

Feature Importance & Selection

**Dina Chahyati*, Adila Alfa Krisnadhi,
Siti Aminah, Aruni Yasmin Azizah, Fariz Darari**

**CSGE603130: Kecerdasan Artifisial dan Sains Data Dasar
Gasal 2022/2023**

Referensi

- [Understanding Feature Importance and How to Implement it in Python | by Terence Shin | Towards Data Science](#)
- [A Complete Guide to Sequential Feature Selection - \(analyticsindiamag.com\)](#)
- [Feature Selection Techniques | 7 Feature Selection Techniques in ML \(analyticsvidhya.com\)](#)
- [Using the Chi-Squared test for feature selection with implementation | by Dr. Saptarsi Goswami | Towards Data Science](#)



FAKULTAS
ILMU
KOMPUTER

Motivasi

Motivasi: Mengapa Perlu Seleksi Fitur?

- Untuk mengurangi **jumlah** fitur yang digunakan
- Untuk mencari **model terbaik** (*better performance*), karena terkadang ada fitur tak relevan yang membuat hasil klasifikasi/regresi/clustering tidak optimal
- Untuk mengurangi **beban komputasi**
- Untuk memperkuat *data understanding*
- Untuk memperkuat *model interpretability*
- Seleksi fitur dilakukan dengan cara membandingkan nilai **feature importance**

Metode Seleksi Fitur

A. Metode Filter

- Cepat dan mudah
- Dilakukan pada saat pra-pemrosesan data, **sebelum** data masuk ke algoritma *machine learning* (klasifikasi/regresi/clustering)
- Contoh: Koefisien korelasi, *variance threshold*, Chi-square-test, dll

B. Metode Wrapper (Greedy Algorithm)

- Umumnya berusaha memilih subset-subset dari himpunan fitur secara iteratif, dengan cara **membandingkan kinerjanya jika subset fitur tersebut dijadikan input** ke algoritma *machine learning*
- Contoh: SFFS, SBFS, dll

C. Metode Embedded

- Kombinasi metode filter dan wrapper. Cepat seperti metode filter dan hasilnya akurat seperti metode wrapper. Seleksi **fitur merupakan bagian dari algoritma** *machine learning* itu sendiri.
- Contoh: Decision Tree, Linear regression Lasso

A. Metode Filter

A. Metode Filter

- Cepat dan mudah
- Dilakukan pada saat pra-pemrosesan data, **sebelum** data masuk ke algoritma *machine learning* (klasifikasi/regresi/clustering)
- Contoh: koefisien korelasi, *variance threshold*, Chi-square-test, dll



- Klasifikasi / Regresi / Clustering hanya dilakukan **satu kali**

A. Metode Filter: Koefisien Korelasi

- Hitung korelasi antara setiap **fitur** dengan **label target**. Fitur dengan korelasi mendekati nol bisa diabaikan.
- Hitung korelasi **antar fitur**. Jika ada 2 fitur dengan korelasi tinggi, ambil salah satu saja.

Gambar diambil dari
[Feature Selection Techniques | 7 Feature Selection Techniques in ML \(analyticsvidhya.com\)](#)



- Apakah PCA masuk ke kategori ini?

A. Metode Filter: Variance Threshold

- Motivasi: data dengan variansi rendah cenderung tidak memberikan banyak informasi
- Cara seleksi:
 - Pastikan data berada pada skala yang sama (normalisasi)
 - Hitung variansi tiap data
 - Abaikan data dengan variansi rendah (berdasarkan threshold tertentu)

A. Metode Filter: Chi-Square Test

- Sebenarnya yang dilakukan oleh Chi-Square Test adalah **Uji Hipotesa**, untuk mengetahui apakah suatu fitur berpengaruh/tidak berpengaruh terhadap label kelas (klasifikasi).
- Digunakan untuk membandingkan dua data bertipe kategorikal.
- Ingat pada kuliah Statprob, Uji Hipotesa digunakan untuk membandingkan **mean**, sehingga menggunakan
 - Tabel distribusi normal (jika variansi populasi diketahui)
 - Tabel distribusi student-t (jika variansi populasi tidak diketahui)
- Pada seleksi fitur, yang ingin dibandingkan adalah **hasil observasi** (observed) dan **hasil yang diharapkan** (expected)
- Jika fitur yang dibandingkan bersifat numerik, uji hipotesa bisa dilakukan dengan F-test (ANOVA – ANalysis Of VAriance)

A. Metode Filter: Chi-Square Test

The Chi-Square Test for Association

Math 137
Fall '13
L. Burger

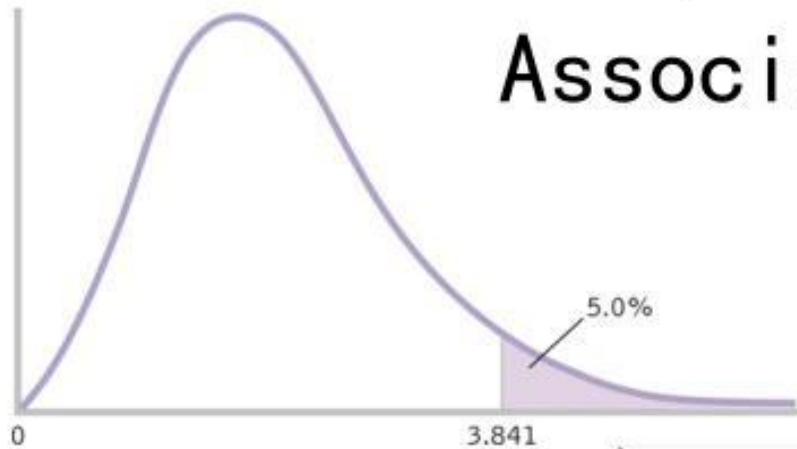


Table 5-2
Critical Values of the χ^2 Distribution

| df \ p | 0.995 | 0.975 | 0.9 | 0.5 | 0.1 | 0.05 | 0.025 | 0.01 | 0.005 | df |
|--------|-------|-------|-------|--------|--------|--------|--------|--------|--------|----|
| 1 | .000 | .000 | 0.016 | 0.455 | 2.706 | 3.841 | 5.024 | 6.635 | 7.879 | 1 |
| 2 | 0.010 | 0.051 | 0.211 | 1.386 | 4.605 | 5.991 | 7.378 | 9.210 | 10.597 | 2 |
| 3 | 0.072 | 0.216 | 0.584 | 2.366 | 6.251 | 7.815 | 9.348 | 11.345 | 12.838 | 3 |
| 4 | 0.207 | 0.484 | 1.064 | 3.357 | 7.779 | 9.488 | 11.143 | 13.277 | 14.860 | 4 |
| 5 | 0.412 | 0.831 | 1.610 | 4.351 | 9.236 | 11.070 | 12.832 | 15.086 | 16.750 | 5 |
| 6 | 0.676 | 1.237 | 2.204 | 5.348 | 10.645 | 12.592 | 14.449 | 16.812 | 18.548 | 6 |
| 7 | 0.989 | 1.690 | 2.833 | 6.346 | 12.017 | 14.067 | 16.013 | 18.475 | 20.278 | 7 |
| 8 | 1.344 | 2.180 | 3.490 | 7.344 | 13.362 | 15.507 | 17.535 | 20.090 | 21.955 | 8 |
| 9 | 1.735 | 2.700 | 4.168 | 8.343 | 14.684 | 16.919 | 19.023 | 21.666 | 23.589 | 9 |
| 10 | 2.156 | 3.247 | 4.865 | 9.342 | 15.987 | 18.307 | 20.483 | 23.209 | 25.188 | 10 |
| 11 | 2.603 | 3.816 | 5.578 | 10.341 | 17.275 | 19.675 | 21.920 | 24.725 | 26.757 | 11 |
| 12 | 3.074 | 4.404 | 6.304 | 11.340 | 18.549 | 21.026 | 23.337 | 26.217 | 28.300 | 12 |
| 13 | 3.565 | 5.009 | 7.042 | 12.340 | 19.812 | 22.362 | 24.736 | 27.688 | 29.819 | 13 |
| 14 | 4.075 | 5.629 | 7.790 | 13.339 | 21.064 | 23.685 | 26.119 | 29.141 | 31.319 | 14 |
| 15 | 4.601 | 6.262 | 8.547 | 14.339 | 22.307 | 24.996 | 27.488 | 30.578 | 32.801 | 15 |

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Gambar diambil dari
[PPT - The Chi-Square Test for Association PowerPoint Presentation, free download - ID:2536011 \(slideserve.com\)](#)

A. Metode Filter: Chi-Square Test

- Contoh Chi-Square Test ([Using the Chi-Squared test for feature selection with implementation | by Dr. Saptarsi Goswami | Towards Data Science](#))

| PId | Pclass | Name | Sex | Age | Fare | Survived |
|-----|--------|---|--------|-----|---------|----------|
| 1 | 3 | Braund, Mr. Owen Harris | male | 22 | 7.25 | 0 |
| 2 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Thayer) | female | 38 | 71.2833 | 1 |
| 3 | 3 | Heikkinen, Miss. Laina | female | 26 | 7.925 | 1 |
| 4 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35 | 53.1 | 1 |
| 5 | 3 | Allen, Mr. William Henry | male | 35 | 8.05 | 0 |
| 6 | 3 | Moran, Mr. James | male | | 8.4583 | 0 |
| 7 | 1 | McCarthy, Mr. Timothy J | male | 54 | 51.8625 | 0 |
| 8 | 3 | Palsson, Master. Gosta Leonard | male | 2 | 21.075 | 0 |
| 9 | 3 | Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) | female | 27 | 11.1333 | 1 |
| 10 | 2 | Nasser, Mrs. Nicholas (Adele Achem) | female | 14 | 30.0708 | 1 |

- Titanic Shipwreck Problem, ingin diketahui: Survived atau Not Survived
- Ingin diketahui pengaruh atribut “Sex” pada “Survived”

A. Metode Filter: Chi-Square Test

Tentukan pernyataan Hipotesa:

- H_0 : The attribute sex has no role to play in the survival of a passenger (The feature is not important)
- H_1 : The attribute sex has a role to play in survival (The feature is important)

A. Metode Filter: Chi-Square Test

| Sex | NotSurvied | Survived | Total |
|--------|------------|----------|-------|
| female | 156 | 307 | 463 |
| male | 708 | 142 | 850 |
| Total | 864 | 449 | 1313 |

$$\chi^2 = \frac{(O_{11} - E_{11})^2}{E_{11}} + \frac{(O_{12} - E_{12})^2}{E_{12}} + \dots + \frac{(O_{mn} - E_{mn})^2}{E_{mn}}$$

$$= \sum_{i=1}^m \sum_{j=1}^n \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

The above table shows Sex vs survival status of the passengers. This is also called the contingency table.

- If they are independent then the male-female ratio across survived and not survived should be the same.
- A ratio of 449/1313 survived
- If there is no dependence or association 463 * 449/1313 of the females should survive i.e 158.13
- This is the expected count under independence
- The calculation for all four cases are given below

$O_{female-survived}$

$E_{female-survived}$

| Categories | Observed | Expected | Chi-square |
|---------------------|----------|----------|------------|
| Female Survived | 307 | 158.33 | 139.5994 |
| Female Not Survived | 156 | 304.67 | 72.54659 |
| Male Survived | 142 | 290.67 | 76.04076 |
| Male not Survived | 708 | 559.33 | 39.51651 |
| | | | 327.7032 |

A. Metode Filter: Chi-Square Test

Selanjutnya:

- Hitung chi-square untuk setiap fitur kategorikal
- Sebenarnya bisa ditentukan nilai level of confidence tertentu untuk menentukan apakah H_0 ditolak atau diterima
- Namun bisa juga diurutkan saja nilai chi-square dari setiap fitur lalu diurutkan, ambil fitur-fitur berdasarkan nilai chi-square yang tertinggi.

A. Metode Filter di Sklearn

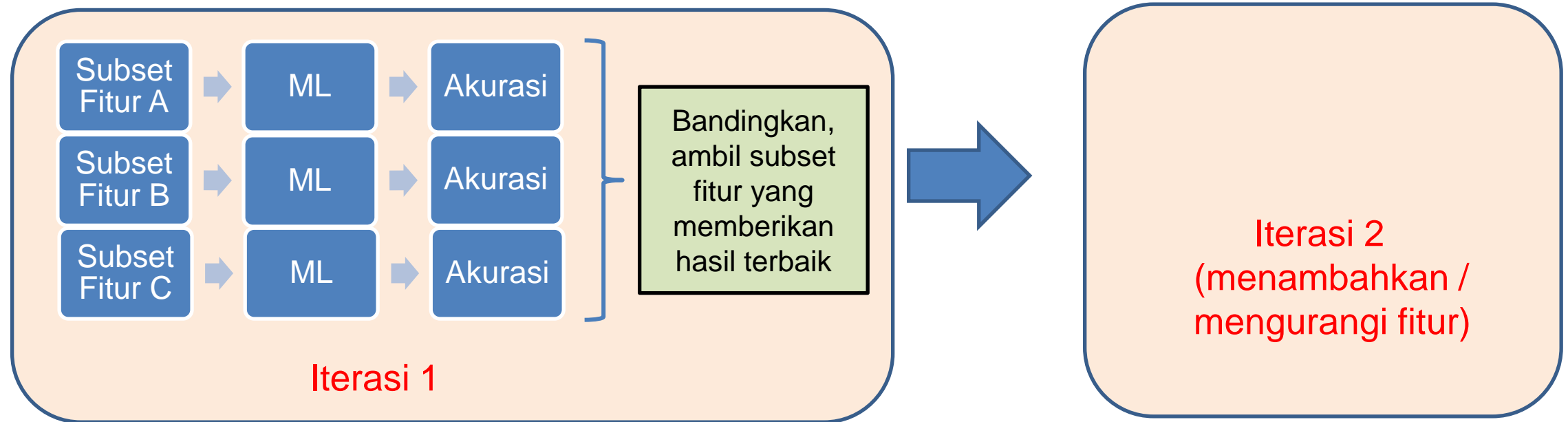
sklearn.feature_selection: Feature Selection

| | |
|--|--|
| <code>feature_selection.chi2(X, y)</code> | Compute chi-squared stats between each non-negative feature and class. |
| <code>feature_selection.f_classif(X, y)</code> | Compute the ANOVA F-value for the provided sample. |
| <code>feature_selection.f_regression(X, y, *[...])</code> | Univariate linear regression tests returning F-statistic and p-values. |
| <code>feature_selection.r_regression(X, y, *[...])</code> | Compute Pearson's r for each features and the target. |
| <code>feature_selection.mutual_info_classif(X, y, *)</code> | Estimate mutual information for a discrete target variable. |
| <code>feature_selection.mutual_info_regression(X, y, *)</code> | Estimate mutual information for a continuous target variable. |

B. Metode Wrapper

B. Metode Wrapper

- Umumnya berusaha memilih subset-subset dari himpunan fitur secara iteratif, dengan cara **membandingkan kinerjanya jika subset fitur tersebut dijadikan input** ke algoritma *machine learning*
- “Kinerja” bisa dihitung dengan akurasi, F1-Score, AUC, dlsb



- Klasifikasi / Regresi / Clustering dilakukan **berulang kali**

B. Metode Wrapper: SFFS

- SFS: Sequential Feature Selection
- Varian: Backward, Forward
- SFFS (Sequential **Forward** Feature Selection)
 - Misal ada 10 fitur (A,B,C,...,J)
 - Kondisi awal: **belum ada** fitur yang terpilih
 - Lakukan 10 eksperimen (klasifikasi/regresi/clustering) menggunakan **satu** fitur saja setiap eksperimen. Ambil fitur dengan akurasi tertinggi (misal fitur B)
 - Lakukan 9 eksperimen menggunakan 1 tambahan fitur (BA, BC, ..., BJ), ambil kombinasi fitur yang memberikan akurasi tertinggi.
 - Demikian seterusnya sehingga diperoleh k buah fitur yang diinginkan.
 - Misal kita menginginkan 3 fitur saja, maka jumlah eksperimen yang dilakukan adalah $10 + 9 + 8 = 27$ kali
 - Jika kita mencoba semua kemungkinan: $C(10,3) = 120$ eksperimen
 - **Sekuensial**: tidak bisa backtrack. Fitur yang sudah terpilih tidak bisa dibuang di langkah selanjutnya

B. Metode Wrapper: SBFS

- SBFS (Sequential **Backward** Feature Selection)
 - Misal ada 10 fitur (A,B,C,...,J)
 - Kondisi awal: **semua** fitur terpilih
 - Lakukan 10 eksperimen (klasifikasi/regresi/clustering) menggunakan **9 dari 10** fitur saja setiap eksperimen. Buang fitur yang memberikan akurasi terburuk.
 - Lakukan 9 eksperimen lagi, dengan membuang salah satu fitur di setiap eksperimen. Buang fitur yang memberikan akurasi terburuk
 - Demikian seterusnya sehingga diperoleh k buah fitur yang diinginkan.
 - **Sekuensial**: tidak bisa backtrack. Fitur yang sudah terbuang tidak bisa dipilih di langkah selanjutnya

hapus fitur yang punya least performance lost

D :

di jalanin sampai 10 kali kalau misalnya gak tau mau stop berapa kali, terus di cek kombinasi dengan akurasi paling baik.

B. Metode Wrapper di sklearn

`sklearn.feature_selection.SequentialFeatureSelector`

```
class sklearn.feature_selection.SequentialFeatureSelector(estimator, *, n_features_to_select='warn', tol=None, direction='forward', scoring=None, cv=5, n_jobs=None)
```

[\[source\]](#)

```
>>> from sklearn.feature_selection import SequentialFeatureSelector
>>> from sklearn.neighbors import KNeighborsClassifier
>>> from sklearn.datasets import load_iris
>>> X, y = load_iris(return_X_y=True)
>>> knn = KNeighborsClassifier(n_neighbors=3)
>>> sfs = SequentialFeatureSelector(knn, n_features_to_select=3)
>>> sfs.fit(X, y)
SequentialFeatureSelector(estimator=KNeighborsClassifier(n_neighbors=3),
                          n_features_to_select=3)
>>> sfs.get_support()
array([ True, False,  True,  True])
```



FAKULTAS
ILMU
KOMPUTER

C. Metode Embedded

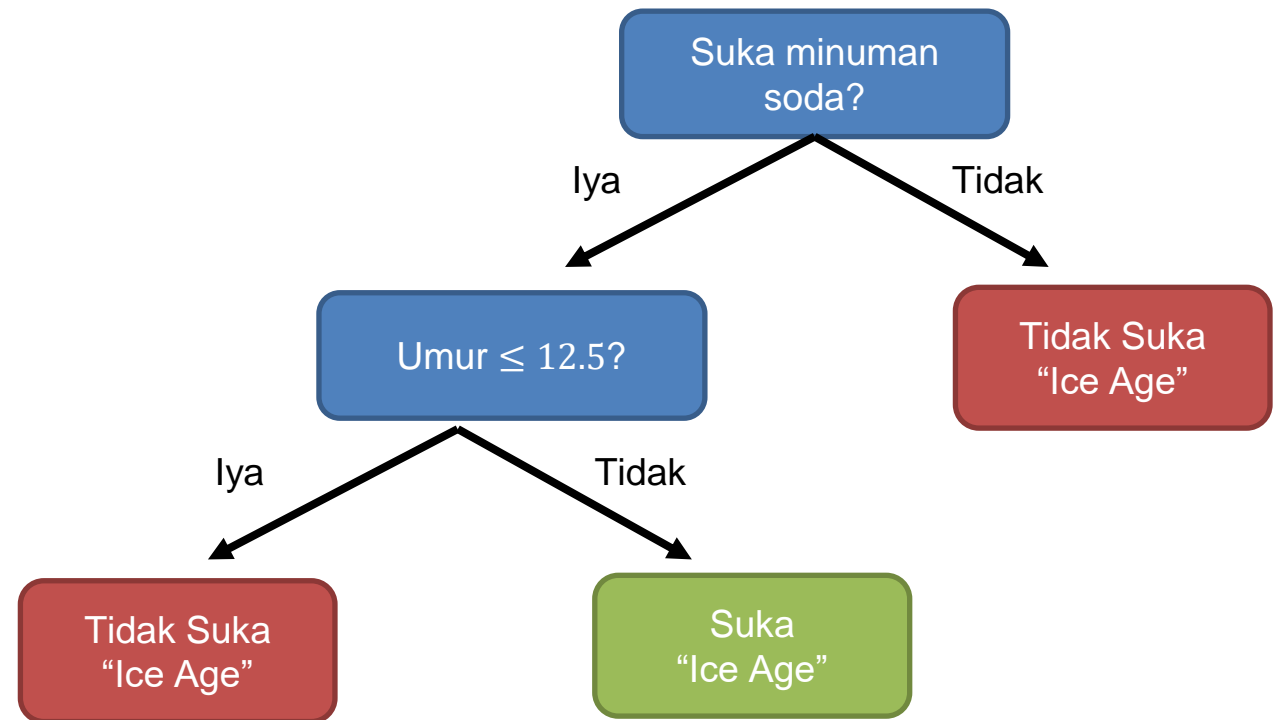
C. Metode Embedded

- Kombinasi metode filter dan wrapper. Cepat seperti metode filter dan hasilnya akurat seperti metode wrapper. Seleksi **fitur merupakan bagian dari algoritma *machine learning*** itu sendiri.
- “Akurat” dalam pengertian: pengaruh seleksi fitur sudah benar-benar dihitung berdasarkan hasil klasifikasi/regresi/clustering (berbeda halnya dengan metode filter).
- Contoh: Decision Tree, Regresi Lasso

C. Metode Embedded: Decision Tree

- Contoh pada slide kuliah sebelumnya, fitur “Suka Popcorn” tidak digunakan di decision tree. Mengapa?
- Apakah nilai Gini Index secara tidak langsung “menyeleksi” fitur?

| No. | Suka popcorn | Suka Minuman Soda | Umur | Suka “Ice Age” |
|-----|--------------|-------------------|------|----------------|
| 1 | Iya | Iya | 7 | Tidak |
| 2 | Iya | Tidak | 12 | Tidak |
| 3 | Tidak | Iya | 18 | Iya |
| 4 | Tidak | Iya | 35 | Iya |
| 5 | Iya | Iya | 38 | Iya |
| 6 | Iya | Tidak | 50 | Tidak |
| 7 | Tidak | Tidak | 83 | Tidak |



C. Metode Embedded: Linear Regression

- Output dari linear regression adalah sebuah fungsi dengan bentuk umum
$$y^* = w_1x_1 + w_2x_2 + w_3x_3 + \cdots w_nx_n + d, \text{ dengan}$$
- $x_1, x_2, x_3, \dots, x_n$ adalah fitur
- $w_1, w_2, w_3, \dots, w_n$ adalah koefisien atau bobot yang terkait dengan fitur
- *Feature importance* bisa langsung diketahui dari koefisien/bobot dari setiap fitur. Semakin besar bobot, semakin penting fitur tersebut.
- Penjelasan lebih lanjut akan dipelajari pada materi tentang Regresi.

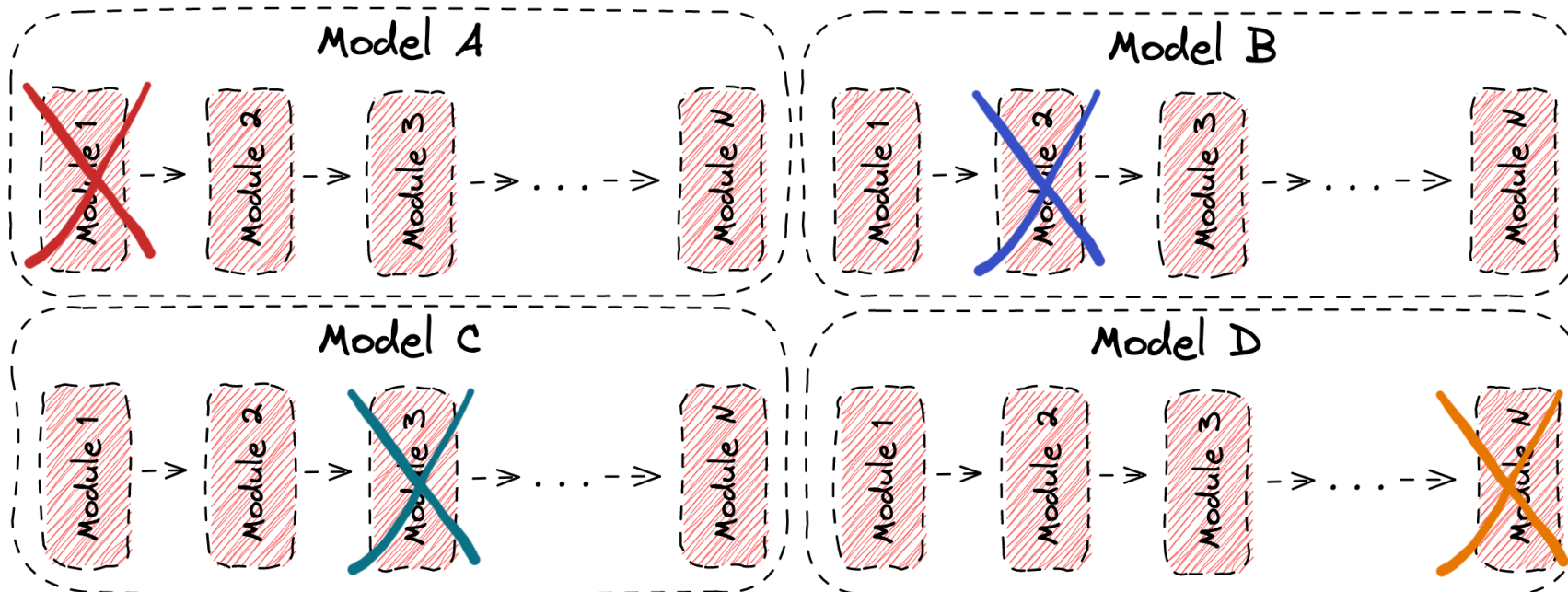


FAKULTAS
**ILMU
KOMPUTER**

More Discussion

Feature Selection vs Ablation Study

- Apa perbedaan antara Feature Selection & Ablation Study pada machine learning?
 - **Feature Selection:** memilih **fitur** (dari data input) yang dapat memberikan kinerja terbaik.
 - **Ablation Study:** mempelajari pengaruh **modul-modul** pada suatu machine learning system terhadap kinerja akhir.



[Machine Learning: What Is Ablation Study? | Baeldung on Computer Science](#)

Bagaimana membandingkan model ML?

- Misalkan kita memiliki 2 atau lebih model ML untuk melakukan klasifikasi terhadap suatu permasalahan. Misalkan: KNN dan CART. Bagaimana menyatakan bahwa satu model lebih baik dari yang lain?
- Apakah cukup hanya dengan melakukan **satu kali** eksperimen?
- Apakah cukup dengan melakukan **k-fold cross validation** lalu mengambil reratanya?
- **Nilai** apa yang perlu dibandingkan? Akurasi, F1-score, sensitivity, specificity, AUC ?

Benchmark Dataset

- Seringkali suatu dataset memiliki sistem **benchmarking**, sehingga dapat membandingkan kinerja suatu model dan model lainnya secara langsung.
- Biasanya model akan diujikan dengan data testing yang sama (yang hanya diketahui oleh pengelola), lalu kinerjanya dievaluasi menggunakan metrik yang sudah ditentukan.
- Jika sudah ada benchmark, maka cukup dibandingkan kinerja model berdasarkan metrik yang sudah ditentukan.
- Model hanya dievaluasi berdasarkan hasil satu kali eksperimen saja.

← ↻ 🏠 🔒 https://motchallenge.net/results/MOT20/ 🔊 ⭐ ⚙️ 📄

Multiple Object Tracking Benchmark

🏠 home ⓘ data ▾ ☰ results ▾ 🖼 vis ★ QVA 🚩 submit ▾ ⓘ FAQ 👤 people 🔑 login ✎ sign up

Benchmark Statistics

| Tracker | ↑MOTA | IDF1 | HOTA | MT | ML | FP | FN | Rcll | Prcn | AssA | DetA | AssRe | AssPr | DetRe | DetPr | LocA | FAF | ID Sw. | Frag | Hz |
|--|-------|------|------|---------------|---------------|--------|---------|------|------|------|------|-------|-------|-------|-------|------|-----|----------------|----------------|------|
| GMOTv2 1. | 77.1 | 74.2 | 61.5 | 895 (72.1) | 117 (9.4) | 43,118 | 73,849 | 85.7 | 91.1 | 59.3 | 63.9 | 66.5 | 72.2 | 71.7 | 76.2 | 83.7 | 9.6 | 1,687 (0.0) | 1,884 (0.0) | 0.9 |
| BOETrackerV2 2. | 74.5 | 73.7 | 60.8 | 792 (63.8) | 163 (13.1) | 18,070 | 112,717 | 78.2 | 95.7 | 60.6 | 61.3 | 65.9 | 78.3 | 65.7 | 80.4 | 84.2 | 4.0 | 1,401 (0.0) | 1,891 (0.0) | 14.9 |
| STC_pub 3. | 73.0 | 67.6 | 56.1 | 833 (67.1) | 147 (11.8) | 30,880 | 106,876 | 79.3 | 93.0 | 52.9 | 59.8 | 59.5 | 72.7 | 65.6 | 76.9 | 83.2 | 6.9 | 2,172 (0.0) | 3,313 (0.0) | 1.4 |
| MrMOT 4. | 67.7 | 67.8 | 53.9 | 689 (55.5) | 162 (13.0) | 32,536 | 131,330 | 74.6 | 92.2 | 52.9 | 55.1 | 58.8 | 71.7 | 60.5 | 74.7 | 81.7 | 7.3 | 3,176 (0.0) | 8,874 (0.0) | 16.9 |
| kalman_pub 5. | 67.0 | 70.2 | 56.4 | 592 (47.7) | 263 (21.2) | 9,685 | 160,303 | 69.0 | 97.4 | 58.3 | 54.8 | 65.0 | 73.6 | 57.7 | 81.4 | 84.1 | 2.2 | 680 (0.0) | 1,738 (0.0) | 17.7 |
| Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang. ByteTrack: Multi-Object Tracking by Associating Every Detection Box . In Proceedings of the European Conference on Computer Vision (ECCV), 2022. | | | | | | | | | | | | | | | | | | | | |
| OUTrack_fm_p 6. | 65.4 | 65.1 | 52.1 | 615 (49.5) | 165 (13.3) | 38,243 | 137,770 | 73.4 | 90.8 | 50.7 | 53.8 | 53.9 | 77.8 | 59.4 | 73.6 | 81.4 | 8.5 | 2,885 (0.0) | 7,205 (0.0) | 5.1 |
| Q. Liu, D. Chen, Q. Chu, L. Yuan, B. Liu, L. Zhang, N. Yu. Online Multi-Object Tracking with Unsupervised Re-Identification Learning and Occlusion Estimation . In Neurocomputing, 2022. | | | | | | | | | | | | | | | | | | | | |

Jika tidak ada benchmark?

- Bagaimana jika kita ingin membandingkan kinerja model yang kita usulkan dengan model peneliti lain yang sudah di publish di jurnal/publikasi lainnya?
 - Pastikan kita menggunakan data training/testing yang sama dengan peneliti lain
 - Jika informasi tersebut tidak ada, lakukan eksperimen ulang menggunakan data yang sama, dengan k-fold cross validation.

J. Zheng et al.

ISPRS Journal of Photogrammetry and Remote Sensing 173 (2021) 95–121

Table 4

The F1-score of other state-of-the-art tree crown detection methods in Site 1.

Growing status observation for oil palm trees using Unmanned Aerial Vehicle (UAV) images

| Method | Healthy palm | Dead palm | Mismanaged palm | Smallish palm | Yellowish palm | Average F1-score |
|--------------------|---------------|---------------|-----------------|---------------|----------------|------------------|
| RF | 79.05% | 0.46% | 0.00% | 37.50% | 12.91% | 25.98% |
| SVM | 77.47% | 0.00% | 0.00% | 33.45% | 5.03% | 23.19% |
| CNN (ResNet-101) | 74.76% | 7.33% | 2.86% | 35.93% | 19.26% | 28.03% |
| Faster R-CNN | 90.46% | 6.47% | 42.48% | 65.76% | 73.54% | 55.74% |
| Grid R-CNN | 90.62% | 13.37% | 41.82% | 66.14% | 69.22% | 56.23% |
| GA Faster R-CNN | 88.60% | 15.67% | 54.70% | 61.52% | 71.18% | 58.33% |
| Cascade R-CNN | 91.22% | 36.36% | 40.00% | 64.48% | 71.46% | 60.71% |
| Libra Faster R-CNN | 91.00% | 30.54% | 55.74% | 65.17% | 69.87% | 62.46% |
| MOPAD (ours) | 91.10% | 55.28% | 51.76% | 77.06% | 88.92% | 72.83% |

Uji Statistik

- Misalkan model kita berhasil mencapai rerata akurasi 76%, sedangkan model lain memperoleh akurasi 74%, apakah kita dapat mengatakan model kita lebih baik secara **signifikan**?
 - Perlu lakukan uji statistik
 - Yang ingin diuji adalah **rerata** akurasi (atau F1-score, sensitivity, dll),
 - Variansi populasi tidak diketahui, hanya variansi sampel (hasil eksperimen) yang diketahui
 - Gunakan **t-test**

t - test

Tahapan t-test adalah sebagai berikut:

1. Buat hipotesa H_0 dan H_1
2. Tentukan level of significance yang ingin digunakan, dapatkan daerah tolak H_0 dan terima H_0
3. Hitung test-score dari data kita
4. Bandingkan apakah test score tersebut berada di daerah tolak atau terima H_0
5. Buat kesimpulan

Contoh

Badu sedang mengembangkan algoritma *sorting* baru. Untuk menguji kinerja algoritma yang dikembangkannya, Badu membandingkannya dengan algoritma *merge-sort*. Badu merasa yakin bahwa algoritma yang dikembangkannya tersebut lebih baik daripada algoritma *merge-sort*.

Badu melakukan beberapa kali percobaan terhadap algoritma yang dikembangkannya dan juga algoritma *merge-sort*. Berikut ini adalah data *response time* yang dicatat Badu untuk kedua algoritma tersebut.

| Eksperimen Ke- | Response Time Algoritma Badu (ms) | Response Time Algoritma Merge-Sort (ms) |
|----------------|-----------------------------------|---|
| 1 | 100 | 150 |
| 2 | 200 | 250 |
| 3 | 100 | 100 |
| 4 | 250 | 200 |
| 5 | 450 | 500 |
| 6 | 350 | 275 |
| 7 | 450 | 450 |
| Rerata | 271,43 | 275,00 |

Dengan memperhitungkan data tersebut dan asumsi bahwa variansi kedua populasi sama, tentukan apakah secara statistik algoritma baru yang dibuat Badu memiliki kinerja yang sama dengan algoritma *merge-sort*, atau malah lebih baik dari algoritma *merge-sort* tersebut? Gunakan *level of significance* 0.1.

Contoh

Tahap 1: Buat hipotesa H0 dan H1

H0: mean response time algoritma Badu **sama dengan** algoritma merge sort. ($\mu_{Badu} = \mu_{MS}$ atau $\mu_{Badu} - \mu_{MS} = 0$)

H1: mean response time algoritma Badu **lebih cepat** dari algoritma merge sort. ($\mu_{Badu} < \mu_{MS}$ atau $\mu_{Badu} - \mu_{MS} < 0$)

Alternatif H0 karena yang dipakai adalah one-tailed:

H0: mean response time algoritma Badu **tidak lebih cepat** dari algoritma merge sort. ($\mu_{Badu} \geq \mu_{MS}$ atau $\mu_{Badu} - \mu_{MS} \geq 0$)

Contoh

Tahap 2: Tentukan *level of significance* yang ingin digunakan, dapatkan daerah tolak/terima H_0

Dari soal, diminta *level of significance* adalah 0.1.

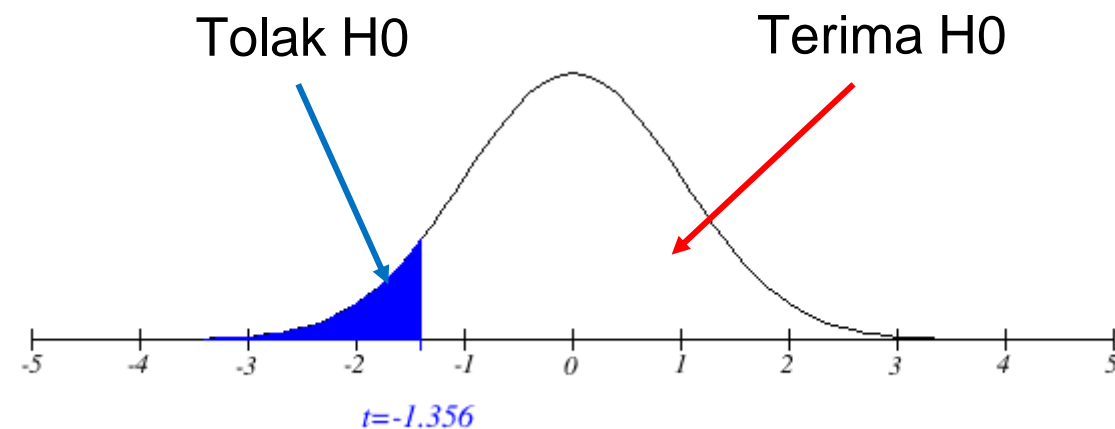
Jumlah data adalah 7, sehingga *degree of freedom* (df) adalah $n+m-2 = 7+7-2=12$.

m adalah jumlah data pada dataet 1 dan n adalah jumlah data pada dataset 2.

Dalam hal ini kebetulan sama yaitu 7.

t-test table

| cum. prob | $t_{.50}$ | $t_{.75}$ | $t_{.80}$ | $t_{.85}$ | $t_{.90}$ | $t_{.95}$ | $t_{.975}$ |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| one-tail | 0.50 | 0.25 | 0.20 | 0.15 | 0.10 | 0.05 | 0.025 |
| two-tails | 1.00 | 0.50 | 0.40 | 0.30 | 0.20 | 0.10 | 0.05 |
| df | | | | | | | |
| 1 | 0.000 | 1.000 | 1.376 | 1.963 | 3.078 | 6.314 | 12.71 |
| 2 | 0.000 | 0.816 | 1.061 | 1.386 | 1.886 | 2.920 | 4.303 |
| 3 | 0.000 | 0.765 | 0.978 | 1.250 | 1.638 | 2.353 | 3.182 |
| 4 | 0.000 | 0.741 | 0.941 | 1.190 | 1.533 | 2.132 | 2.776 |
| 5 | 0.000 | 0.727 | 0.920 | 1.156 | 1.476 | 2.015 | 2.571 |
| 6 | 0.000 | 0.718 | 0.906 | 1.134 | 1.440 | 1.943 | 2.447 |
| 7 | 0.000 | 0.711 | 0.896 | 1.119 | 1.415 | 1.895 | 2.365 |
| 8 | 0.000 | 0.706 | 0.889 | 1.108 | 1.397 | 1.860 | 2.306 |
| 9 | 0.000 | 0.703 | 0.883 | 1.100 | 1.383 | 1.833 | 2.262 |
| 10 | 0.000 | 0.700 | 0.879 | 1.093 | 1.372 | 1.812 | 2.228 |
| 11 | 0.000 | 0.697 | 0.876 | 1.088 | 1.363 | 1.796 | 2.201 |
| 12 | 0.000 | 0.695 | 0.873 | 1.083 | 1.356 | 1.782 | 2.179 |
| 13 | 0.000 | 0.694 | 0.870 | 1.079 | 1.350 | 1.771 | 2.160 |



Contoh

Tahap 3: Hitung test-score (TS) dari data kita

| Set Data Ke- | Response Time Algoritma Badu (ms) | Response Time Algoritma Merge-Sort (ms) |
|--------------|-----------------------------------|---|
| 1 | 100 | 150 |
| 2 | 200 | 250 |
| 3 | 100 | 100 |
| 4 | 250 | 200 |
| 5 | 450 | 500 |
| 6 | 350 | 275 |
| 7 | 450 | 450 |
| Rerata | 271,43 | 275,00 |
| Std Deviasi | 149,60 | 149,30 |
| Variansi | 22380,95 | 22291,67 |

$$TS = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{S_p^2 \left(\frac{1}{n} + \frac{1}{m} \right)}} \quad S_p^2 = \frac{(n-1)S_1^2 + (m-1)S_2^2}{n+m-2}$$

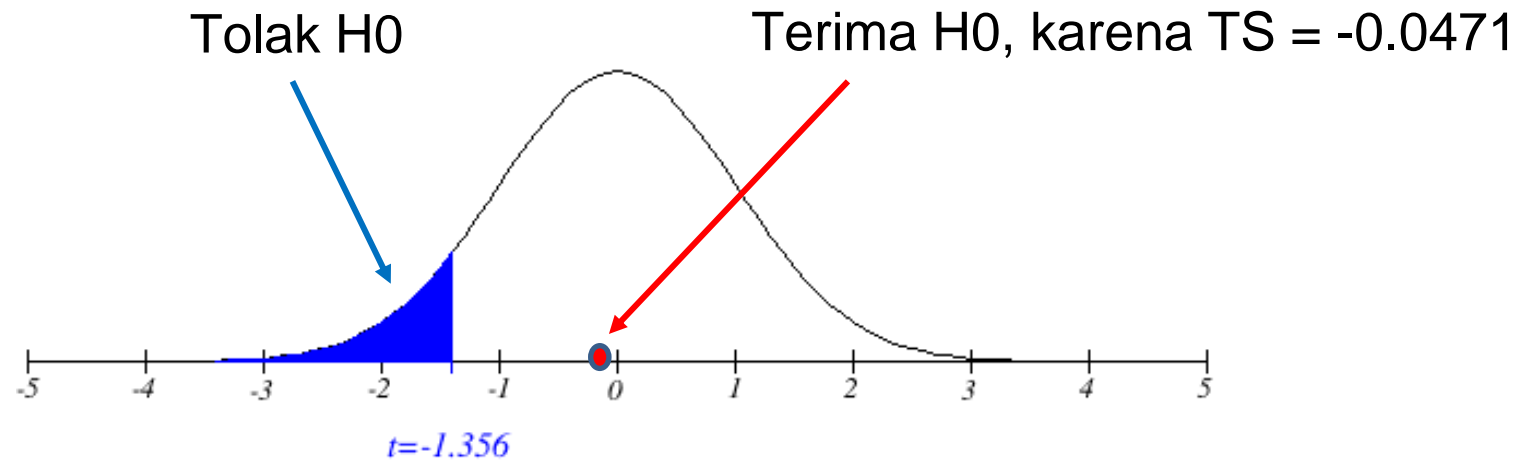
$$S_p^2 = \frac{6(22380,95) + 6(22291,67)}{7 + 7 - 2} = 22336,31$$

Pada eksperimen kita, variansi populasi tidak diketahui, tapi diasumsikan sama (anggap saja sama-sama σ^2)
 S_p^2 disebut dengan **pooled estimator** untuk σ^2 .

$$TS = \frac{271,43 - 275}{\sqrt{22336,31 \left(\frac{1}{7} + \frac{1}{7} \right)}} = \frac{-3,57}{79,89} = -0,00471$$

Contoh

Tahap 4: Bandingkan apakah test score tersebut berada di daerah tolak atau terima H_0



Tahap 5: Buat kesimpulan

Karena $TS > -1.356$, maka **tidak terdapat cukup bukti untuk mendukung klaim bahwa algoritma Badu lebih cepat daripada algoritma MS pada *level of significance* 0.1**



FAKULTAS
ILMU
KOMPUTER

TERIMA KASIH