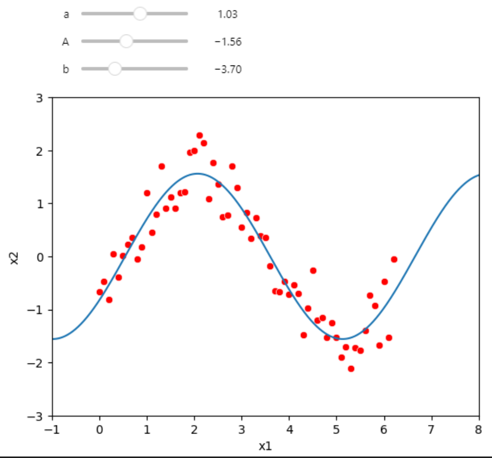
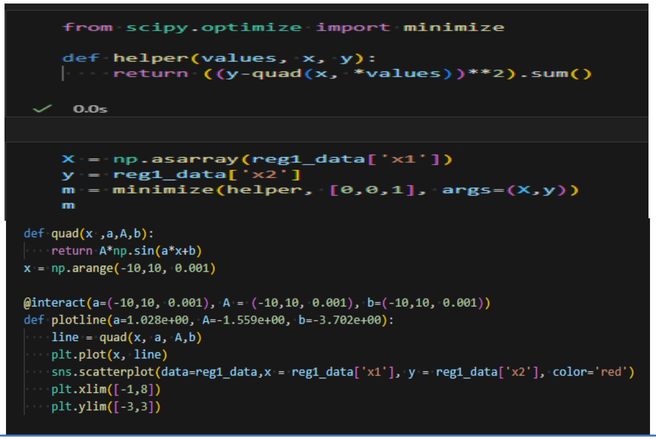
**A.** The data seems to be of some sort of wave form, maybe at different time intervals. It largely resembles a sine wave or an alternating behaviour.





```
from scipy.optimize import minimize

def helper(values, x, y):
    return ((y-quad(x, *values))**2).sum()
```

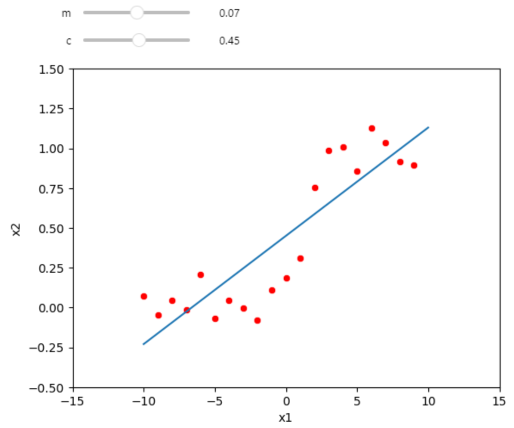
0.0s

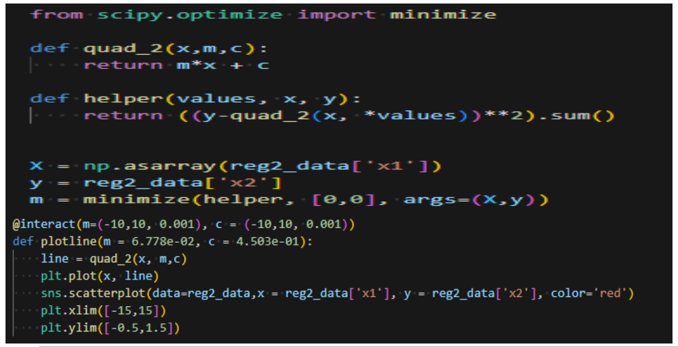
```
X = np.asarray(reg1_data['x1'])
y = reg1_data['x2']
m = minimize(helper, [0,0,1], args=(X,y))
m
```

```
def quad(x ,a,A,b):
    return A*np.sin(a*x+b)
x = np.arange(-10,10, 0.001)

@interact(a=(-10,10, 0.001), A = (-10,10, 0.001), b=(-10,10, 0.001))
def plotline(a=1.028e+00, A=-1.559e+00, b=-3.702e+00):
    line = quad(x, a, A,b)
    plt.plot(x, line)
    sns.scatterplot(data=reg1_data,x = reg1_data['x1'], y = reg1_data['x2'], color='red')
    plt.xlim([-1,8])
    plt.ylim([-3,3])
```

In regression 1, as stated above there seems to be some form of oscillatory behaviour. Thus, a sine function was chosen and was used to fit the dataset. I used the minimization of mean squared error to obtain the optimal values for amplitude, frequency and phase shift of the sine curve. This shows an acceptable behaviour.

**B.** The data is highly noisy and very few samples are provided to make any proper sense of the same. There seems to be an increasing trend, though this cannot be stated conclusively.

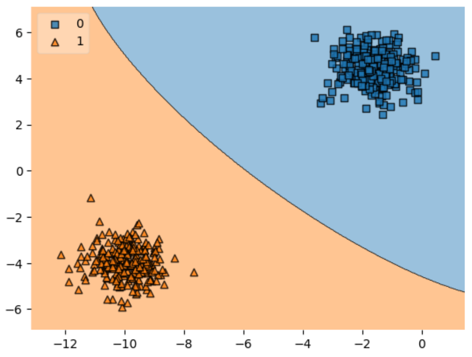


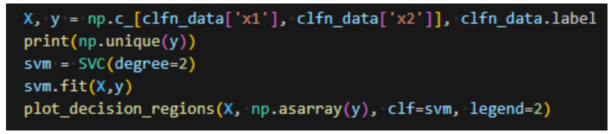


```
from scipy.optimize import minimize

def quad_2(x,m,c):
    return m*x + c

def helper(values, x, y):
    return ((y-quad_2(x, *values))**2).sum()

X = np.asarray(reg2_data['x1'])
y = reg2_data['x2']
m = minimize(helper, [0,0], args=(X,y))
@interact(m=(-10,10, 0.001), c = (-10,10, 0.001))
def plotline(m = 6.778e-02, c = 4.503e-01):
    line = quad_2(x, m,c)
    plt.plot(x, line)
    sns.scatterplot(data=reg2_data,x = reg2_data['x1'], y = reg2_data['x2'], color='red')
    plt.xlim([-15,15])
    plt.ylim([-0.5,1.5])
```

In regression 2, there is hardly any data samples to work with and thus a line was chosen to be perhaps the best way to fit a curve. We followed a similar minimization approach as that of 1, although it was difficult to gain any particular insight into the results. A ramp or Heaviside function was also considered but then disregarded as this would result in overfitting.

**C.** The classification dataset can be seen clustered into 2 major groups. It could represent 2 major categories in the dataset and there don't seem to be any particular outliers of great interest.





```
X, y = np.c_[clfn_data['x1'], clfn_data['x2']], clfn_data.label
print(np.unique(y))
svm = SVC(degree=2)
svm.fit(X,y)
plot_decision_regions(X, np.asarray(y), clf=svm, legend=2)
```

In the classification dataset, a support vector machine was used to show the distinction between he two clusters. SVMs work by finding the hyperplane that best separates the two classes of data. This method was chosen as it is proven to show a high degree of accuracy and a clean visualization for the user.

Alap Mundayoor, M.N. 23351429, IdM: ow07ysym