 100.71587672,
        90.56879769,  48.48894374,  48.82756487, 119.21903393,
        89.81323926, 114.48411555,  64.40638359,  54.9757049 ,
        92.00159837,  66.27570062,  75.68680026, 122.10421434,
        45.18682604,  75.68680026,  99.24391795,  83.30689906,
       126.12273239, 101.81005628, 128.29151246,  11.8016714 ,
        61.87940335,  20.51594975, 121.387814  ,  78.94975989,
        35.41752622,  37.94450645,  10.72707089,   8.53871178,
        43.01804597, 107.59999613,  50.61856572,  65.16194202,
       122.84019373,  76.74182173,  58.63602278, 115.23967397,
        37.92492741,  84.73969974,  94.56773669,  56.82544288,
        98.18889648, 102.88465679,  90.88783977,  86.56985868,
        49.94132347,  54.25930456, 104.69523668,  99.60211812,
        85.137058  ,  98.52751761,  82.94869889,  18.34716968,
       135.55341108,  19.44134924, 126.10315335, 101.09365593,
       106.50581658,  94.17037844,  50.29952364,  90.17143943,
       120.67141366,  99.60211812,  94.88677878, 123.1983939 ,
        72.06564047,  75.70637931,  90.17143943, 105.78941624,
        58.59686469, 107.59999613, 117.76665421,  24.87308893,
       127.91373324,  91.62381916,  53.20428309, 106.16719545,
        77.13917999, 111.93755627, 111.57935609,  21.23235009,
        23.79848841,  62.99316195])
```

In [39]:
```
mse_mlr=mean_squared_error(pred,test_y)
mse_mlr
```

Out[39]:
```
154.311721846551
```

In [31]:
```
#6.Fit an MLR model to the data.

print("Intercept: ", mlr_model.intercept_)
print("Coefficients: \n")
print("Number of Features:",mlr_model.coef_[0])
print("Number of Bugs Reported:",mlr_model.coef_[1])
```

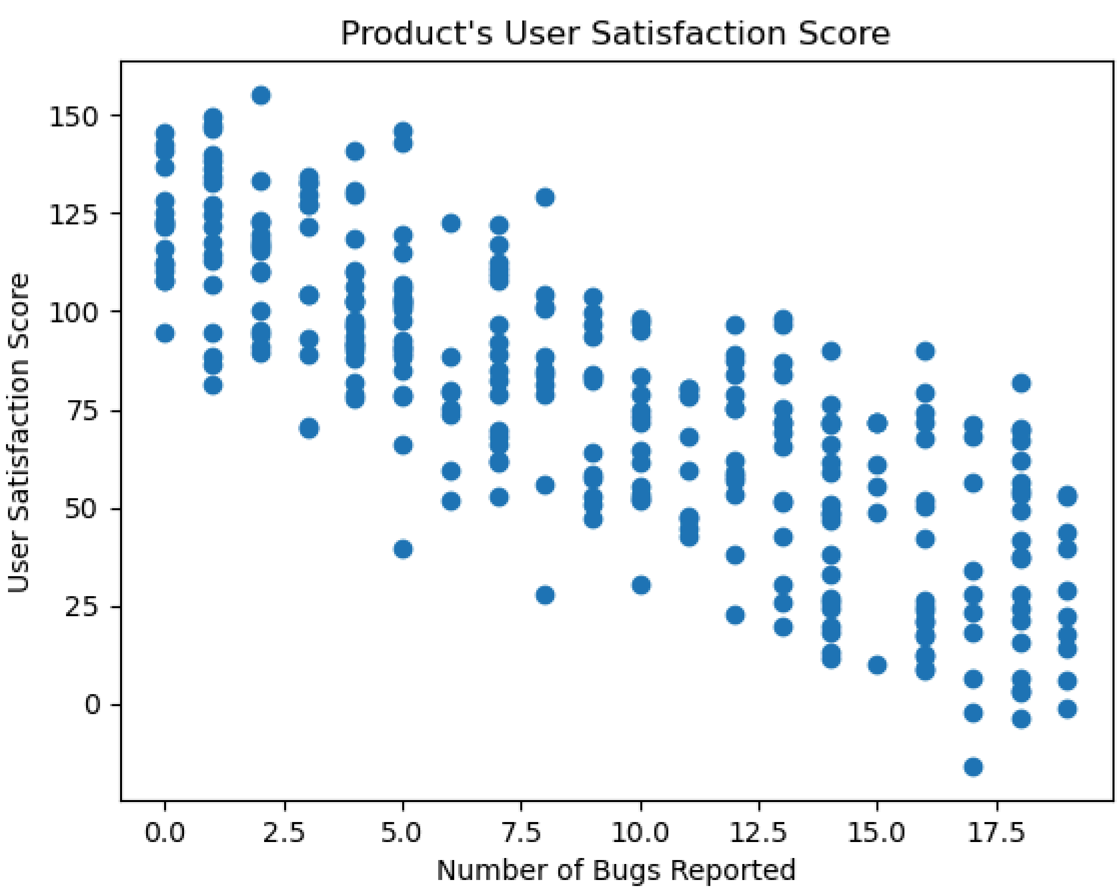
```
Intercept:  79.9852023102521
Coefficients: 

Number of Features: 1.4523797252294435
Number of Bugs Reported: -4.715339346133173
```

In [19]:
```
#7. Compute the necessary evaluation metrics, justify your choice, and analyze
#testing with new values

new_entry0=input("please enter the Number of Features :")
new_entry1=input("please enter the Number of Bugs Reported :")
new_entry=[[float(new_entry0),float(new_entry1)]]
pred_new=mlr_model.predict(new_entry)
print(pred_new)
```

```
please enter the Number of Features :6
please enter the Number of Bugs Reported :9
[46.26142655]

C:\Users\91637\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning:
X does not have valid feature names, but LinearRegression was fitted with fe
ature names
  warnings.warn(
```

In [20]:
```
#Finding Linear Model
indp_vars=Synthetic_app_data['Number of Bugs Reported'].values.reshape(-1,1)
dep_var=Synthetic_app_data['User Satisfaction Score'].values
plt.scatter(indp_vars,dep_var)
plt.xlabel("Number of Bugs Reported")
plt.ylabel("User Satisfaction Score")
plt.title("Product's User Satisfaction Score")
```

Out[20]:
```
Text(0.5, 1.0, "Product's User Satisfaction Score")
```



In [32]:
```
train_x1,test_x1,train_y1,test_y1=train_test_split(indp_vars,dep_var,test_size
```

In [33]:
```
lr_model=LinearRegression()
```

In [34]:
```
lr_model.fit(train_x1,train_y1)
```

Out[34]:
```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [35]:
```
pred=lr_model.predict(test_x1)
pred
```

Out[35]:
```
array([110.37680676, 110.37680676,  95.61858642,  85.77977287,
        66.10214575,  66.10214575,  95.61858642,  85.77977287,
        66.10214575,  26.74689152,  51.34392541, 100.5379932 ,
        95.61858642, 115.29621354,  85.77977287,  75.94095931,
        85.77977287,  31.6662983 ,  61.18273897, 120.21562032,
        61.18273897,  61.18273897, 105.45739998,  66.10214575,
        95.61858642,  80.86036609, 105.45739998,  41.50511186,
        71.02155253,  41.50511186, 110.37680676,  66.10214575,
        26.74689152,  41.50511186,  26.74689152,  36.58570508,
        51.34392541, 100.5379932 ,  75.94095931,  56.26333219,
       110.37680676,  95.61858642,  46.42451864,  85.77977287,
        61.18273897,  85.77977287,  61.18273897,  41.50511186,
        71.02155253,  95.61858642, 110.37680676,  71.02155253,
        26.74689152,  66.10214575, 100.5379932 , 110.37680676,
        51.34392541,  95.61858642,  61.18273897,  31.6662983 ,
       105.45739998,  26.74689152, 115.29621354,  71.02155253,
       105.45739998,  95.61858642,  31.6662983 , 100.5379932 ,
       100.5379932 , 110.37680676, 105.45739998, 115.29621354,
        51.34392541,  41.50511186, 100.5379932 ,  95.61858642,
        85.77977287, 100.5379932 , 100.5379932 ,  41.50511186,
       120.21562032, 100.5379932 ,  31.6662983 ,  80.86036609,
        61.18273897, 120.21562032, 115.29621354,  51.34392541,
        26.74689152,  46.42451864])
```

In [36]:
```
mse1=mean_squared_error(pred,test_y1)
mse1
```

Out[36]:
```
445.57838814765375
```

In [40]:
```
print("The mean squared error for MLR is: ", mse_mlr)
print("The mean squared error for SLR is: ", mse1)
print("So when we compared the errors we could see that the Multiple regressic
```

```
The mean squared error for MLR is:  154.311721846551
The mean squared error for SLR is:  445.57838814765375
So when we compared the errors we could see that the Multiple regression mod
el has less error and help us to do the good prediction
```

In [ ]: