Linear and Polynomial Regression

Simple Linear Model

When to Use: Use when you have one input variable (x) and want to fit a straight line to predict the output (y).

$$f_{w,b}(x) = wx + b$$

x : input, y : target, w : weight, b : bias.
Prediction Code:

```
def predict(x_input, w, b):
    return w*x_input + b
```

Cost Function (MSE)

Purpose: Measures how well your model's predictions match actual values, guiding adjustments to w and b.

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

MSE Code:

```
def compute_cost(x, y, w, b):
    m = len(x)
    cost = 0.0
    for i in range(m):
        f_wb = w*x[i] + b
        cost += (f_wb - y[i])**2
    return cost/(2*m)
```

Gradient & Gradient Descent

Purpose: Update the model parameters by moving them in the direction of decreasing cost, ensuring better predictions.

$$\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})x^{(i)}, \quad \frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})$$

Update step: $w \leftarrow w - \alpha\frac{\partial J}{\partial w}$, $b \leftarrow b - \alpha\frac{\partial J}{\partial b}$

Multiple Linear Regression

When to Use: Extend linear regression to multiple input features ($\mathbf{x}$) to model more complex relationships.

$$f_{\mathbf{w},b}(\mathbf{x}) = \sum_{j=1}^{n} w_j x_j + b$$

Prediction Code:

```
import numpy as np

def predict_multiple(x_input, w, b):
    return np.dot(x_input, w) + b
```

Polynomial Regression

When to Use: Model nonlinear relationships by introducing polynomial terms of your input variable x.

$$f_{w,b}(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$$

Regularization (Ridge & Lasso)

When to Use: Prevent overfitting by penalizing large weights, improving the model's ability to generalize.
Ridge (L2):

$$J_{\text{ridge}}(\mathbf{w}, b) = J(\mathbf{w}, b) + \lambda\sum_{j=1}^{n}(w_j)^2$$

Lasso (L1):

$$J_{\text{lasso}}(\mathbf{w}, b) = J(\mathbf{w}, b) + \lambda\sum_{j=1}^{n}|w_j|$$

Evaluation Metrics

Purpose: Quantify how well your model performs, guiding model selection and improvement.

- MAE: $\frac{1}{m}\sum|f_{w,b}(x) - y|$
- RMSE: $\sqrt{\frac{1}{m}\sum(f_{w,b}(x) - y)^2}$
- MAPE: $\frac{100\%}{m}\sum\frac{|f_{w,b}(x)-y|}{|y|}$
- R^2: $1 - \frac{\sum(y-f_{w,b}(x))^2}{\sum(y-\bar{y})^2}$

Practical Tips

Purpose: Improve training efficiency, model robustness, and ensure better generalization.

- Normalize/Standardize input features.
- Experiment with different learning rates (α).
- Use cross-validation for hyperparameter tuning and reliable performance estimation.
- Monitor the cost function $J(w,b)$ during training for convergence.
- Combine polynomial features with regularization for complex data without overfitting.