M2-L2 Problem 2 (5 points)

In this problem we will perform least-squares regression using sklearn's built-in tools.

First, you will generate a standard linear least squares regression model with `LinearRegression`.

Next, you will use stochastic gradient descent to train another model with `SGDRegressor`.

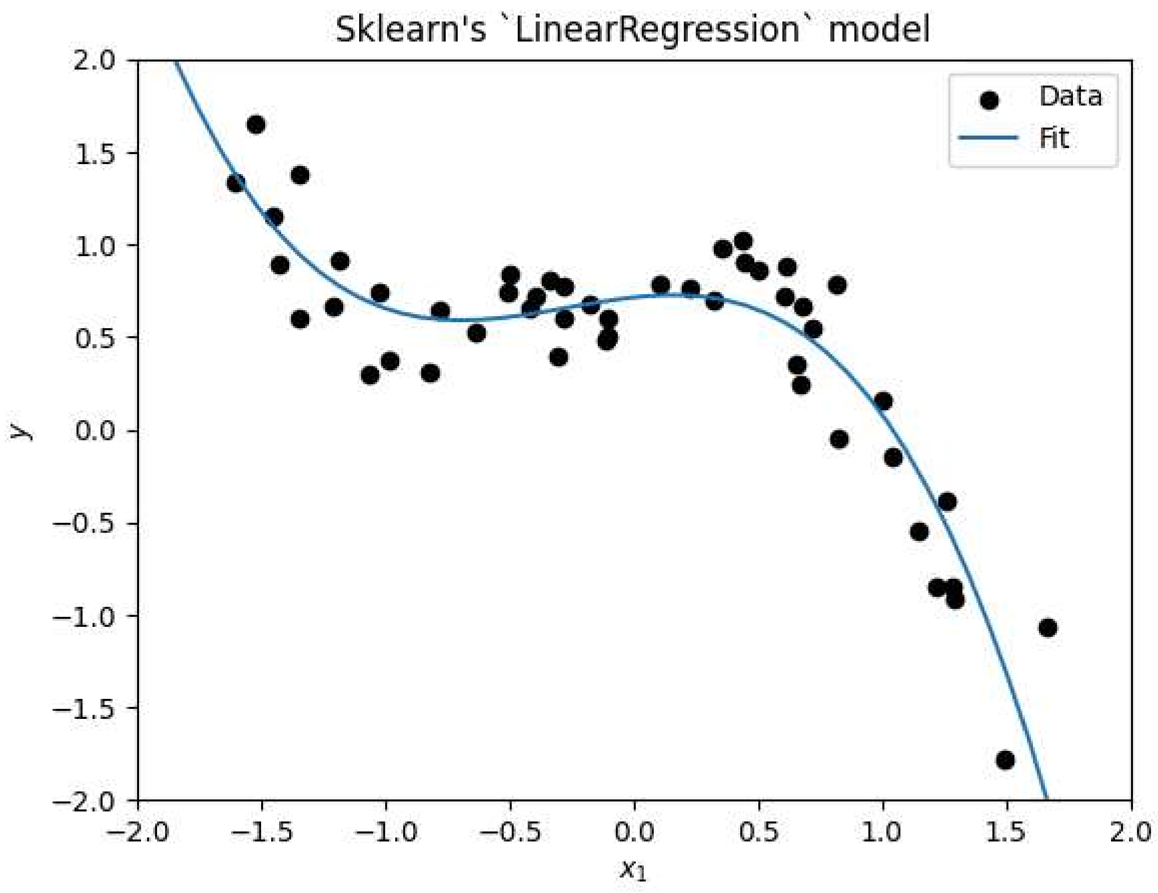
Run this cell to perform the required imports and load the data:

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import SGDRegressor

def plot_data_with_regression(x_data, y_data, x_reg, y_reg, title=""):
    plt.figure()

    plt.scatter(x_data.flatten(), y_data.flatten(), label="Data", c="black")
    plt.plot(x_reg.flatten(), y_reg.flatten(), label="Fit")

    plt.legend()
    plt.xlabel(r"$x_1$")
    plt.ylabel(r"$y$")
    plt.xlim(-2,2)
    plt.ylim(-2,2)
    plt.title(title)
    plt.show()

x = np.array([-1.52362349, -1.60576489, -1.34827768, -1.45340266, -1.42652973, -1.2
y = np.array([ 1.65517515,  1.33249684,  1.38328432,  1.1531808 ,  0.89478436,0.667
X = np.vstack([x*x*x, x*x, x, np.ones_like(x)]).T

xreg = np.linspace(-2,2)
Xreg = np.vstack([xreg*xreg*xreg, xreg*xreg, xreg, np.ones_like(xreg)]).T
```

Least Squares Regression

We have provided a demonstration of least squares regression using sklearn:

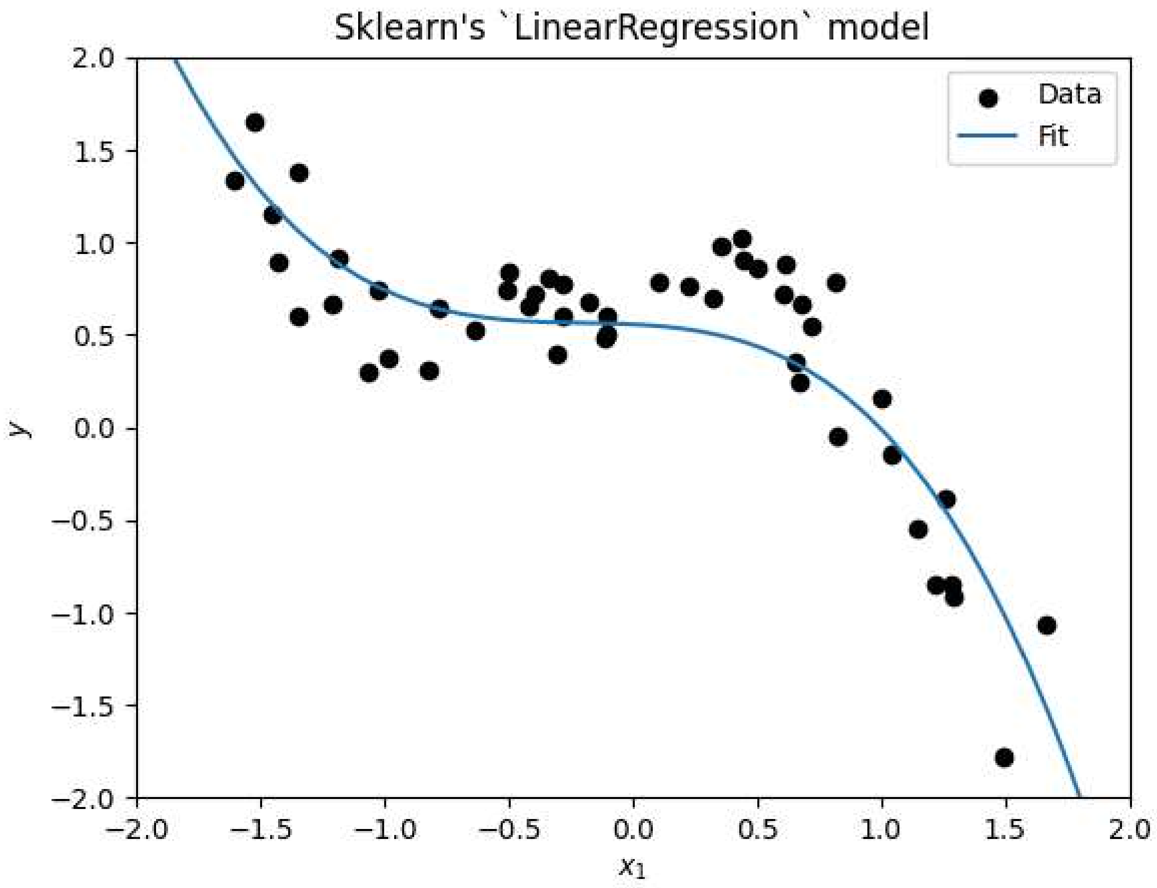
In [3]:
```
model = LinearRegression()
model.fit(X,y)

yreg = model.predict(Xreg)
plot_data_with_regression(x, y, xreg, yreg, "Sklearn's `LinearRegression` model")
```



Using SGD

Now use stochastic gradient descent to solve the same problem and make a similar plot, but for a `SGDRegressor` model instead of `LinearRegression`:

In [4]:
```
# YOUR CODE GOES HERE

model = SGDRegressor()
model.fit(X,y)

yreg = model.predict(Xreg)
plot_data_with_regression(x, y, xreg, yreg, "Sklearn's `LinearRegression` model")
```


Sklearn's `LinearRegression` model
Data
Fit
y
x_1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

```
In [ ]:
```