**Mata Kuliah Coding & Machine Learning**

**Laporan Tugas Modul Hal. 26**

**Dosen Pengampu: Sri Wulandari, S.Kom., M.Cs.**



**Disusun oleh:**

**Lathif Ramadhan (5231811022)**

**PROGRAM STUDI SAINS DATA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**YOGYAKARTA**

**2024**

# Daftar Isi

[Daftar Isi 2](#_Toc192114761)

[1. Berikan penjelasan perbedaan dan contoh implementasi dari regresi linear, logistic regression dan GLM! 5](#_Toc192114762)

[A. Regresi Linear 5](#_Toc192114763)

[Rumus Regresi Linear 5](#_Toc192114764)

[Contoh Implementasi 5](#_Toc192114765)

[Kapan Menggunakan Regresi Linear? 7](#_Toc192114766)

[B. Logistic Regression 8](#_Toc192114767)

[Rumus Logistic Regression 8](#_Toc192114768)

[Contoh Implementasi 8](#_Toc192114769)

[Kapan Menggunakan Logistic Regression? 10](#_Toc192114770)

[C. Generalized Linear Model (GLM) 11](#_Toc192114771)

[Contoh Implementasi 11](#_Toc192114772)

[Kapan Menggunakan GLM? 12](#_Toc192114773)

[Tabel Perbedaan 13](#_Toc192114774)

[2. Perhatikan parameter-parameter yang dimiliki oleh algoritma-algoritma pembelajaran terbimbing yang telah dijelaskan pada bab 2 ini, berikan penjelasan makna dari parameter-parameter tersebut! 14](#_Toc192114775)

[A. Regresi Linear di RapidMiner 14](#_Toc192114776)

[1. Feature Selection 14](#_Toc192114777)

[2. Alpha 14](#_Toc192114778)

[3. Max Iterations 14](#_Toc192114779)

[4. Forward Alpha 14](#_Toc192114780)

[5. Backward Alpha 15](#_Toc192114781)

[6. Eliminate Colinear Features 15](#_Toc192114782)

[7. Min Tolerance 15](#_Toc192114783)

[8. Use Bias 15](#_Toc192114784)

[9. Ridge 15](#_Toc192114785)

[B. Logistic Regression di RapidMiner 15](#_Toc192114786)

[1. Kernel Type 15](#_Toc192114787)

[2. Kernel Gamma 16](#_Toc192114788)

[3. Kernel Sigma1, Sigma2, Sigma3 16](#_Toc192114789)

[4. Kernel Shift 16](#_Toc192114790)

[5. Kernel Degree 16](#_Toc192114791)

[6. Kernel A dan B 17](#_Toc192114792)

[7. C (Complexity Constant) 17](#_Toc192114793)

[8. Start Population Type 17](#_Toc192114794)

[9. Max Generations 17](#_Toc192114795)

[10. Generations Without Improval 17](#_Toc192114796)

[11. Population Size 17](#_Toc192114797)

[12. Tournament Fraction 17](#_Toc192114798)

[13. Keep Best 17](#_Toc192114799)

[14. Mutation Type 18](#_Toc192114800)

[15. Selection Type 18](#_Toc192114801)

[16. Crossover Prob 18](#_Toc192114802)

[17. Use Local Random Seed 18](#_Toc192114803)

[18. Local Random Seed 18](#_Toc192114804)

[19. Show Convergence Plot 18](#_Toc192114805)

[C. Generalized Linear Models (GLM) di RapidMiner 18](#_Toc192114806)

[1. Family 18](#_Toc192114807)

[2. Solver 19](#_Toc192114808)

[3. Link 19](#_Toc192114809)

[4. Reproducible 19](#_Toc192114810)

[5. Maximum Number of Threads 19](#_Toc192114811)

[6. Specify Beta Constraints 20](#_Toc192114812)

[7. Use Regularization 20](#_Toc192114813)

[8. Lambda 20](#_Toc192114814)

[9. Lambda Search 20](#_Toc192114815)

[10. Number of Lambdas 20](#_Toc192114816)

[11. Lambda Min Ratio 20](#_Toc192114817)

[12. Early Stopping 20](#_Toc192114818)

[13. Stopping Rounds 20](#_Toc192114819)

[14. Stopping Tolerance 21](#_Toc192114820)

[15. Alpha 21](#_Toc192114821)

[16. Standardize 21](#_Toc192114822)

[17. Non-Negative Coefficients 21](#_Toc192114823)

[18. Compute P-Values 21](#_Toc192114824)

[19. Remove Collinear Columns 21](#_Toc192114825)

[20. Add Intercept 21](#_Toc192114826)

[21. Missing Values Handling 21](#_Toc192114827)

[22. Max Iterations 21](#_Toc192114828)

[23. Beta Constraints 22](#_Toc192114829)

[24. Max Runtime Seconds 22](#_Toc192114830)

[25. Expert Parameters 22](#_Toc192114831)

[Kesimpulan 22](#_Toc192114832)

[3. Dengan menggunakan parameter-parameter yang telah saudara berikan penjelasan, ubahlah semua model yang ada di bab 2 ini dengan parameter tersebut, carilah parameter yang menghasilkan akurasi terbaik! 23](#_Toc192114833)

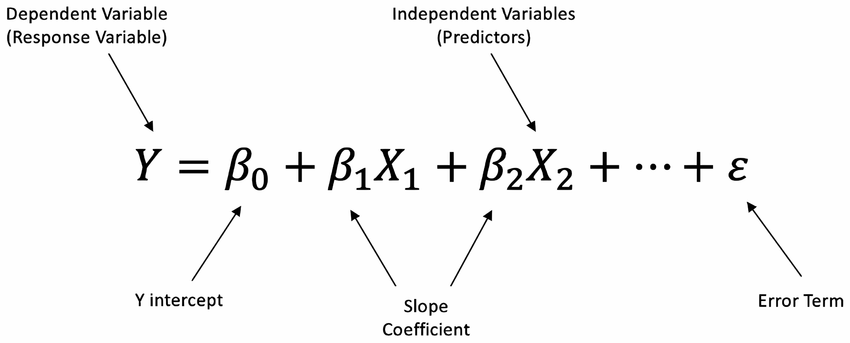
[Referensi 24](#_Toc192114834)

# Berikan penjelasan perbedaan dan contoh implementasi dari regresi linear, logistic regression dan GLM!

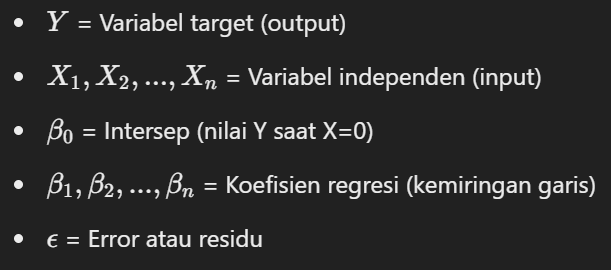
## A. Regresi Linear

**Regresi linear** adalah metode yang digunakan untuk memprediksi nilai **kontinu** berdasarkan hubungan linear antara variabel independen dan variabel dependen. Model ini bekerja dengan mencari garis lurus terbaik yang meminimalkan selisih antara nilai prediksi dan nilai aktual.

### Rumus Regresi Linear



Dimana:



### Contoh Implementasi

Misalkan kita ingin memprediksi **harga rumah** berdasarkan luas tanah dan jumlah kamar. Kita bisa menggunakan regresi linear untuk menemukan hubungan antara luas rumah dan harganya.

**Contoh kode Python:**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

# Data (Luas rumah dalam meter persegi dan harga dalam juta rupiah)

X = np.array([30, 50, 70, 90, 110]).reshape(-1, 1)

y = np.array([300, 500, 700, 900, 1100])

# Membuat model regresi linear

model = LinearRegression()

model.fit(X, y)

# Prediksi harga rumah

X\_test = np.array([60]).reshape(-1, 1)

y\_pred = model.predict(X\_test)

print(f"Prediksi harga rumah untuk luas 60m²: {y\_pred[0]:.2f} juta")

# Visualisasi

plt.scatter(X, y, color='blue', label='Data aktual')

plt.plot(X, model.predict(X), color='red', label='Garis regresi')

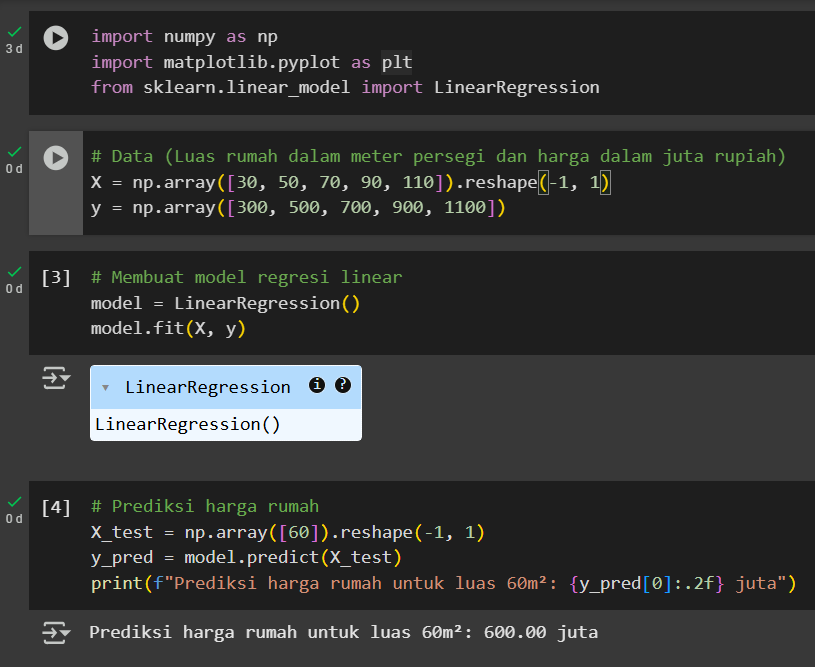
plt.xlabel('Luas Rumah (m²)')

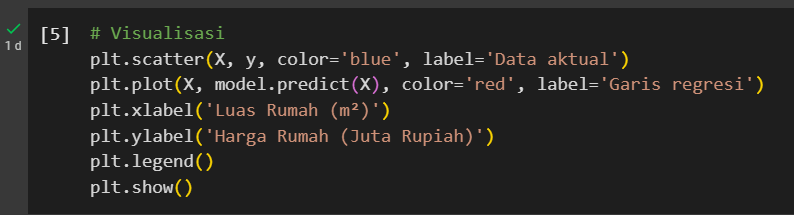
plt.ylabel('Harga Rumah (Juta Rupiah)')

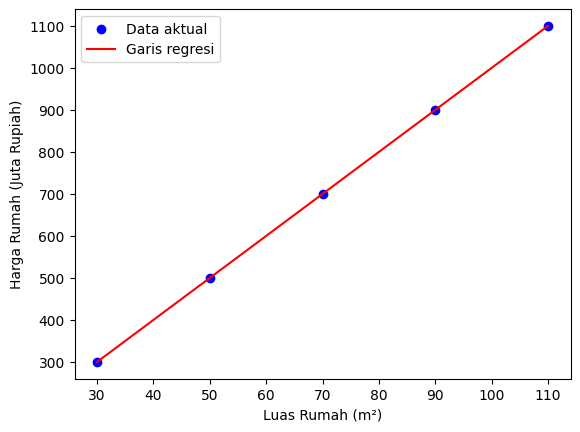
plt.legend()

plt.show()

**Berikut implementasinya di Google Colab:**







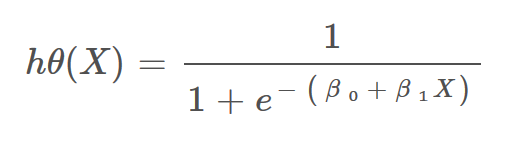
### Kapan Menggunakan Regresi Linear?

* Jika variabel target berupa **angka kontinu** (contoh: harga, gaji, suhu).
* Jika hubungan antara variabel independen dan target **bersifat linear**.

## B. Logistic Regression

**Logistic Regression** bukan digunakan untuk prediksi angka kontinu, melainkan untuk **klasifikasi biner** (ya/tidak, positif/negatif, sukses/gagal). Metode ini digunakan untuk menghitung probabilitas suatu kejadian terjadi atau tidak terjadi.

### Rumus Logistic Regression



Fungsi yang digunakan dalam logistic regression adalah **sigmoid function**, yang mengubah nilai regresi linear menjadi nilai probabilitas antara 0 dan 1.

### Contoh Implementasi

Misalnya, kita ingin memprediksi apakah seseorang akan diterima di universitas berdasarkan nilai ujian mereka.

**Contoh kode Python:**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression

# Data: Nilai ujian dan status diterima (1) atau tidak diterima (0)

X = np.array([50, 55, 60, 65, 70, 75, 80, 85, 90, 95]).reshape(-1, 1)

y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])

# Membuat model logistic regression

model = LogisticRegression()

model.fit(X, y)

# Prediksi peluang diterima untuk nilai 72

X\_test = np.array([72]).reshape(-1, 1)

y\_pred\_prob = model.predict\_proba(X\_test)

print(f"Peluang diterima dengan nilai 72: {y\_pred\_prob[0][1]:.2f}")

# Visualisasi

plt.scatter(X, y, color='blue', label='Data aktual')

plt.plot(X, model.predict\_proba(X)[:, 1], color='red', label='Kurva Sigmoid')

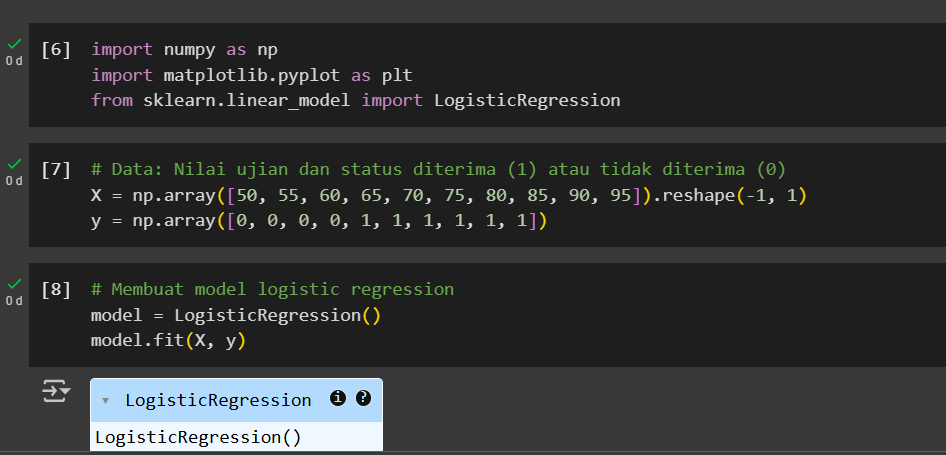
plt.xlabel('Nilai Ujian')

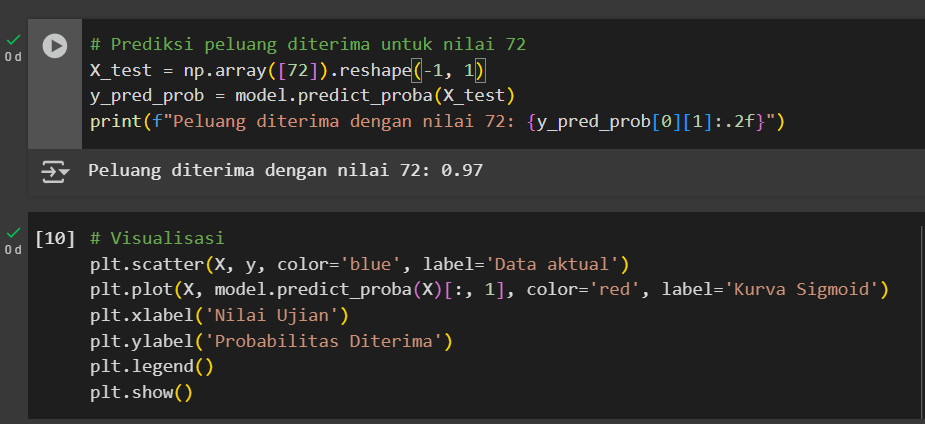
plt.ylabel('Probabilitas Diterima')

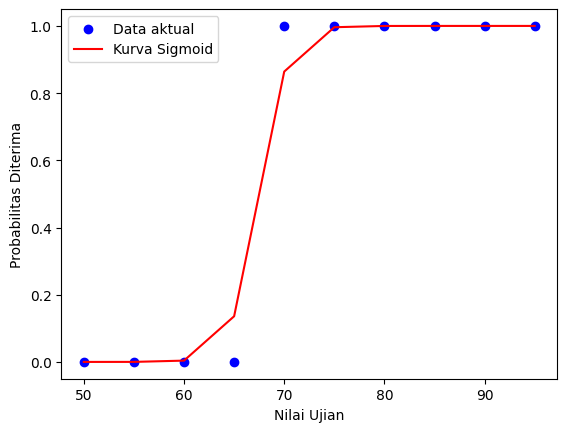
plt.legend()

plt.show()

**Berikut implementasinya di Google Colab:**







### Kapan Menggunakan Logistic Regression?

* Jika variabel target berupa **kategori (0 atau 1, spam atau bukan spam, lulus atau tidak lulus)**.
* Jika ingin menghitung **probabilitas** suatu kejadian.

## C. Generalized Linear Model (GLM)

**GLM (Generalized Linear Model)** adalah model statistik yang lebih fleksibel dibandingkan regresi linear dan logistic regression. GLM dapat menangani berbagai jenis distribusi data, seperti **normal, binomial, poisson, gamma**, dan lainnya.

GLM memiliki tiga komponen utama:

1. **Distribusi eksponensial** (Normal, Poisson, Binomial, dll).
2. **Fungsi link** (Linear, Logit, Log, dll).
3. **Model linear umum** yang menghubungkan variabel independen dengan variabel dependen.

### Contoh Implementasi

Jika kita ingin memprediksi jumlah pelanggan yang datang ke toko dalam sehari (data **count**, cocok untuk distribusi Poisson).

**Contoh kode Python (GLM dengan Poisson Regression):**

import statsmodels.api as sm

import numpy as np

# Data jumlah iklan dan pelanggan datang

X = np.array([1, 2, 3, 4, 5]) # Jumlah iklan

y = np.array([3, 7, 9, 14, 18]) # Jumlah pelanggan

# Membuat model GLM dengan distribusi Poisson

X = sm.add\_constant(X) # Tambahkan konstanta untuk intersep

model = sm.GLM(y, X, family=sm.families.Poisson()).fit()

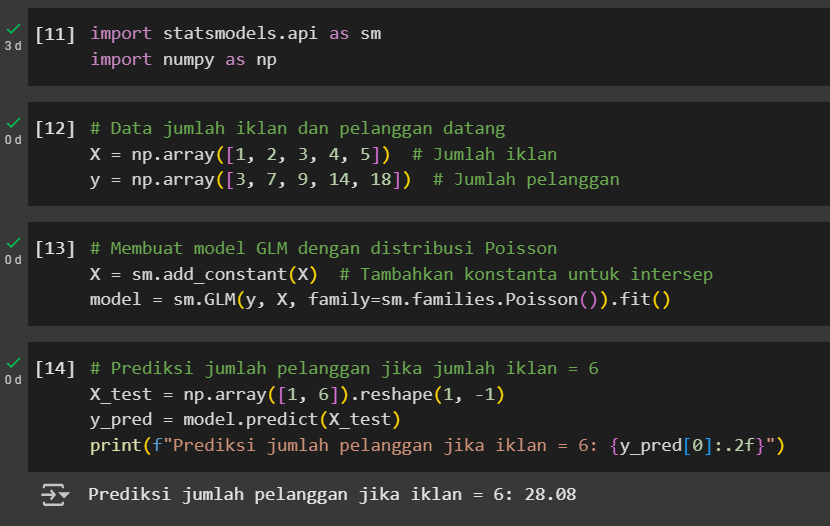
# Prediksi jumlah pelanggan jika jumlah iklan = 6

X\_test = np.array([1, 6]).reshape(1, -1)

y\_pred = model.predict(X\_test)

print(f"Prediksi jumlah pelanggan jika iklan = 6: {y\_pred[0]:.2f}")

**Berikut implementasinya di Google Colab:**



### Kapan Menggunakan GLM?

* Jika data **tidak berdistribusi normal** (contoh: data count atau proporsi).
* Jika ingin memilih fungsi link yang lebih fleksibel.

## Tabel Perbedaan

Berikut tabel perbedaan Regresi Linear, Logistic Regression, dan Generalized Linear Model (GLM)

|  |  |  |  |
| --- | --- | --- | --- |
| **Aspek** | **Regresi Linear** | **Logistic Regression** | **Generalized Linear Model (GLM)** |
| **Jenis Model** | Regresi | Klasifikasi | Regresi Generalisasi |
| **Tujuan** | Memprediksi nilai kontinu berdasarkan hubungan linear. | Memprediksi kategori biner berdasarkan probabilitas. | Memodelkan hubungan variabel dengan distribusi eksponensial. |
| **Jenis Data Target (Y)** | Variabel Kontinu (misalnya: harga, suhu, pendapatan). | Variabel Kategorikal Biner (misalnya: lulus/gagal, sehat/sakit). | Variabel Kontinu atau Diskrit (misalnya: jumlah pelanggan, kecelakaan). |
| **Fungsi Aktivasi** | Identitas (Linear). | Sigmoid (Logit Function). | Bergantung pada distribusi (log, identity, logit, dll). |
| **Distribusi Data** | Normal (Gaussian) | Binomial | Normal, Binomial, Poisson, Gamma, dll. |
| **Persamaan Dasar** | Y = β0 + β1X1 + β2X2 + ... + βnXn + ε | P(Y=1) = 1 / (1 + e^-(β0 + β1X1 + ... + βnXn)) | g(E(Y)) = β0 + β1X1 + ... + βnXn, dengan g sebagai fungsi link. |
| **Hasil Output** | Nilai numerik kontinu. | Probabilitas antara 0 dan 1, diklasifikasikan ke 0 atau 1. | Nilai numerik kontinu, count, atau proporsi. |
| **Metode Estimasi** | Ordinary Least Squares (OLS). | Maximum Likelihood Estimation (MLE). | Maximum Likelihood Estimation (MLE) untuk berbagai distribusi. |
| **Kapan Digunakan?** | Jika target kontinu dan hubungan linear. | Jika ingin klasifikasi biner dan menghitung probabilitas. | Jika data tidak berdistribusi normal. |
| **Contoh Kasus** | Prediksi harga rumah berdasarkan luas tanah. | Prediksi apakah email adalah spam atau bukan. | Prediksi jumlah pelanggan ke toko berdasarkan iklan. |
| **Kelebihan** | Mudah dipahami dan diimplementasikan. | Dapat digunakan untuk klasifikasi probabilistik. | Fleksibel untuk berbagai distribusi data. |
| **Kekurangan** | Tidak bisa digunakan jika hubungan tidak linear. | Hanya menangani klasifikasi biner, tidak multi-kelas. | Lebih kompleks dan butuh pemahaman distribusi. |

# Perhatikan parameter-parameter yang dimiliki oleh algoritma-algoritma pembelajaran terbimbing yang telah dijelaskan pada bab 2 ini, berikan penjelasan makna dari parameter-parameter tersebut!

## A. Regresi Linear di RapidMiner

Operator Regresi Linear di RapidMiner digunakan untuk memodelkan hubungan linear antara variabel dependen (target) dan variabel independen (fitur). Berikut adalah parameter-parameter penting yang perlu diperhatikan:

### 1. Feature Selection

* Ini adalah parameter tingkat ahli yang menentukan metode seleksi fitur yang akan digunakan selama regresi. Ada beberapa opsi yang tersedia:
  + **none**: Tidak menggunakan seleksi fitur.
  + **M5 prime**: Menggunakan metode M5 prime untuk seleksi fitur.
  + **greedy**: Menggunakan metode greedy untuk seleksi fitur.
  + **T-Test**: Menggunakan uji statistik T-Test untuk seleksi fitur.
  + **iterative T-Test**: Menggunakan T-Test secara iteratif untuk seleksi fitur.
* Seleksi fitur membantu memilih fitur yang paling relevan untuk model, sehingga meningkatkan performa dan mengurangi kompleksitas.

### 2. Alpha

* Parameter ini hanya tersedia jika feature selection diatur ke T-Test. Ini menentukan nilai alpha (tingkat signifikansi) yang digunakan dalam uji T-Test.
* Nilai alpha yang lebih kecil akan membuat seleksi fitur lebih ketat, hanya memilih fitur yang sangat signifikan.

### 3. Max Iterations

* Parameter ini hanya tersedia jika feature selection diatur ke iterative T-Test. Ini menentukan jumlah maksimum iterasi yang dilakukan dalam proses seleksi fitur iteratif.
* Iterasi yang lebih banyak memungkinkan algoritma untuk mengevaluasi lebih banyak kombinasi fitur, tetapi juga membutuhkan waktu lebih lama.

### 4. Forward Alpha

* Parameter ini hanya tersedia jika feature selection diatur ke iterative T-Test. Ini menentukan nilai alpha yang digunakan saat menambahkan fitur ke model (forward selection).
* Mengontrol seberapa ketat algoritma dalam menambahkan fitur baru selama iterasi.

### 5. Backward Alpha

* Parameter ini hanya tersedia jika feature selection diatur ke iterative T-Test. Ini menentukan nilai alpha yang digunakan saat menghapus fitur dari model (backward selection).
* Mengontrol seberapa ketat algoritma dalam menghapus fitur yang tidak signifikan selama iterasi.

### 6. Eliminate Colinear Features

* Parameter ini menentukan apakah algoritma akan mencoba menghapus fitur yang memiliki kolinearitas tinggi (korelasi tinggi) selama regresi.
* Menghapus fitur yang berkorelasi tinggi dapat meningkatkan stabilitas model dan mengurangi overfitting.

### 7. Min Tolerance

* Parameter ini hanya tersedia jika eliminate colinear features diatur ke true. Ini menentukan toleransi minimum untuk menghapus fitur yang berkolinear.
* Nilai toleransi yang lebih kecil akan membuat algoritma lebih ketat dalam menghapus fitur yang berkorelasi.

### 8. Use Bias

* Parameter ini menentukan apakah model akan menghitung nilai intercept atau tidak.
* Jika diaktifkan, model akan lebih fleksibel dalam menyesuaikan data. Jika tidak, garis regresi akan dipaksa melalui origin (0,0).

### 9. Ridge

* Parameter ini menentukan nilai parameter ridge yang digunakan dalam ridge regression.
* Ridge regression membantu mengurangi overfitting dengan menambahkan penalti pada koefisien model. Nilai ridge yang lebih besar akan meningkatkan efek regularisasi.

## B. Logistic Regression di RapidMiner

Operator Logistic Regression di RapidMiner digunakan untuk masalah klasifikasi biner atau multikelas. Berikut adalah parameter-parameter pentingnya:

### 1. Kernel Type

* Parameter ini memilih jenis fungsi kernel yang akan digunakan. Opsi yang tersedia:
  + **dot**: Kernel dot didefinisikan sebagai k(x,y)=x⋅y*k*(*x*,*y*)=*x*⋅*y* (produk dalam dari x*x* dan y*y*).
  + **radial**: Kernel radial didefinisikan sebagai exp⁡(−g∥x−y∥2)exp(−*g*∥*x*−*y*∥2), di mana g*g* adalah gamma.
  + **polynomial**: Kernel polinomial didefinisikan sebagai k(x,y)=(x⋅y+1)d*k*(*x*,*y*)=(*x*⋅*y*+1)*d*, di mana d*d* adalah derajat polinomial.
  + **sigmoid**: Kernel sigmoid didefinisikan sebagai tanh⁡(a⋅x⋅y+b)tanh(*a*⋅*x*⋅*y*+*b*), di mana a*a* adalah alpha dan b*b* adalah konstanta intercept.
  + **anova**: Kernel anova didefinisikan sebagai (∑exp⁡(−g(x−y)))d(∑exp(−*g*(*x*−*y*)))*d*, di mana g*g* adalah gamma dan d*d* adalah derajat.
  + **epachnenikov**: Kernel Epanechnikov didefinisikan sebagai 34(1−u2)43​(1−*u*2) untuk u*u* antara -1 dan 1.
  + **gaussian combination**:  
    Kernel kombinasi Gaussian.
  + **multiquadric**: Kernel multiquadric didefinisikan sebagai ∥x−y∥2+c2∥*x*−*y*∥2+*c*2​.
* Pemilihan kernel memengaruhi bagaimana data dipetakan ke ruang fitur yang lebih tinggi, yang dapat meningkatkan performa model.

### 2. Kernel Gamma

* Parameter gamma untuk kernel radial atau anova.
* Gamma mengontrol seberapa jauh pengaruh satu titik data terhadap titik lainnya. Nilai gamma yang besar membuat model lebih sensitif terhadap data.

### 3. Kernel Sigma1, Sigma2, Sigma3

* Parameter sigma untuk kernel epachnenikov, gaussian combination, atau multiquadric.
* Parameter ini mengontrol bentuk dan skala dari kernel.

### 4. Kernel Shift

* Parameter shift untuk kernel multiquadric.
* Menggeser kernel untuk menyesuaikan distribusi data.

### 5. Kernel Degree

* Parameter derajat untuk kernel polynomial, anova, atau epachnenikov.
* Derajat yang lebih tinggi memungkinkan model untuk menangkap hubungan yang lebih kompleks dalam data.

### 6. Kernel A dan B

* Parameter alpha (*a*) dan intercept (*b*) untuk kernel sigmoid.
* Parameter ini mengontrol bentuk dari fungsi sigmoid.

### 7. C (Complexity Constant)

* Konstanta kompleksitas yang mengatur toleransi untuk misklasifikasi. Nilai C yang lebih tinggi memungkinkan batas yang lebih "lunak", sedangkan nilai C yang lebih rendah menciptakan batas yang lebih "keras".
* Nilai C yang terlalu besar dapat menyebabkan overfitting, sedangkan nilai yang terlalu kecil dapat menyebabkan underfitting.

### 8. Start Population Type

* Menentukan jenis inisialisasi populasi awal.
* Inisialisasi populasi yang baik dapat mempercepat konvergensi algoritma.

### 9. Max Generations

* Menentukan jumlah generasi maksimum sebelum algoritma dihentikan.
* Mengontrol berapa lama algoritma akan berjalan.

### 10. Generations Without Improval

* Menentukan kriteria berhenti awal, yaitu berhenti setelah n generasi tanpa peningkatan performa.
* Membantu menghentikan algoritma jika tidak ada peningkatan lebih lanjut.

### 11. Population Size

* Menentukan ukuran populasi, yaitu jumlah individu per generasi. Jika diatur ke -1, semua contoh akan dipilih.
* Ukuran populasi yang lebih besar dapat meningkatkan diversitas solusi, tetapi juga membutuhkan lebih banyak komputasi.

### 12. Tournament Fraction

* Menentukan fraksi populasi saat ini yang akan digunakan sebagai anggota turnamen.
* Mengontrol seleksi individu untuk reproduksi.

### 13. Keep Best

* Menentukan apakah individu terbaik harus bertahan ke generasi berikutnya (elitist selection).
* Memastikan solusi terbaik tidak hilang selama evolusi.

### 14. Mutation Type

* Menentukan jenis operator mutasi.
* Mutasi membantu menjaga diversitas populasi.

### 15. Selection Type

* Menentukan skema seleksi untuk algoritma evolusioner.
* Skema seleksi memengaruhi bagaimana individu dipilih untuk reproduksi.

### 16. Crossover Prob

* Menentukan probabilitas individu untuk dipilih untuk crossover.
* Mengontrol seberapa sering crossover terjadi.

### 17. Use Local Random Seed

* Menentukan apakah seed acak lokal akan digunakan untuk randomisasi.
* Menggunakan seed yang sama akan menghasilkan randomisasi yang sama, yang berguna untuk reproduktibilitas.

### 18. Local Random Seed

* Menentukan seed acak lokal. Hanya tersedia jika use local random seed diatur ke true.
* Memastikan hasil yang konsisten.

### 19. Show Convergence Plot

* Menentukan apakah plot konvergensi akan ditampilkan.
* Membantu memantau performa algoritma selama pelatihan.

## C. Generalized Linear Models (GLM) di RapidMiner

Operator GLM di RapidMiner adalah versi yang lebih fleksibel, mendukung berbagai distribusi dan fungsi link. Berikut adalah parameter-parameter pentingnya:

**Penjelasan Parameter GLM di RapidMiner**

### 1. Family

* Parameter ini menentukan distribusi keluarga eksponensial yang digunakan. Opsi yang tersedia:
  + **AUTO**: Pemilihan otomatis. Menggunakan multinomial untuk label polinomial, binomial untuk label binomial, dan gaussian untuk label numerik.
  + **gaussian**: Data harus numerik (real atau integer).
  + **binomial**: Data harus binomial atau polinomial dengan 2 level/kelas.
  + **multinomial**: Data harus polinomial dengan lebih dari dua level/kelas.
  + **poisson**: Data harus numerik dan non-negatif (integer).
  + **gamma**: Data harus numerik, kontinu, dan positif (real atau integer).
  + **tweedie**: Data harus numerik, kontinu (real), dan non-negatif.
* Pemilihan family yang tepat sangat penting untuk memastikan model sesuai dengan jenis data yang digunakan.

### 2. Solver

* Memilih solver yang digunakan untuk optimasi. Opsi yang tersedia:
  + **AUTO**: Pemilihan otomatis.
  + **IRLSM**: Cepat untuk masalah dengan jumlah prediktor kecil dan untuk pencarian lambda dengan penalti L1.
  + **L\_BFGS**: Lebih baik untuk dataset dengan banyak kolom.
  + **COORDINATE\_DESCENT** (eksperimental): IRLSM dengan pembaruan kovarians.
  + **COORDINATE\_DESCENT\_NAIVE** (eksperimental): IRLSM dengan pembaruan naif.
* Pemilihan solver memengaruhi kecepatan dan efisiensi model.

### 3. Link

* Fungsi link menghubungkan prediktor linear dengan fungsi distribusi. Opsi yang tersedia:
  + **family\_default**: Menggunakan link kanonik untuk family yang dipilih.
  + **identity**: Untuk family gaussian, poisson, dan gamma.
  + **log**: Untuk family gaussian, poisson, dan gamma.
  + **inverse**: Untuk family gaussian dan gamma.
* Fungsi link yang tepat memastikan hubungan yang sesuai antara prediktor dan distribusi.

### 4. Reproducible

* Membuat pembangunan model dapat direproduksi. Jika diaktifkan, parameter **maximum\_number\_of\_threads** mengontrol tingkat paralelisme.
* Memastikan hasil yang konsisten saat model dibangun ulang.

### 5. Maximum Number of Threads

* Mengontrol tingkat paralelisme dalam pembangunan model.
* Menentukan seberapa banyak sumber daya komputasi yang digunakan.

### 6. Specify Beta Constraints

* Jika diaktifkan, batasan beta untuk atribut reguler dapat diberikan.
* Membatasi nilai koefisien untuk memastikan interpretasi yang lebih baik.

### 7. Use Regularization

* Mengaktifkan regularisasi. Jika diaktifkan, parameter lambda, alpha, dan pencarian lambda dapat ditentukan.
* Regularisasi membantu mengurangi overfitting.

### 8. Lambda

* Parameter lambda mengontrol jumlah regularisasi yang diterapkan. Jika lambda adalah 0.0, tidak ada regularisasi yang diterapkan.
* Nilai lambda yang lebih besar meningkatkan efek regularisasi.

### 9. Lambda Search

* Jika diaktifkan, pencarian lambda akan dilakukan mulai dari lambda maksimum.
* Membantu menemukan nilai lambda yang optimal.

### 10. Number of Lambdas

* Jumlah nilai lambda yang akan dicari jika lambda search diaktifkan.
* Menentukan seberapa banyak nilai lambda yang akan dievaluasi.

### 11. Lambda Min Ratio

* Nilai lambda minimum sebagai fraksi dari lambda.max.
* Mengontrol rentang nilai lambda yang akan dicari.

### 12. Early Stopping

* Menghentikan pencarian lambda lebih awal berdasarkan parameter stopping rounds dan stopping tolerance.
* Menghemat waktu komputasi jika model sudah konvergen.

### 13. Stopping Rounds

* Jumlah iterasi tanpa peningkatan sebelum pencarian dihentikan.
* Mengontrol kapan pencarian dihentikan.

### 14. Stopping Tolerance

* Toleransi relatif untuk kriteria penghentian berbasis metrik.
* Menentukan seberapa kecil peningkatan yang dianggap signifikan.

### 15. Alpha

* Parameter alpha mengontrol distribusi antara penalti L1 (Lasso) dan L2 (Ridge regression). Nilai 1.0 untuk alpha menghasilkan Lasso, sedangkan nilai 0.0 menghasilkan Ridge regression.
* Mengontrol jenis regularisasi yang diterapkan.

### 16. Standardize

* Menstandarisasi kolom numerik untuk memiliki mean nol dan varian satu.
* Memastikan semua fitur memiliki skala yang sama.

### 17. Non-Negative Coefficients

* Membatasi koefisien (bukan intercept) untuk menjadi non-negatif.
* Berguna untuk interpretasi model yang lebih baik.

### 18. Compute P-Values

* Menghitung nilai p. Hanya bekerja dengan solver IRLSM dan tanpa regularisasi.
* Memberikan informasi statistik tentang signifikansi koefisien.

### 19. Remove Collinear Columns

* Menghapus kolom yang bergantung linear jika ada.
* Meningkatkan stabilitas model.

### 20. Add Intercept

* Menambahkan konstanta ke model.
* Memungkinkan model untuk lebih fleksibel.

### 21. Missing Values Handling

* Menangani nilai yang hilang. Opsi yang tersedia:
  + Melewatkan nilai yang hilang.
* Mengganti nilai yang hilang dengan nilai rata-rata.
* Memastikan data yang digunakan lengkap.

### 22. Max Iterations

* Jumlah maksimum iterasi untuk pelatihan model.
* Mengontrol berapa lama algoritma akan berjalan.

### 23. Beta Constraints

* Batasan untuk nilai beta. Setiap baris berisi nama atribut, kategori, batas bawah, batas atas, dan nilai beta yang diberikan.
* Membatasi nilai koefisien untuk interpretasi yang lebih baik.

### 24. Max Runtime Seconds

* Waktu maksimum yang diizinkan untuk pelatihan model dalam detik.
* Mengontrol durasi pelatihan.

### 25. Expert Parameters

* Parameter ini untuk penyesuaian lebih lanjut. Biasanya nilai default sudah cukup baik.
* Untuk penyesuaian lebih lanjut jika diperlukan.

## Kesimpulan

* Regresi Linear di RapidMiner fokus pada pemodelan hubungan linear dengan parameter seperti penggunaan intercept dan penanganan multikolinearitas.
* Logistic Regression digunakan untuk klasifikasi dengan parameter seperti regularisasi dan iterasi maksimum.
* GLM adalah yang paling fleksibel, mendukung berbagai distribusi dan fungsi link, dengan parameter seperti pemilihan keluarga distribusi dan fungsi link.

# Dengan menggunakan parameter-parameter yang telah saudara berikan penjelasan, ubahlah semua model yang ada di bab 2 ini dengan parameter tersebut, carilah parameter yang menghasilkan akurasi terbaik!

# Referensi

* <https://images.app.goo.gl/3HdSSQYbvA19gUXc9>
* [https://miro.medium.com/v2/resize:fit:1046/1\*l59BUnPwWHMf1H-GNxgZHQ.png](https://miro.medium.com/v2/resize:fit:1046/1*l59BUnPwWHMf1H-GNxgZHQ.png)
* <https://docs.rapidminer.com/2025.0/studio/operators/modeling/predictive/functions/linear_regression.html>
* <https://docs.rapidminer.com/2025.0/studio/operators/modeling/predictive/logistic_regression/logistic_regression_evolutionary.html>
* <https://docs.rapidminer.com/2025.0/studio/operators/modeling/predictive/functions/generalized_linear_model.html>