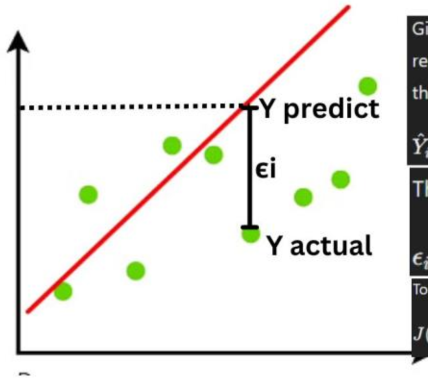
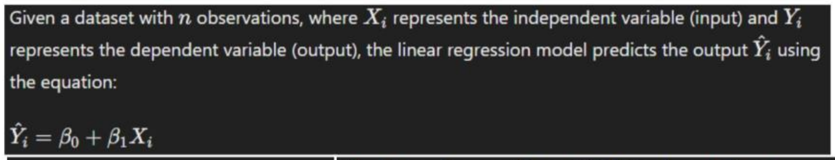
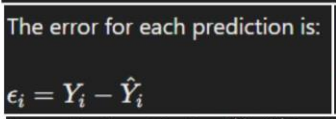
Linear Regression

- For Continuous Variables(features)
- **Simple Linear Regression** - single independent variable. The relationship between X and Y
- **Multiple Linear Regression** - involves two or more independent variables.
- **best fit line** - Finding the best fit line using the Mean Squared Error (MSE) involves identifying the line that minimizes the average of the squared differences between the observed values and the values predicted by the line.

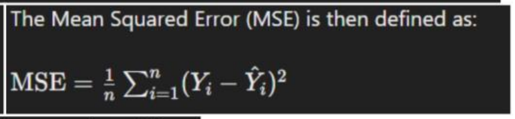



Given a dataset with n observations, where X_i represents the independent variable (input) and Y_i represents the dependent variable (output), the linear regression model predicts the output $\hat{Y}_i$ using the equation:

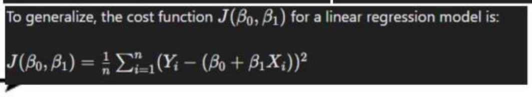
$$\hat{Y}_i = \beta_0 + \beta_1 X_i$$


The error for each prediction is:

$$\epsilon_i = Y_i - \hat{Y}_i$$

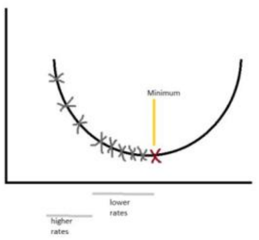

The Mean Squared Error (MSE) is then defined as:

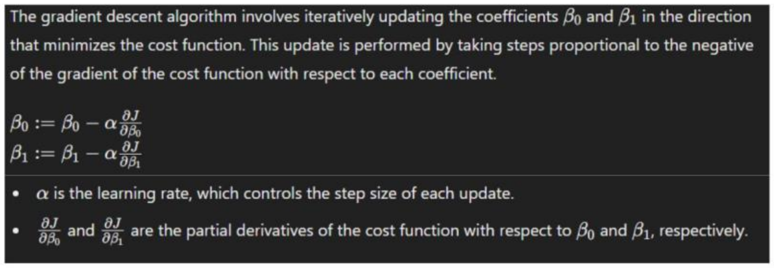
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$


To generalize, the cost function $J(\beta_0, \beta_1)$ for a linear regression model is:

$$J(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - (\beta_0 + \beta_1 X_i))^2$$

Update the Coefficients




The gradient descent algorithm involves iteratively updating the coefficients β_0 and β_1 in the direction that minimizes the cost function. This update is performed by taking steps proportional to the negative of the gradient of the cost function with respect to each coefficient.

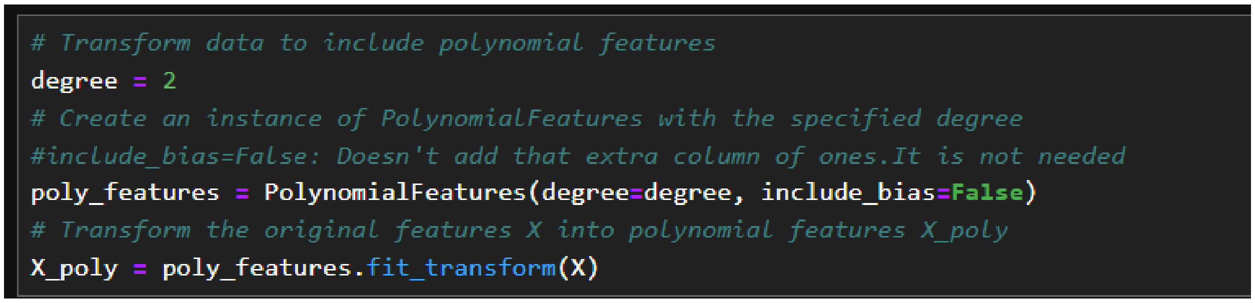
$$\beta_0 := \beta_0 - \alpha\frac{\partial J}{\partial \beta_0}$$
$$\beta_1 := \beta_1 - \alpha\frac{\partial J}{\partial \beta_1}$$

- α is the learning rate, which controls the step size of each update.
- $\frac{\partial J}{\partial \beta_0}$ and $\frac{\partial J}{\partial \beta_1}$ are the partial derivatives of the cost function with respect to β_0 and β_1, respectively.

- **Cost Function**
  - linear regression, the most commonly used cost function is the Mean Squared Error (MSE).
  - MSE measures the average of the squares of the errors.
  - error is the difference between the observed value and the predicted value.
- **Gradient descent** is a powerful optimization algorithm used to minimize the cost function in linear regression. By iteratively updating the coefficients in the direction that reduces the error, gradient descent helps find the best-fitting line that represents the relationship between the independent and dependent variables in the data.

- **Polynomial Regression**
  This approach can capture the non-linear relationships between variables, which linear regression cannot. (like curve relationships)

  We use PolynomialFeatures from scikit-learn to transform the feature matrix X to include polynomial terms up to the specified degree (2, 3, 4), (here 2)



```python
# Transform data to include polynomial features
degree = 2
# Create an instance of PolynomialFeatures with the specified degree
#include_bias=False: Doesn't add that extra column of ones.It is not needed
poly_features = PolynomialFeatures(degree=degree, include_bias=False)
# Transform the original features X into polynomial features X_poly
X_poly = poly_features.fit_transform(X)
```

$Y = m_1X + m_2X + c$
polynomial regression to estimate the coefficients m_1, m_2, c that minimize the error between the observed and predicted prices.