Nama : Muchamad Aqiel Hilman Maulandany (1301170428)
Kelas : IF 41 10

# TUGAS BESAR MATA KULIAH PEMBELAJARAN MESIN

Studi kasus FIFA 2020

## Problem

Dalam tugas berikut diberikan suatu dataset FIFA 2020, yang berupa data-data statistik pemain bola. Permasalahan yang akan diambil yaitu menentukan posisi pemain dalam team berdasarkan kemampuannya. Hal tersebut dapat berguna untuk merekomendasikan dan memposisikan pemain baru dalam tim. Sehingga dapat mengurangi kesalahan dalam mengkomposisikan tim. Hal ini sangat sulit untuk dilakukan secara manual, dikarenakan banyaknya posisi yang dapat diisi. Maka dari itu diperlukan pendekatan dengan Machine Learning, sehingga dapat menentukan posisi paling optimal untuk pemain. Dalam melakukan pendekatan dengan Machine Learning dapat melalui beberapa tahap yaitu :

## Data Preparation

Dataset yang digunakan merupakan data pemain game fifa 2020. Data yang diberikan terdiri dari 104 kolom, dengan 18.278 baris. Hal yang pertama kali diperlukan yaitu memotong jumlah kolom dataset. Tahapan ini akan mengambil kolom" yang berelasi dengan kolom yang akan dicari yaitu *team_position*. Kolom yang akan diambil ada 49 kolom, diantaranya :

```
team_position,overall,potential,pace,shooting,passing,dribbling,defending,physic,gk_diving,gk_handling,gk_kicking,gk_reflexes,gk_speed,gk_positioning,attacking_crossing,attacking_finishing,attacking_heading_accuracy,attacking_short_passing,attacking_volleys,skill_dribbling,skill_curve,skill_fk_accuracy,skill_long_passing,skill_ball_control,movement_acceleration,movement_sprint_speed,movement_agility,movement_reactions,movement_balance,power_shot_power,power_jumping,power_stamina,power_strength,power_long_shots,mentality_aggression,mentality_interceptions,mentality_positioning,mentality_vision,mentality_penalties,mentality_composure,defending_marking,defending_standing_tackle,defending_sliding_tackle,goalkeeping_diving,goalkeeping_handling,goalkeeping_kicking,goalkeeping_positioning, goalkeeping_reflexes
```

Dikarenakan ke 49 kolom tersebut berkaitan dengan penentuan posisi pemain dalam tim. Posisi pemain dalam tim merupakan data categorical yang terdiri dari 29 kategori, yaitu :

```
'RW' 'LW' 'CAM' 'GK' 'RCM' 'LCB' 'ST' 'CDM'
'LDM' 'RM' 'RCB' 'LCM' 'LM'
 'CF' 'SUB' 'LB' 'LS' 'RB' 'RDM' 'RES' 'RAM'
'RS' 'RF' 'CM' 'CB' 'LF'
 'LAM' 'RWB' 'LWB'
```

## Feature Engineering

Setelah dataset di pangkas, maka akan dilakukan pembersihan di tahap ini. Terdapat 9 teknik dalam melakukan feature engineering. Namun tidak perlu menggunakan semuanya.

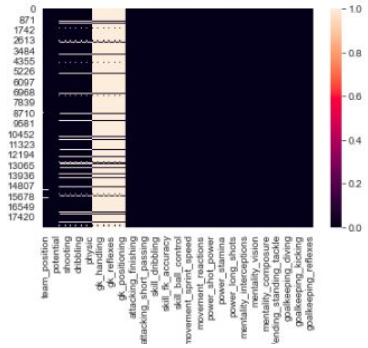
Nama : Muchamad Aqiel Hilman Maulandany (1301170428)
Kelas : IF 41 10

Saya menggunakan 3 teknik dalam melakukan pembersihan data yaitu :

1. **Imputation**

   Dataset yang diberikan masih banyak data yang tidak valid atau bernilai null. Maka dari itu diperlukan pemberian nilai untuk setiap data yang kosong dengan imputation. Dataset yang telah diringkas terdiri dari 48 kolom numerical dan 1 kolom categorical. Maka perlu dipisah terlebih dahulu.

   Untuk kolom numerical dilakukan imputation dengan mengganti NaN atau null dengan value 0. Dikarenakan persebaran datanya antara grup kolom kiper dan selain kiper memiliki kebalikan.



   Dimana jika ada null pada kolom group goalkeeper ability, kolom group kolom ability lainnya terisi, dan begitupun sebaliknya. Maka diasumsikan ability yang null, bernilai default 0.

   Pada kolom categorical, baris data yang bernilai NaN atau Null di hapus. Dikarenakan jumlahnya tidak banyak dan diasumsikan merupakan kesalahan input.

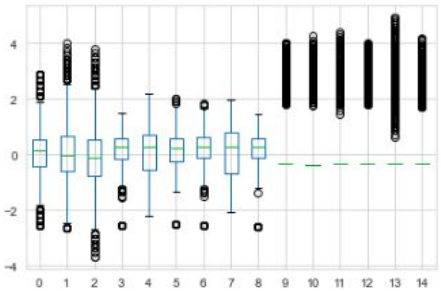
2. **Binning / Mapping**

   Setelah semua data tidak terdapat null, maka proses yang dilakukan selanjutnya yaitu mengubah kolom kategori menjadi angka. Dengan rules sebagai berikut :

```
# Label mapping
position_codes = {
    'RW' : 0,    'LW' : 1,    'CAM' : 2,    'GK' : 3,    'RCM' : 4,
    'LCB' : 5,    'ST' : 6,    'CDM' : 7,    'LDM' : 8,    'RM' : 9,
    'RCB' : 10,    'LCM' : 11,    'LM' : 12,    'CF' : 13,    'SUB' : 14,
    'LB' : 15,    'LS' : 16,    'RB' : 17,    'RDM' : 18,    'RES' : 19,
    'RAM' : 20,    'RS' : 21,    'RF' : 22,    'CM' : 23,    'CB' : 24,
    'LF' : 25,    'LAM' : 26,    'RWB' : 27,    'LWB' : 28
}
```

3. **Scaling**

   Setelah itu dilakukan proses Scaling dengan menggunakan **Standardization**. Proses ini diperlukan untuk menyamakan range data yang berbeda. Selain itu **Standardization** juga dapat mengurangi outlier dari data, menjadi seperti ini :



4. **Split Feature**

   Tahapan ini membagi dataset menjadi 2 feature yang masing-masingnya terdiri dari data train dan data test. Feature 1 terdiri dari ability umum, dan Feature 2 terdiri dari ability khusus/detail.

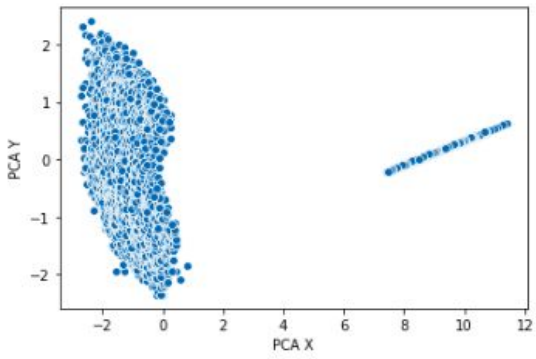
Nama : Muchamad Aqiel Hilman Maulandany (1301170428)
Kelas : IF 41 10

## Clustering

Proses clustering menggunakan dataset feature 1. Dalam melakukan clustering terdapat beberapa tahapan diantaranya :

### 1. Reduksi Dimensi

Kolom yang dimiliki feature 1, ada 15 kolom. Sehingga tidak dapat direpresentasikan ke dalam 2 dimensi (sumbu X dan Y). Maka dari itu perlu di reduksi dengan metode **PCA (Principal Component Analysis).** Sehingga persebaran data dapat direpresentasikan menjadi seperti berikut ini :

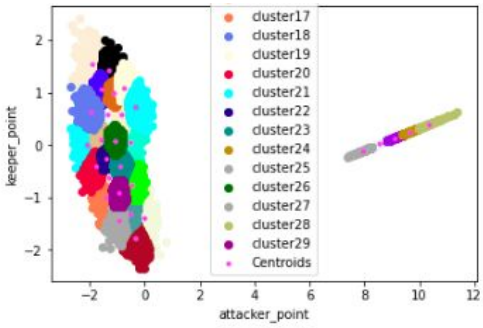



### 2. Proses Clustering menggunakan K-Means

Selanjutnya dilakukan proses clustering dengan menggunakan K-Means. **K-Means** dipilih dikarenakan mudah untuk di implementasi dan umum digunakan. Tahapan dalam algoritma K-Means yaitu :

a. Pilih K buah titik centroid secara acak.
b. Kelompokkan data sehingga terbentuk K buah cluster dengan titik centroid dari setiap cluster merupakan titik centroid yang telah dipilih sebelumnya
c. Perbaharui nilai titik centroid
d. Ulangi langkah b dan c sampai nilai dari titik centroid tidak lagi berubah

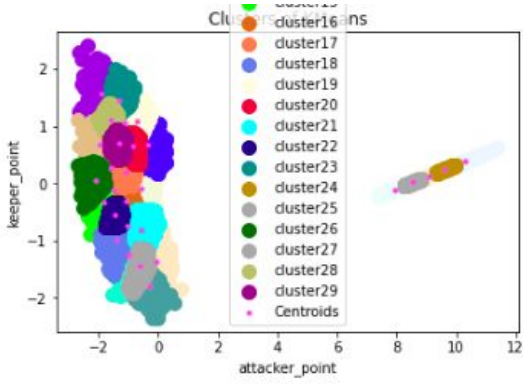
Pada algoritma yang saya buat parameter yang perlu di tuning yaitu nilai K dan Jumlah Literasi. Saya menggunakan nilai K = 29, dikarenakan terdapat 29 kategori posisi team. Lalu literasi yang dilakukan sebanyak 150 kali literasi. Sehingga menghasilkan persebaran data sebagai berikut :




### 3. Eksperimen dengan library K-Means

Untuk membandingkan hasil algoritma yang saya buat, maka saya bandingkan dengan algoritma bawaan dari **SkLearn.**

Dengan menggunakan parameter yang sama, menghasilkan persebaran sebagai berikut :



Nama : Muchamad Aqiel Hilman Maulandany (1301170428)
Kelas : IF 41 10

**4. Evaluasi**

Berdasarkan kedua jenis perbandingan diatas, menunjukkan perbedaan dalam pembagian cluster dan waktu eksekusi. Waktu eksekusi menggunakan algoritma yang saya buat lebih lama dibandingkan dengan library Sklearn, Hasil cluster memiliki beberapa kemiripan, seperti hasil persebaran centroid yang sama. Kanan 5 centroid dan kiri 24 centroid. Namun range dari tiap cluster memiliki perbedaan.

**5. Kesimpulan**

Dari hasil cluster tersebut maka dapat digunakan untuk penentuan posisi team. Sebagai contoh, jika terdapat data baru yang berada pada wilayah centroid 1, maka posisi team tersebut goalkeeper, dan sebagainya. Perkembangan yang dapat dilakukan selanjutnya yaitu menggunakan ELBOW untuk menentukan jumlah K yang sesuai.

## Classification

Proses yang dapat dilakukan selain clustering yaitu classification. Dalam melakukan klasifikasi saya menggunakan 2 jenis algoritma yaitu SVM dan KNN, serta setiap algoritma menggunakan 2 jenis fitur.

**SVM Classification**

SVM (Support Vector Machine) klasifikasi merupakan salah satu klasifikasi yang paling banyak digunakan. Selain karena hasil akurasi yang tinggi, jenis data yang ada pada dataset juga memungkinkan untuk menggunakan metode ini.

Pada eksperimen ini dilakukan tuning parameter dengan menggunakan **RandomSearchCV.** Hal tersebut diperlukan untuk mendapatkan parameter yang paling optimal. Rules pada Saya menggunakan rules tuning sebagai berikut :

| | |
|---|---|
| C | [.0001, .001, .01] |
| kernel | ['linear', 'rbf', 'poly'] |
| gamma | [.0001, .001, .01, .1, 1, 10, 100] |
| degree | [1, 2, 3, 4, 5] |

Sehingga menghasilkan HyperParameter sebagai berikut :

| Feature 1 | Feature 2 |
|---|---|
| Parameter : {'kernel': 'poly', 'gamma': 10, 'degree': 4, 'C': 0.01} | Parameter : {'kernel': 'poly', 'gamma': 10, 'degree': 4, 'C': 0.01} |
| Accuracy : 0.991912293337101 | Accuracy : 0.8748366562380387 |

Setelah itu dibandingkan dengan menggunakan tanpa tuning parameter, sehingga menghasilkan perbandingan sebagai berikut :

| | Model | Training Set Accuracy | Test Set Accuracy |
|---|---|---|---|
| 0 | SVM feature 1 - Tunning | 1.00000 | 0.995935 |
| 1 | SVM feature 1 - Original | 0.95330 | 0.943089 |
| 2 | SVM feature 2 - Tunning | 1.00000 | 0.889874 |
| 3 | SVM feature 2 - Original | 0.90347 | 0.880266 |

Nama : Muchamad Aqiel Hilman Maulandany (1301170428)
Kelas : IF 41 10

**KNN Classification**

Algoritma Lainnya yaitu menggunakan algoritma KNN ( K Nearest Neighbour) . KNN mengklasifikasi berdasarkan tetangga terdekat. Sehingga menurut pendapat saya, setiap data pada dataset saling berdekatan dan algoritma ini dapat digunakan.

Pada eksperimen ini dilakukan tuning parameter dengan menggunakan **GridSearchCV.** Hal tersebut diperlukan untuk mendapatkan parameter yang paling optimal. Rules pada Saya menggunakan rules tuning sebagai berikut :

| | |
|---|---|
| n_neighbors | [5,6,7,8,9,10] |
| leaf_size | [1,2,3,5] |
| weights | ['uniform', 'distance'] |
| algorithm | ['auto', 'ball_tree','kd_tree','brute'] |
| n_jobs | [-1] |

Sehingga menghasilkan HyperParameter sebagai berikut :

| Feature 1 | Feature 2 |
|---|---|
| Parameter : {'algorithm': 'auto', 'leaf_size': 1, 'n_jobs': -1, 'n_neighbors': 5, 'weights': 'distance'} | Parameter : {'algorithm': 'auto', 'leaf_size': 1, 'n_jobs': -1, 'n_neighbors': 10, 'weights': 'distance'} |
| Accuracy : 0.8684453227931489 | Accuracy : 0.6714756258234519 |

Setelah itu dibandingkan dengan menggunakan tanpa tuning parameter, sehingga menghasilkan perbandingan sebagai berikut :

| | Model | Training Set Accuracy | Test Set Accuracy |
|---|---|---|---|
| 0 | KNN feature 1 - Tunning | 1.000000 | 0.875462 |
| 1 | KNN feature 1 - Original | 0.918145 | 0.874723 |
| 2 | KNN feature 2 - Tunning | 1.000000 | 0.678492 |
| 3 | KNN feature 2 - Original | 0.774589 | 0.654472 |

**Evaluasi**

Hasil test Akurasi pada metode SVM hampir mendekati sempurna 99%. Jika tanpa tuning hanya memiliki tingkat akurasi 87%. Sehingga hal tersebut membuktikan jika tuning diperlukan untuk meningkatkan akurasi. Selain itu pengolahan dataset juga menjadi penentu dalam tingginya akurasi. Pada metode SVM menggunakan data yang telah di scaling dengan menggunakan standardization, sehingga dapat mengurangi outlier dan skala jarak antar data. Hal tersebut membuat SVM memiliki tingkat akurasi yang tinggi.

Pada Metode KNN, pada feature 2 terdapat data yang tidak seimbang atau terdapat outlier. Sehingga mengakibatkan tingkat akurasi yang rendah. Maka dari itu diperlukan penyesuaian feature 2, atau diganti dengan algoritma lainnya.

**Kesimpulan**

Berdasarkan kedua eksperimen diatas dapat disimpulkan kalau metode SVM merupakan metode yang terbaik. Feature yang memungkinkan untuk menentukan team_position adalah feature 1 dengan akurasi 99 %.