DOLab

live class

Supervised Learning – Regression (I)

Outline

- Introduction to Supervised Learning
- Regression Algorithms
- Linear Regression
- Modelling
- Model Evaluation

Introduction to Supervised Learning

Types of Machine Learning

Berdasarkan keberadaan “target variabel”

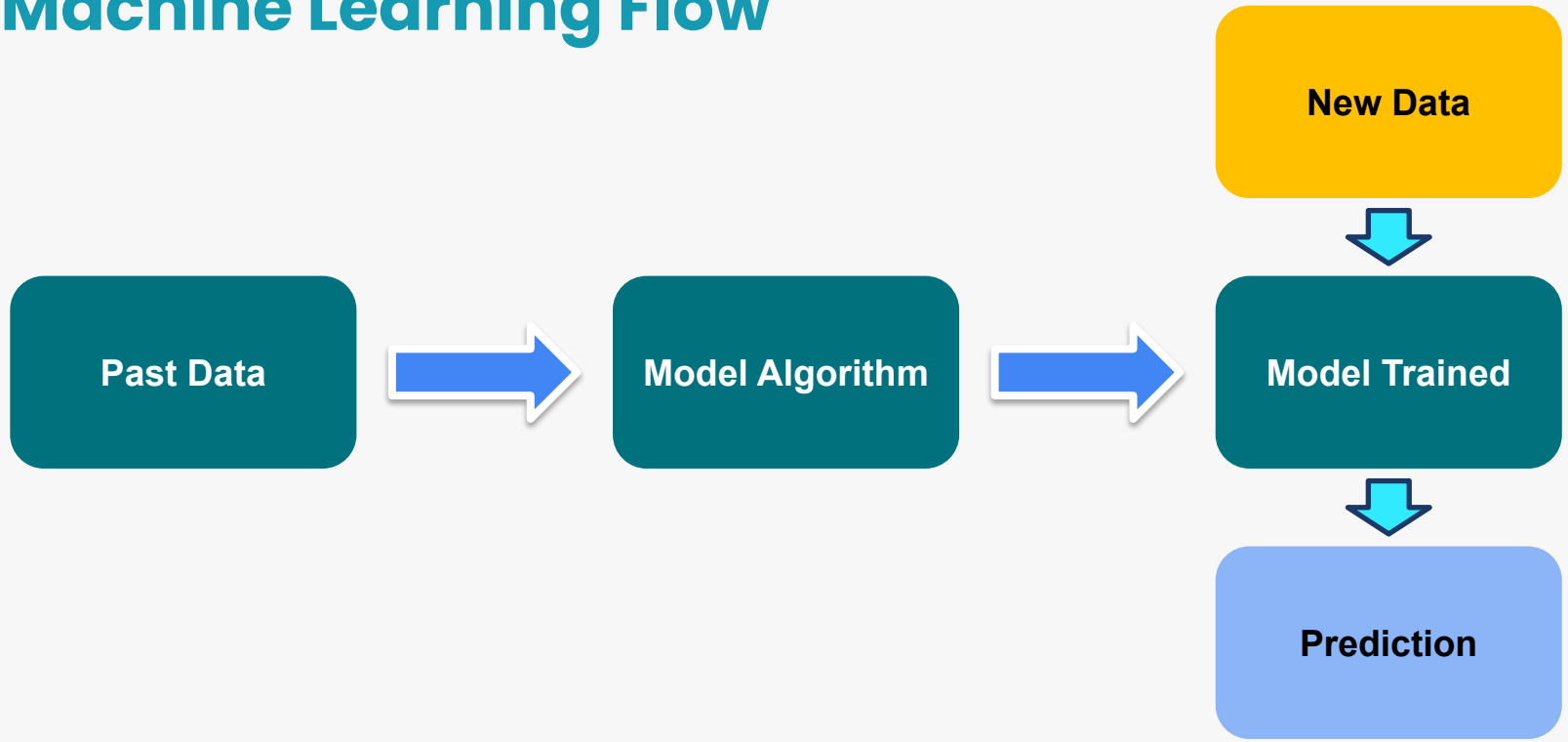
1. Supervised learning

- Supervised = variabel target prediksi (y) diberikan di past data
  - (x1,y1), (x2,y2), (x3,y3), etc..
- Objective: memprediksi y seakurat mungkin dari data x yang baru
- Berdasarkan jenis target variabel, dapat dibedakan jadi 2
  - Regresi
  - Klasifikasi

2. Unsupervised learning

- Unsupervised = tidak ada target variabel di past data
  - (x1), (x2), (x3), ...
- Objective: mencari pattern yang tersembunyi dari data
  - Clustering
  - Dimensionality reduction

Machine Learning Flow



Supervised Learning

Logika

1. Gunakan data yang ada untuk mempelajari pola hubungan antara feature dan target
2. Model yang sudah ditrain digunakan untuk prediksi data baru

features → | | *target*

Past Data

gre_score	toefl_score	univ_ranking	motiv_letter_strength	recommendation_strength	gpa	research_exp	admit_prob
337	118	4	4.5	4.5	9.65	1	0.92
324	107	4	4.0	4.5	8.87	1	0.76
316	104	3	3.0	3.5	8.00	1	0.72
322	110	3	3.5	2.5	8.67	1	0.80

and many more rows…

New Data

314	103	2	2.0	3.0	8.21	0	?

Regression Algorithms

Regression

Regresi adalah metode statistik yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen (variabel prediktor) dengan variabel dependen (variabel respons) yang bersifat kontinu.

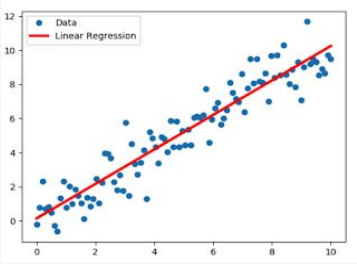
Tujuan utama regresi adalah untuk memahami dan menggambarkan hubungan antara variabel input (variabel independen) dengan variabel output (variabel dependen) serta memprediksi nilai output berdasarkan variabel input.

Beberapa jenis regresi :

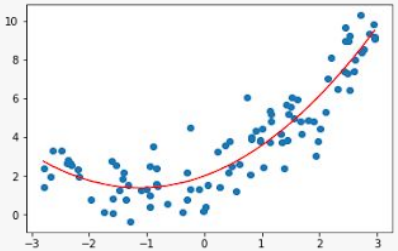
- Simple and Multiple Linear Regression
- Logistic Regression
- Polynomial Regression
- Ridge and Lasso Regression

Setiap jenis regresi memiliki karakteristik dan asumsi sendiri. Pilihan jenis regresi yang tepat tergantung pada karakteristik data, bentuk hubungan antara variabel, dan tujuan analisis yang ingin dicapai.

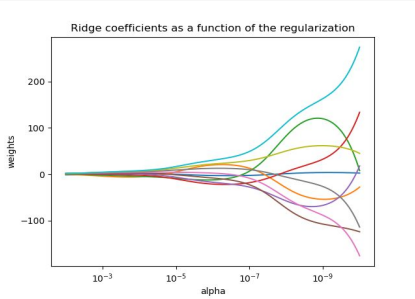
Jenis Regression



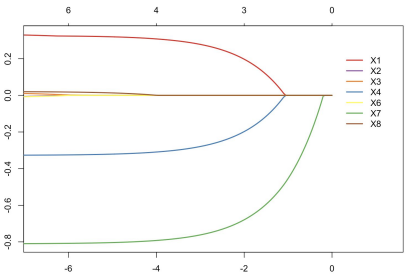
Linear Regression



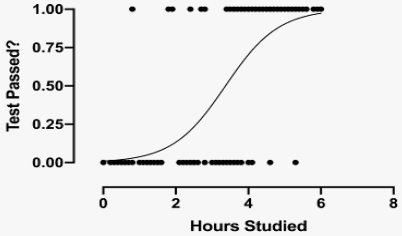
Polynomial Regression



Ridge Regression



Lasso Regression



Logistic Regression

https://scikit-learn.org/stable/modules/linear_model.html

Linear Regression

Linear Regression

- Linear regression merupakan model statistik paling tua
- Linear regression tujuannya adalah membuat garis lurus yang paling cocok terhadap data yang ada

- Model matematika

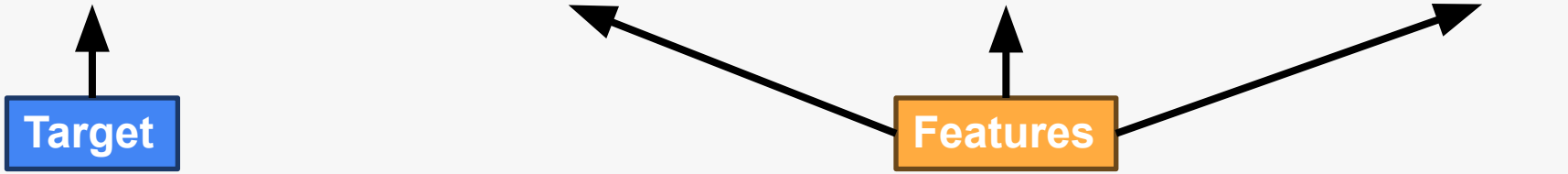
Simple Linear Regression → $y = b_0 + b_1 x$

Multiple Linear Regression → $y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$

Karakteristik supervised model untuk regresi :
- y atau target disediakan
- y atau target bersifat kontinu

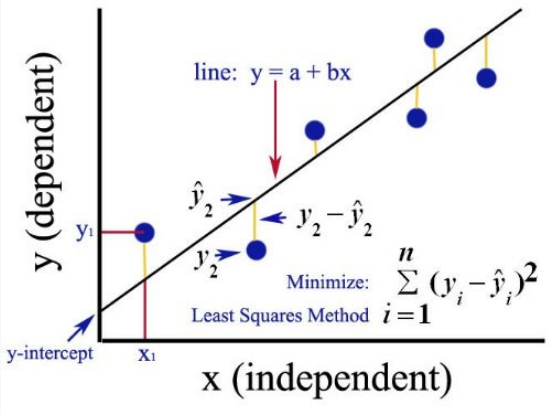
Regression Model

Predict price house based on features



- Featurenya adalah Area, No. Rooms, and No. Floors
- Target adalah Price

Simple Linear Regression



Selisih antara data actual (y_i) dengan hasil model $(\hat{y}_i)$ kemudian akan disebut error.

Error yang dihasilkan tentu bisa melebihi hasil model (data actual > hasil model) maka error akan bernilai positif (+) dan bisa juga kurang (data actual < hasil model) maka error akan bernilai (-) sehingga Ketika dirata-rata akan saling mengurangi maka salah satu caranya adalah dengan meng**kuadrat**kan selisih

Hasil model yang baik tentu error yang dihasilkan harus se**minimal** mungkin maka ide dari model ini jika di konstruksikan akan menjadi

$$\min \left\{ \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right\}$$

Metode ini disebut dengan Ordinary Least Square atau OLS

Mengenal Persamaan Garis

Bentuk Umum

Bentuk umum persamaan garis lurus (kurva lurus) adalah

$$Y = \beta_0 + \beta_1 X_1$$

Intercept
Akan mempengaruhi nilai awal garis (0)

slope
Akan mempengaruhi kemiringan garis

Misal

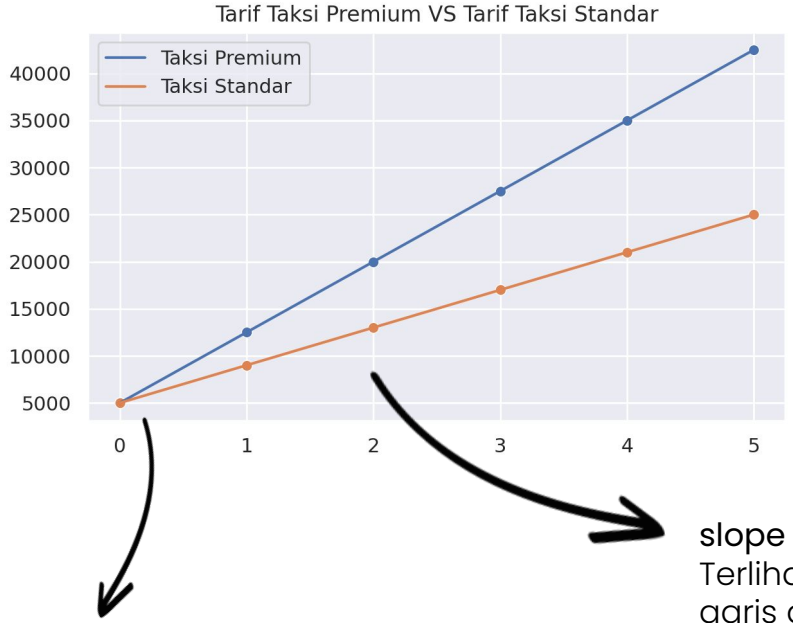
Perusahaan taksi mempunyai 2 jenis taksi yakni taksi premium dan taksi standar. Pada pemesanan awal kedua jenis taksi memberlakukan tarif yang sama yakni Rp. 5000. Namun tarif argo berdasarkan jarak antar taksi berbeda. Untuk taksi premium dikenakan Rp. 7500 / kilometer jarak ditempuh sedangkan taksi standar Rp. 4000 / kilometer jarak ditempuh.

Taksi Premium

Jarak	Tarif awal	Tarif Jarak	Total
1 km	5000	7500	12500
2 km	5000	15000	20000
3 km	5000	22500	27500

Taksi Standar

Jarak	Tarif awal	Tarif Jarak	Total
1 km	5000	4000	9000
2 km	5000	8000	13000
3 km	5000	12000	17000



$$Tarif\ Taksi = Tarif\ Awal + Tarif\ Per\ Kilometer * Jarak\ ditempuh$$

Variabel terikat

$$Premium = 5000 + 7500 * x$$
$$Standar = 5000 + 4000 * x$$

Variabel bebas

Intercept

slope

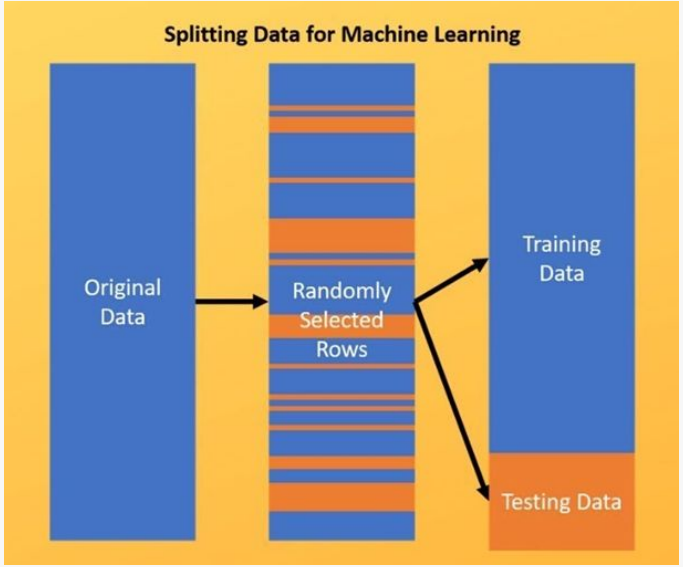
slope
Terlihat perbedaan kemiringan antara dua garis dimana semakin tinggi nilai slope semakin curam garis yang terbentuk

intercept
Tarif awal = 5000 maka jika jarak yang ditempuh misalkan 0 km maka penumpang tetap akan dikenakan tarif 5000

Modelling

Aturan dalam Modelling

- Semua data tidak bisa digunakan untuk train model.
- Harus dibagi beberapa porsi untuk test data



- Tujuan dilakukan split data menjadi train dan test adalah agar model kita bekerja baik terhadap data yang belum pernah dilihat.
- Test data ditujukan untuk meniru data yang tidak terlihat

Modelling using Linear Regression

Steps :

- Pisahkan feature dan target
  feature dan target
- Split data : train – test
  feature_train, feature_test, target_train, target_test
- Tentukan model
  linreg = LinearRegression()
- Train model
  linreg.fit(feature_train, target_train)
- Predict test data
  linreg.predict(feature_test)
- Evaluasi model
  Gunakan beberapa metrics untuk regresi (RMSE, R2 Score)

Modelling using Linear Regression

Regression menggunakan 1 feature yang digunakan untuk prediksi

	Temperature	Curah_Hujan
0	35	85
1	32	70
2	33	75
3	35	80
4	36	85

Temperature akan dijadikan feature yang digunakan untuk prediksi Curah_Hujan

Modelling using Linear Regression

Data Split

```
from sklearn.model_selection import train_test_split

feature = data.drop(columns='Curah_Hujan')
target = data[['Curah_Hujan']]

feature_train, feature_test, target_train, target_test = train_test_split(feature,
                                                                          target,
                                                                          test_size=0.20,
                                                                          random_state=42)
```

- Pemisahan feature dan target
- Train test split dengan ratio 80 : 20
- Random state digunakan untuk reproducibility

Modelling using Linear Regression

Model Training and Coefficient

```
from sklearn.linear_model import LinearRegression

# Tentukan model yang akan digunakan
linreg = LinearRegression()

# Train Model
linreg.fit(feature_train, feature_train)
```

	feature	coefficient
0	intercept	2.3842
1	Temperature	1.2835

Curah_Hujan = 2,3842 + 1,2835 Temperature

- Ketika Temperature 0, Curah_Hujannya adalah 2,3842
- Setiap kenaikan 1 Temperature, mengakibatkan kenaikan 1,2835 Curah Hujan

Multiple Linear Regression

Steps :

- Pisahkan feature dan target
  feature dan target
- Split data : train – test
  feature_train, feature_test, target_train, target_test
- Multicollinearity Check
  Calculate VIF Score dan Analisis Korelasi
- Tentukan model
  linreg = LinearRegression()
- Train model
  linreg.fit(feature_train, target_train)
- Predict test data
  lin_reg.predict(feature_test)
- Evaluasi model
  Gunakan beberapa metrics untuk regresi (RMSE, R2 Score)

Multicollinearity

- **Multicollinearity : kondisi dimana dua atau lebih variable memiliki korelasi tinggi antara satu sama lain.**
- **Multikolinieritas dapat mempengaruhi interpretasi dan estimasi koefisien regresi serta mempengaruhi kinerja model.**
- **Deteksi bisa dilakukan menggunakan Variance Inflation Factor (VIF)**

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

- VIF = 1 □ No multicollinearity
- VIF antara 4 – 10 □ Moderate multicollinearity
- VIF di atas 10 □ Multicollinearity parah

Multicollinearity

Variance Inflation Factor (VIF)

```
from statsmodels.stats.outliers_influence import variance_inflation_factor as vif
from statsmodels.tools.tools import add_constant

X_vif = add_constant(X_train)

vif_df = pd.DataFrame([vif(X_vif.values, i)
               for i in range(X_vif.shape[1])],
              index=X_vif.columns).reset_index()
vif_df.columns = ['feature','vif_score']
vif_df = vif_df.loc[vif_df.feature!='const']
vif_df
```

	feature	vif_score
1	CRIM	1.682416
2	ZN	2.273766
3	INDUS	4.241354
4	CHAS	1.089018
5	NOX	4.452894
6	RM	2.085863
7	AGE	3.163989
8	DIS	4.033996
9	RAD	7.100781
10	TAX	9.157594
11	PTRATIO	1.814771

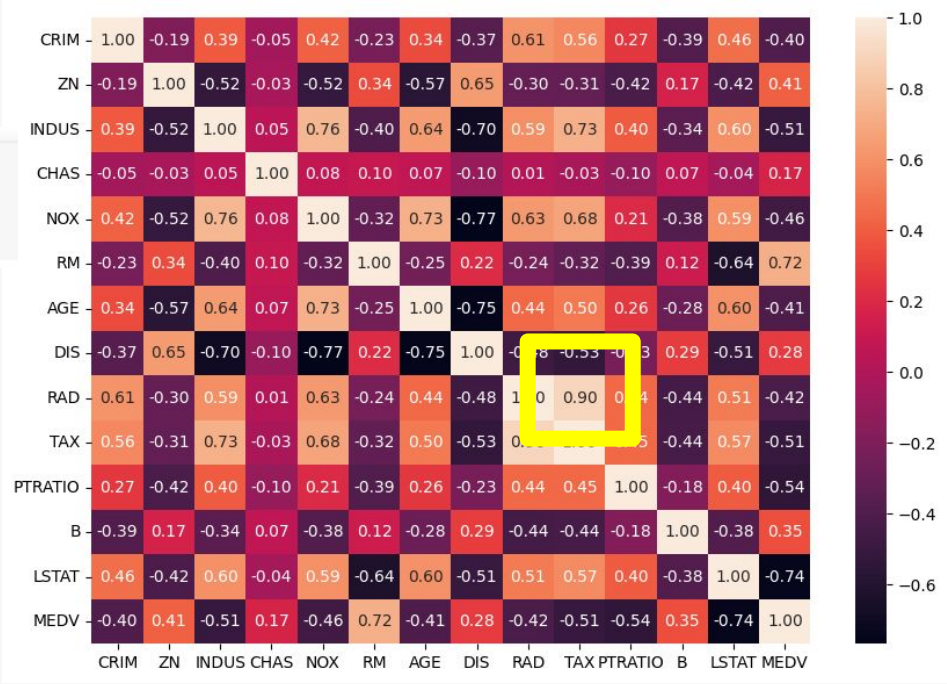
Multicollinearity

Feature Correlation

Gambar korelasi heatmap

```
plt.figure(figsize=(10,7))
sns.heatmap(df.corr(), annot=True, fmt='.2f')
```

- TAX dan RAD mempunyai korelasi yang tinggi
- Drop RAD karena TAX berkorelasi lebih tinggi terhadap MEDV yang merupakan target.



Model Evaluation

R-squared Score

Metriks evaluasi yang digunakan untuk mengukur sejauh mana variable dependent dapat dijelaskan oleh model regresi.

- R-squared = 0 berarti model tidak dapat menjelaskan variasi apapun dalam data
- R-squared = 1 berarti model mampu menjelaskan seluruh variasi dalam data

Range R-squared berada di antara 0 dan 1 dimana semakin besar semakin bagus.

```
from sklearn.metrics import r2_score

y_predict_train = multi_reg.predict(X_train)
r2_score(y_predict_train, y_train)
```

```
0.734629719912701
```

Interpretation :
73,46% sukses dijelaskan menggunakan fitur yang ada pada model

Root Mean Squared Error

Model performance dilakukan pada test data

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

```
from sklearn.metrics import mean_squared_error

y_predict_test = multi_reg.predict(X_test)
np.sqrt(mean_squared_error(y_predict_test, y_test))
```

```
5.7775677975912645
```

Interpretation :

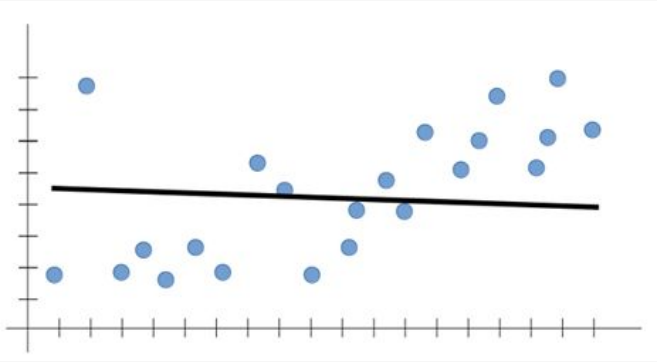
- Standar deviasi error dari prediksi sebesar 5,77
- Dari garis regresi, deviasi errornya antara +- 5,77

Underfitting dan Overfitting

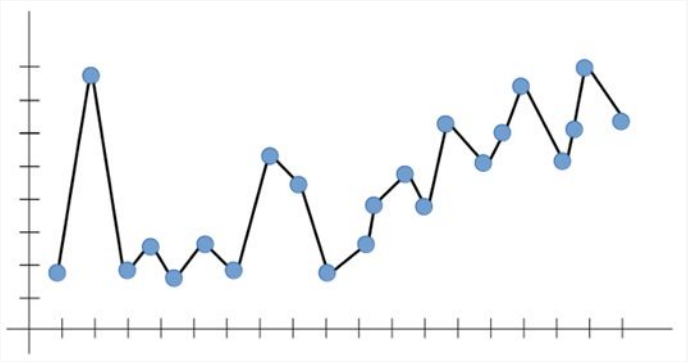
Underfitting : Ketidakmampuan model untuk mempelajari hubungan antar variable dalam data serta tidak mampu untuk memprediksi atau mengklasifikasikan data point yang baru.
Efek : Model terlalu sederhana dan tidak menangkap *trend* dari dataset yang berarti tidak mempelajari pola dari data.

Overfitting : Keadaan dimana model berusaha untuk mempelajari seluruh detail termasuk noise yang ada dalam data dan berusaha untuk mengikutsertakan semua data point ke dalam garis.
Efek : Akurasi tinggi pada data train, namun gagal memprediksi pada data baru

Underfitting dan Overfitting



Underfitting



Overfitting

Terima Kasih