Date.....

Writing my own class...

```
class My_BatchGradientDescent:

    def __init__(self, learning_rate, epochs):      # Constructor
        self.intercept_ = None
        self.coef_ = None
        self.learning_rate = learning_rate
        self.epochs = epochs

    def fit(self, X_train, Y_train):
        # Initializing to coef and intercept.
        self.intercept_ = 0
        self.coef = np.ones(X_train[1])
        for a in range(self.epochs):
            # Updating the values of coef and intercept.--> self.intercept_
            Y_Predicted = np.dot(X_train, self.coef_)# Vectorization
            intercept_der = -2 * np.mean(Y_train - Y_Predicted)
            self.intercept_ = (self.intercept_ - (self.learning_rate * intercept_der))

            coef_der = -2 * np.dot((Y_train, Y_Predicted), X_train)/X_train.
                                                                shape[0]
            self.coef_ = (self.coef_ - (self.learning_rate * coef_der))

def predict(X_test, self):

    return np.dot(X_train, self.coef_) + self.intercept_
    # As per the Formula [mx+b]
```