

Nama : Zahrani Cahya Priesa

NIM : 1103223074

Mata Kuliah : Machine Learning

### Analisis Cara Kerja Model Linier (*Chapter 4*)

#### I. Regresi Linier (*Linear Regression*)

**Cara Kerja Inti:** Regresi Linier bertujuan untuk menemukan **koefisien** () yang paling cocok untuk membentuk garis lurus yang meminimalkan *Root Mean Square Error* (RMSE) antara garis prediksi dan titik data aktual.

- **Formula Model:** Prediksi () dihitung sebagai:
- **Fungsi Biaya (*Cost Function*):** Kode Anda menggunakan MSE (Mean Squared Error). Tujuannya adalah meminimalkan fungsi biaya ini.

#### Metode Pelatihan:

##### 1. Metode *Normal Equation*:

- **Cara Kerja Coding:** Kode menghitung . Ini adalah solusi analitis (solusi langsung) yang memberikan optimal secara eksak.
- **Keuntungan:** Cepat untuk dataset kecil, tidak memerlukan penskalaan fitur.
- **Kekurangan:** Menjadi lambat dan membutuhkan banyak memori jika jumlah fitur () sangat besar (kompleksitas hingga ).

##### 2. Metode *Gradient Descent* (GD):

- **Cara Kerja Code:** Ini adalah proses iteratif. Model dimulai dengan acak, lalu secara bertahap menyesuaikan dalam arah yang mengurangi fungsi biaya.
- **Rumus Iterasi Kunci:** , di mana adalah *learning rate* dan adalah gradien fungsi biaya.
- **Jenis-jenis GD yang Ada di Notebook:**
  - **Batch Gradient Descent (BGD):** Menggunakan **seluruh** set pelatihan untuk menghitung gradien pada setiap langkah. Ini lambat, tetapi menjamin menemukan minimum global (jika fungsi biayanya cembung).
  - **Stochastic Gradient Descent (SGD):** Mengambil hanya **satu instans** pelatihan secara acak pada setiap langkah. Jauh lebih cepat dan dapat

mengatasi set data yang sangat besar. Namun, karena keacakan, fungsi biaya tidak akan pernah benar-benar mencapai minimum, tetapi akan berosilasi di sekitarnya.

- **Mini-batch Gradient Descent:** Kompromi, menggunakan sekumpulan kecil (*mini-batch*) instans. Lebih stabil daripada SGD.

## II. Regresi Polinomial (Polynomial Regression)

**Cara Kerja Inti:** Ini bukan model non-linier yang sebenarnya, tetapi merupakan **Regresi Linier** yang diterapkan pada fitur-fitur data yang dipangkatkan. Ini memungkinkan model untuk menemukan pola non-linier.

- **Cara Kerja Coding:**

1. Menggunakan `PolynomialFeatures` untuk mentransformasi data input (misalnya, menjadi  $x^2$ ).
2. Model `LinearRegression` kemudian dilatih pada data yang sudah ditransformasi tersebut.

- **Implikasi:** Derajat (*degree*) polinomial yang tinggi (misalnya,  $x^5$ ) dapat menyebabkan **Overfitting** yang parah, di mana model menyesuaikan terlalu dekat dengan *noise* dalam data latih.

## III. Kurva Pembelajaran (*Learning Curves*)

**Tujuan:** Menganalisis bagaimana kinerja model berubah seiring dengan bertambahnya jumlah instans pelatihan.

- **Cara Kerja Coding:** Plot menunjukkan RMSE (atau metrik lainnya) terhadap ukuran set pelatihan.
- **Interpretasi Kunci:**
  - **Underfitting (Kurva Jauh):** Kedua kurva (pelatihan dan validasi) tinggi dan berdekatan. Menandakan model terlalu sederhana.
  - **Overfitting (Kesenjangan Besar):** Kurva pelatihan sangat rendah, tetapi kurva validasi tinggi, dan ada kesenjangan besar di antara keduanya. Menandakan model terlalu kompleks.

## IV. Regularisasi (Regularization)

**Tujuan:** Mencegah *Overfitting* dengan menambahkan **penalti** ke fungsi biaya, yang memaksa model untuk menjaga agar bobot fitur ( $w$ ) tetap kecil dan sederhana.

- **Cara Kerja Coding:**

1. **Ridge Regression (L2 Regularization):** Menambahkan ke fungsi biaya. Ini menekan bobot fitur mendekati nol, tetapi **tidak sepenuhnya nol**.
2. **Lasso Regression (L1 Regularization):** Menambahkan ke fungsi biaya. Ini memiliki efek samping yang berguna: ia cenderung membuat bobot fitur yang tidak penting **menjadi nol**, secara efektif melakukan pemilihan fitur otomatis.
3. **Elastic Net:** Gabungan dari Ridge dan Lasso. Umumnya lebih disukai daripada Lasso murni.
  - **Hiperparameter Penting:** Parameter (*regularization strength*) harus disetel. Nilai yang lebih tinggi berarti regularisasi yang lebih kuat.

## V. Regresi Logistik dan Softmax

**Tujuan:** Model ini digunakan untuk tugas **Klasifikasi**, bukan Regresi.

1. **Regresi Logistik (Binary Classification):**

- **Cara Kerja Inti:** Model ini menghitung probabilitas bahwa instans termasuk dalam kelas positif (1) dengan melewati prediksi linier () melalui **fungsi Sigmoid** ().
- **Keluaran:** Hasilnya adalah probabilitas antara 0 dan 1.
- **Keputusan:** Jika probabilitas  $> 0.5$ , diklasifikasikan sebagai positif.

2. **Regresi Softmax (Multiclass Classification):**

- **Cara Kerja Inti:** Regresi Logistik yang digeneralisasi. Untuk setiap instans, model menghitung skor untuk setiap kelas , lalu menggunakan **fungsi Softmax** untuk mengubah skor ini menjadi probabilitas kelas .
- **Fungsi Softmax:**
- **Keputusan:** Kelas dengan probabilitas tertinggi dipilih sebagai prediksi.
- **Contoh di Notebook:** Model dilatih pada dataset Iris untuk memprediksi spesies bunga.

## Kesimpulan

*Kode* dalam `Chapter_4_training_linear_models.ipynb` secara efektif mendemonstrasikan **landasan *Machine Learning* Linier**, bergerak dari solusi analitis (Normal Equation) ke solusi iteratif (Gradient Descent), dan kemudian memperkenalkan teknik untuk mengatasi masalah *Underfitting* dan *Overfitting* (PolynomialFeatures dan Regularization). Semua konsep ini penting sebagai blok bangunan dasar sebelum pindah ke model yang lebih kompleks.