HANDOUT MAHASISWA

Regresi Linear untuk Prediksi Variabel Kontinu

Mata Kuliah: Data Science

Program Studi Teknik Informatika
Universitas Muhammadiyah Pontianak
Dosen: Yulrio Brianorman, S.Si., M.T.

Tahun Akademik 2025/2026

Ringkasan

Handout ini merupakan ringkasan materi Regresi Linear yang mencakup konsep dasar, formulasi matematis, metrik evaluasi, asumsi-asumsi penting, dan implementasi praktis. Dokumen ini dirancang sebagai referensi cepat untuk mahasiswa dalam memahami dan mengimplementasikan model regresi linear untuk prediksi variabel kontinu.

1 Pengenalan

1.1 Apa itu Regresi Linear?

Poin Penting

Regresi Linear adalah metode statistik untuk memodelkan hubungan **linear** antara variabel dependen (target) dengan satu atau lebih variabel independen (prediktor).

Karakteristik Utama:

- Supervised Learning (memerlukan label/target)
- Prediksi nilai kontinu (bukan kategori)
- Model parametrik (memiliki parameter yang dipelajari)
- Interpretable (mudah diinterpretasikan)

1.2 Jenis Regresi Linear

1.2.1 Simple Linear Regression

Model dengan **satu variabel prediktor**:

$$y = \beta_0 + \beta_1 x + \epsilon$$

Contoh

Prediksi harga rumah berdasarkan luas bangunan saja.

1.2.2 Multiple Linear Regression

Model dengan **banyak variabel prediktor**:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$$

Contoh

Prediksi harga rumah berdasarkan luas bangunan, jumlah kamar, umur bangunan, dan lokasi.

2 Formulasi Matematis

2.1 Model Regresi Linear

Formula

Bentuk Umum:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

Bentuk Matriks:

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}$$

Notasi:

- $\hat{y}$: Nilai prediksi (predicted value)
- β_0: Intercept (konstanta/bias)
- $\beta_1, ..., \beta_n$: Koefisien regresi (weights)
- $x_1, ..., x_n$: Variabel independen (features)
- ϵ: Error/residual

2.2 Interpretasi Parameter

Intercept (β_0):

- Nilai prediksi ketika semua $x_i = 0$
- Baseline atau nilai dasar

Koefisien (β_i):

- Perubahan rata-rata pada y untuk setiap 1 unit perubahan x_i
- Dengan asumsi variabel lain konstan (*ceteris paribus*)
- Tanda (+/-) menunjukkan arah hubungan

Contoh

Jika model prediksi harga rumah adalah:

$$\text{Harga} = 50 + 10 \times \text{Luas} + 30 \times \text{Kamar}$$

Interpretasi:

- Harga dasar: Rp 50 juta
- Setiap penambahan 1 m^2 luas $\rightarrow$ harga naik Rp 10 juta
- Setiap penambahan 1 kamar $\rightarrow$ harga naik Rp 30 juta

2.3 Cost Function: Mean Squared Error

Formula

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Tujuan: Minimumkan MSE untuk menemukan parameter optimal $\boldsymbol{\beta}$
Mengapa kuadrat?

- Error positif dan negatif tidak saling menghilangkan
- Memberikan penalti lebih besar untuk error besar (outlier)
- Sifat matematis baik (differentiable, convex)

2.4 Metode Optimasi

2.4.1 Ordinary Least Squares (OLS)

Formula

Normal Equation:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Kelebihan:

- Solusi eksak dalam satu langkah
- Tidak perlu hyperparameter tuning

Kekurangan:

- Komputasi mahal untuk data besar: $O(n^3)$
- Memerlukan $\mathbf{X}^T\mathbf{X}$ invertible

2.4.2 Gradient Descent

Formula

Update Rule:

$$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} MSE$$

Kelebihan:

- Cocok untuk dataset besar
- Lebih memory-efficient

Kekurangan:

- Memerlukan hyperparameter tuning (α)
- Iteratif, butuh banyak epoch

3 Metrik Evaluasi

3.1 Mean Squared Error (MSE)

Formula

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- **Satuan**: Kuadrat dari satuan target
- **Range**: $[0, \infty)$
- **Interpretasi**: Semakin kecil semakin baik
- **Sensitif** terhadap outlier

3.2 Root Mean Squared Error (RMSE)

Formula

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

- **Satuan**: Sama dengan target
- **Interpretasi**: Rata-rata error tipikal
- Lebih mudah diinterpretasikan dibanding MSE

3.3 Mean Absolute Error (MAE)

Formula

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **Satuan**: Sama dengan target
- **Lebih robust** terhadap outlier dibanding MSE/RMSE
- Semua error diberi bobot sama

3.4 Coefficient of Determination (R^2)

Formula

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} = 1 - \frac{SS_{res}}{SS_{tot}}$$

- **Range**: $(-\infty, 1]$ (biasanya $[0, 1]$)
- **Interpretasi**:
  - $R^2 = 1$: Model sempurna
  - $R^2 = 0$: Model tidak lebih baik dari mean
  - $R^2 < 0$: Model lebih buruk dari mean
- $R^2 = 0.85 \rightarrow$ model menjelaskan 85% variansi data

Perhatian!

Jangan hanya fokus pada R^2 tinggi! Model bisa overfit dengan R^2 tinggi di training tapi rendah di testing.

4 Asumsi Regresi Linear

4.1 1. Linearitas

Definisi: Hubungan antara X dan y bersifat linear
Validasi: Scatter plot, residual plot
Jika dilanggar:

- Transformasi variabel (log, sqrt, polynomial)
- Gunakan model non-linear

4.2 2. Independence

Definisi: Observasi independen satu sama lain
Validasi: Durbin-Watson test, plot residual vs time
Jika dilanggar:

- Gunakan time series models
- Mixed-effects models

4.3 3. Homoscedasticity

Definisi: Variansi residual konstan untuk semua nilai prediksi
Validasi: Residual plot (harus acak, tidak berpola)
Jika dilanggar:

- Transformasi target (log, sqrt)
- Weighted Least Squares

4.4 4. Normalitas Residual

Definisi: Residual mengikuti distribusi normal
Validasi: Q-Q plot, histogram residual, Shapiro-Wilk test
Jika dilanggar:

- Transformasi target
- Hapus outlier ekstrem

4.5 5. No Multicollinearity

Definisi: Prediktor tidak berkorelasi tinggi satu sama lain
Validasi:

- Correlation matrix (korelasi > 0.8 = masalah)
- Variance Inflation Factor (VIF > 10 = masalah)

Formula

$$VIF_i = \frac{1}{1 - R_i^2}$$

Jika dilanggar:

- Hapus salah satu fitur berkorelasi tinggi
- PCA (Principal Component Analysis)
- Regularisasi (Ridge, Lasso)

5 Regularisasi

5.1 Mengapa Perlu Regularisasi?

Perhatian!

Overfitting: Model terlalu kompleks, performa baik di training tapi buruk di testing.

Solusi: Tambahkan penalty term pada loss function untuk membatasi magnitude koefisien.

5.2 Ridge Regression (L2 Regularization)

Formula

$$Loss = MSE + \alpha \sum_{j=1}^{p} \beta_j^2$$

Karakteristik:

- Menyusutkan koefisien mendekati 0 (tidak tepat 0)
- Cocok ketika semua fitur berkontribusi
- Menangani multicollinearity dengan baik

5.3 Lasso Regression (L1 Regularization)

Formula

$$Loss = MSE + \alpha \sum_{j=1}^{p} |\beta_j|$$

Karakteristik:

- Dapat membuat koefisien = 0 (feature selection)
- Menghasilkan sparse model
- Cocok ketika banyak fitur tidak relevan

5.4 Perbandingan Ridge vs Lasso

Aspek	Ridge	Lasso
Penalty	$\sum \beta_j^2$	$\sum \|\beta_j\|$
Koefisien = 0?	Tidak	Ya
Feature Selection	Tidak	Ya
Multicollinearity	Menangani baik	Pilih satu

6 Implementasi Praktis dengan Python

6.1 Setup Environment

```
# Import libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

6.2 Workflow Standar

1. Load dan Eksplorasi Data

```
data = pd.read_csv('dataset.csv')
print(data.head())
print(data.info())
print(data.describe())
```

2. Preprocessing

```
# Handle missing values
data = data.dropna()

# Feature scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

3. Model Training

```
# Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)

# Lihat koefisien
print(f"Intercept: {model.intercept_}")
print(f"Coefficients: {model.coef_}")
```

4. Evaluasi

```
# Prediksi
y_pred = model.predict(X_test)

# Metrik
mse = mean_squared_error(y_test, y_pred)
```

```
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.4f}")
```

5. **Visualisasi**

```
# Actual vs Predicted
plt.scatter(y_test, y_pred)
plt.plot([y_test.min(), y_test.max()],
         [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()

# Residual plot
residuals = y_test - y_pred
plt.scatter(y_pred, residuals)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel('Predicted')
plt.ylabel('Residuals')
plt.show()
```

7 Tips dan Best Practices

7.1 Data Preprocessing

Poin Penting

Garbage In, Garbage Out: Kualitas model sangat bergantung pada kualitas data!

1. **Handle Missing Values**
   - Mean/median imputation untuk numerik
   - Mode untuk kategorikal
   - Hapus jika terlalu banyak missing (>30%)
2. **Handle Outliers**
   - Identifikasi: IQR method, Z-score
   - Treatment: Remove, cap, atau transform
3. **Feature Scaling**
   - Standardization: mean=0, std=1
   - Normalization: range 0-1
   - Penting untuk Gradient Descent dan Regularisasi

7.2 Model Validation

Perhatian!

JANGAN PERNAH evaluasi model di training data!

Train-Test Split:

- Umum: 70-80% training, 20-30% testing
- Random state untuk reproducibility

Cross-Validation:

- K-Fold CV (K=5 atau 10)
- Lebih robust daripada single split
- Gunakan untuk hyperparameter tuning

7.3 Interpretasi Hasil

Yang harus dilaporkan:

1. Model equation dengan koefisien
2. Interpretasi setiap koefisien
3. Metrik evaluasi (RMSE, R^2)
4. Validasi asumsi (residual plot)
5. Limitasi dan rekomendasi

8 Kesalahan Umum dan Cara Menghindarinya

8.1 Kesalahan 1: Overfitting

Tanda-tanda:

- R^2 training tinggi, testing rendah
- Model terlalu kompleks (banyak fitur)

Solusi:

- Gunakan regularisasi (Ridge/Lasso)
- Feature selection
- Cross-validation

8.2 Kesalahan 2: Multicollinearity

Tanda-tanda:

- Koefisien tidak stabil
- VIF > 10
- High correlation antara features

Solusi:

- Hapus salah satu fitur berkorelasi
- Gunakan Ridge regression
- PCA

8.3 Kesalahan 3: Tidak Validasi Asumsi

Dampak:

- Hasil inferensi tidak valid
- Model tidak reliable

Solusi:

- Selalu cek residual plot
- Lakukan diagnostic tests
- Transform data jika perlu

9 Checklist Sebelum Submit Tugas

☐ Data sudah bersih (no missing, outliers handled)

☐ Feature scaling sudah dilakukan

☐ Train-test split sudah benar

☐ Model sudah ditraining dengan benar

☐ Evaluasi menggunakan test set (bukan training)

☐ Metrik sudah dihitung (MSE, RMSE, R^2)

☐ Visualisasi sudah ada (actual vs predicted, residual plot)

☐ Asumsi sudah divalidasi

☐ Interpretasi koefisien sudah dijelaskan

☐ Kesimpulan dan rekomendasi sudah ditulis

☐ Code sudah bersih dan commented

☐ Notebook sudah di-export ke PDF

10 Referensi Belajar

10.1 Textbook (Gratis Online)

1. **An Introduction to Statistical Learning**
   James, G., et al. (2021)
   https://www.statlearning.com/

2. **The Elements of Statistical Learning**
   Hastie, T., et al. (2009)
   https://hastie.su.domains/ElemStatLearn/

10.2 Online Courses

1. **Kaggle Learn - Intro to Machine Learning**
   https://www.kaggle.com/learn/intro-to-machine-learning

2. **StatQuest with Josh Starmer** (YouTube)
   Penjelasan konsep dengan animasi yang bagus

10.3 Dokumentasi

1. **Scikit-learn Documentation**
   https://scikit-learn.org/stable/modules/linear_model.html

2. **Pandas Documentation**
   https://pandas.pydata.org/docs/

10.4 Dataset Practice

1. Kaggle: https://www.kaggle.com/datasets

2. UCI ML Repository: https://archive.ics.uci.edu/

3. Scikit-learn built-in datasets

11 FAQ (Frequently Asked Questions)

Q: Apa bedanya regresi dan klasifikasi?
A: Regresi memprediksi nilai kontinu (harga, suhu), klasifikasi memprediksi kategori (spam/not spam, jenis bunga).

Q: Apakah regresi linear hanya untuk data linear?
A: Untuk model linear, tapi bisa diterapkan pada data non-linear dengan feature engineering (polynomial, log transform).

Q: Mengapa perlu split train-test?
A: Untuk mendeteksi overfitting dan mengukur performa model pada data baru yang belum pernah dilihat.

Q: Kapan menggunakan Ridge vs Lasso?
A: Ridge ketika semua fitur relevan. Lasso ketika banyak fitur tidak relevan dan ingin feature selection otomatis.

Q: Apa artinya R^2 negatif?
A: Model lebih buruk dari hanya menggunakan mean sebagai prediksi. Indikasi model sangat buruk atau salah implementasi.

Q: Apakah $R^2 = 0.9$ selalu bagus?
A: Tidak selalu! Bisa jadi overfitting. Harus cek R^2 di test set juga.

Selamat Belajar!
Jika ada pertanyaan, jangan ragu untuk bertanya saat kuliah atau praktikum.
Good luck with your Data Science journey!
