a. Anxiety Disorder

```
# creating a regression object

regression_anxiety = LinearRegression()
```

```
# fitting the regression object onto the training set

regression_anxiety.fit(X_train, y_anxiety_train)
```

```
LinearRegression()
```

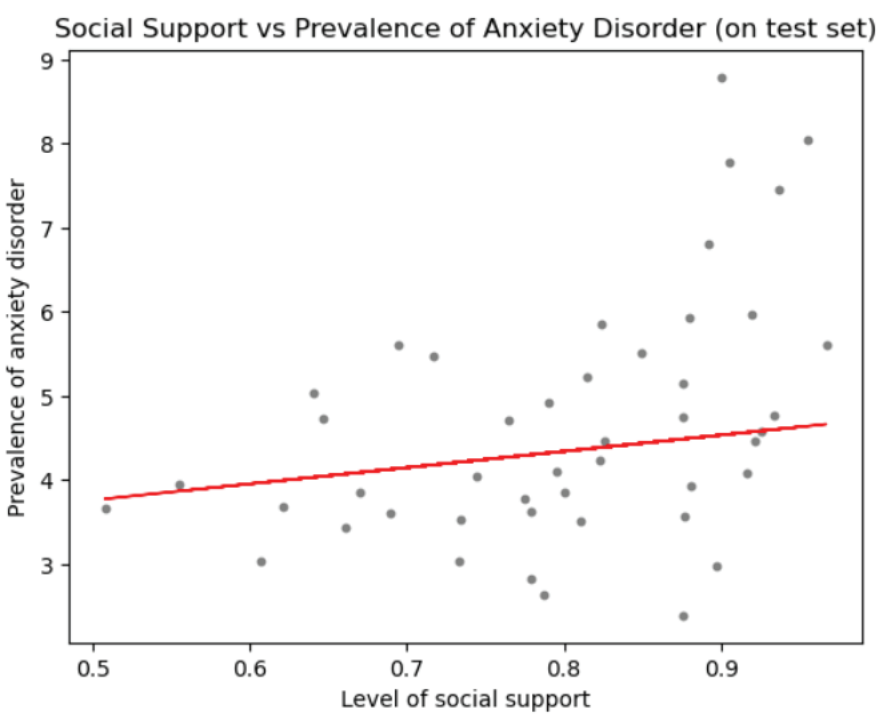
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# predicting the values of y using X

y_anxiety_predicted = regression_anxiety.predict(X_test)
```

```
# creating a scatterplot that shows the regression line on the test set

plot_test = plt
plot_test.scatter(X_test, y_anxiety_test, color='gray', s = 10)
plot_test.plot(X_test, y_anxiety_predicted, color='red', linewidth =1)
plot_test.title('Social Support vs Prevalence of Anxiety Disorder (on test set)')
plot_test.xlabel('Level of social support')
plot_test.ylabel('Prevalence of anxiety disorder')
plot_test.show()
```




```
# creating objects that contain the model summary statistics

rmse_anxiety = mean_squared_error(y_test, y_anxiety_predicted)
r2_anxiety = r2_score(y_test, y_anxiety_predicted)
```

```
# printing the model summary statistics to evaluate performance of model

print('Slope:' ,regression_anxiety.coef_)
print('Mean squared error: ', rmse_anxiety)
print('R2 score: ', r2_anxiety)
```

```
Slope: [[1.93456081]]
Mean squared error:  1.89766449905987
R2 score:  0.06902149280153613
```

```
# creating a dataframe comparing actual and predicted values of y

data = pd.DataFrame({'Actual': y_anxiety_test.flatten(), 'Predicted': y_anxiety_predicted.flatten()})
data.head(30)
```