**LAPORAN TUGAS BESAR**

**MACHINE LEARNING**



**Disusun oleh :**

**Muhammad Ihza Mahendra 1301174682**

**IF-41-GAB**

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS INFORMATIKATELKOM UNIVERSITY**

**2020**

**Report Machine Learning Last Task**

**Muhammad Ihza Mahendra (1301174682)**

**Overview**

In 21st century, the field of Artificial Intelligence has grown rapidly faster every six months, any type of real work task is able to finish using Artificial Intelligence methods, especially in machine learning.

In this section, the main discussion of our problem is about classifying fifa20 using machine learning or as we called supervised learning and also cluster the data that often we called as unsupervised learning in the machine learning field

1. **Problem Formulation**

to be specific, a problem that we have here talked about fifa20 as it has been mention in the overview. What we have here is the dataset that requires with preprocessing, learning and post-processing, but generally there are 4 main steps, which is Manage Data - Training/Learning - Evaluation - Deploy Model. And here we will perform 2 tasks, which is Classification and Clustering.

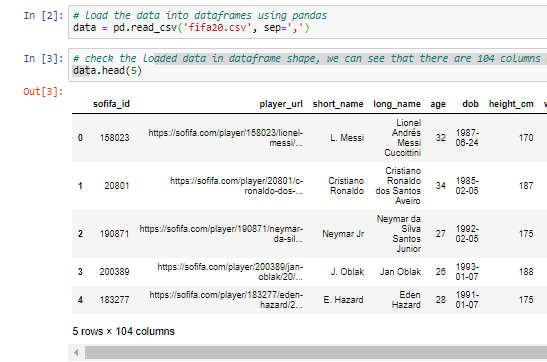
1. **Data Preparation and Exploration**

There are to section for this step in order:

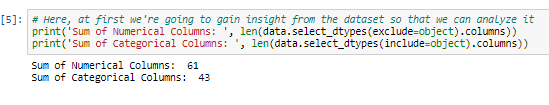
1. Classification

This is the first task that I perform which is include all the pre-processing and exploration of the dataset. Here are the steps :

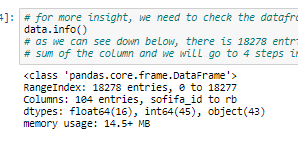
* Read the dataset



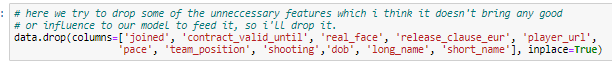
* Sum the categorical and numerical features



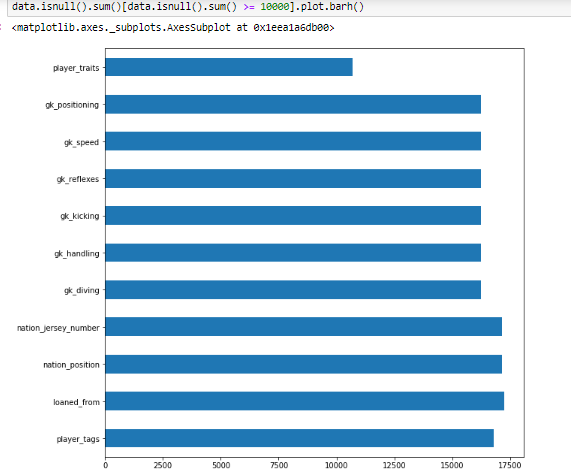
* Read the .info() of the dataset that consist into several types of data : int, float and object



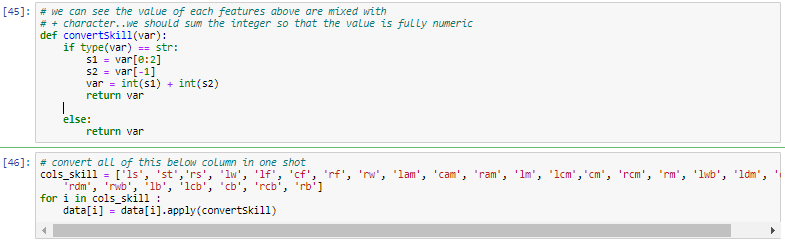
* Drop unnecessary column/features.



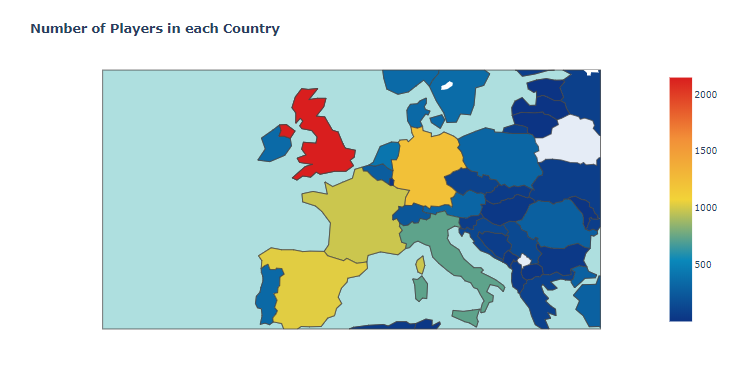
* Check missing values that more than 10 thousand rows and drop it, because it is impossible to perform imputation for this features.



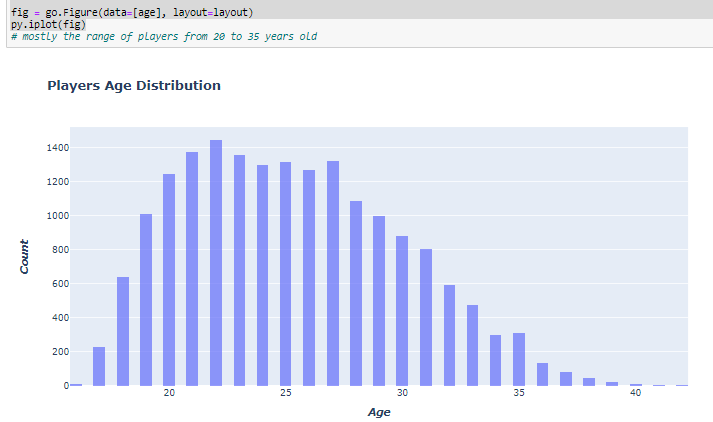
* Drop more features with missing value
* Then i drop player\_positions because it has no value, and I take skill converter to convert the skill columns



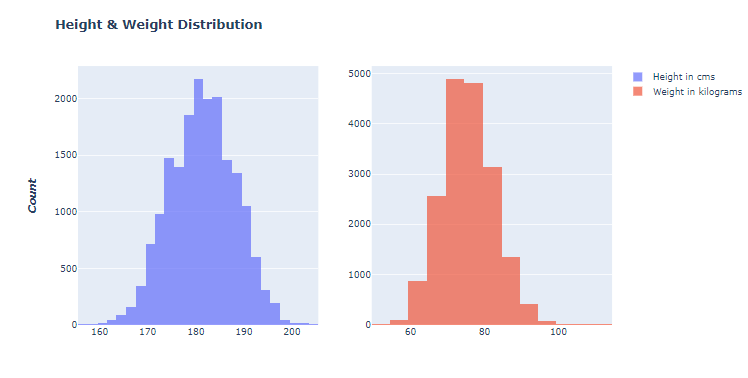
* Then convert position of the player into 4 category which is Midfielder, Goal Keeper, Defender and Forwarder.
* Drop values that exclude from the 4 category and use manual imputation that I code from scratch, and then explore more the dataset to see more features that is important or not.
* Explore more nationality by interactive earth map plot which is useless, I’ll drop them



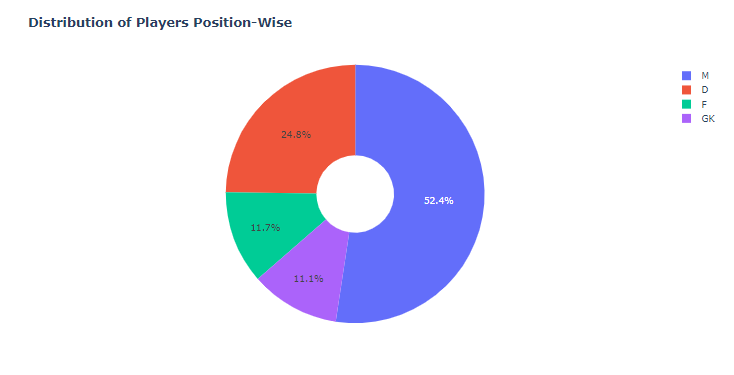
* Explore age which is also useless and need to be dropped



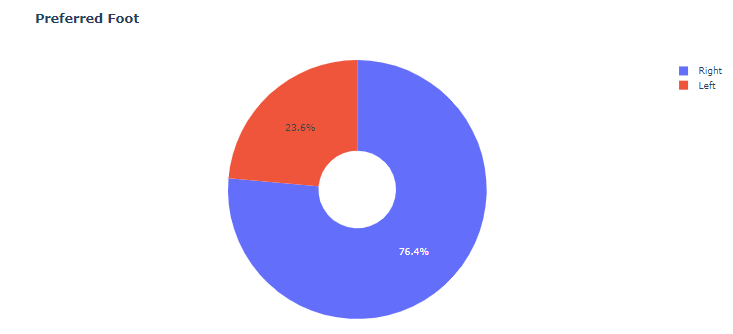
* Explore weight and height distribution player



* Because we’ve impute the missing value of player positions in midfielder position, that is why the distribution has major and minor



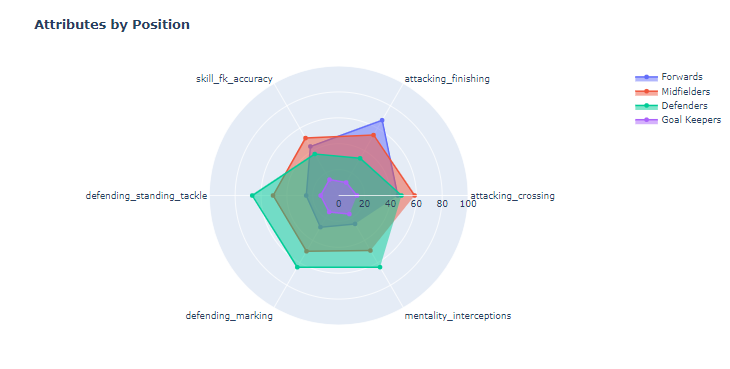
* See the preferred foot distribution which is useless and need to be drop



* Analyse work rate and overall with violin plot to see wether they are dependent or independent and the result is independent and they will be drop



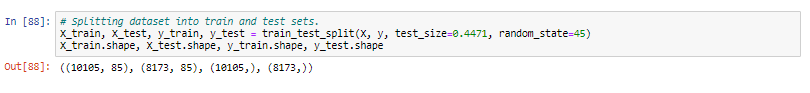
* I also plot the player positions skill to analyse their trend of skills



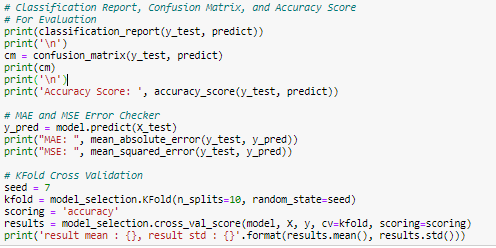
And now we will perform training, evaluation and propose model to deploy and there are 2 models to proposed here from this source which is Random Forest and XGBoost :



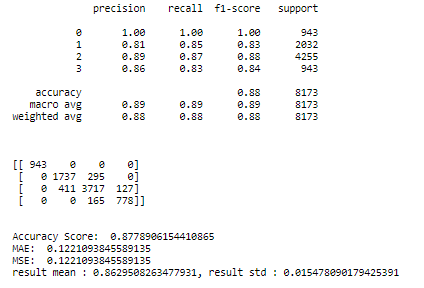
And the feature to classify is the player\_positions, and I split the data



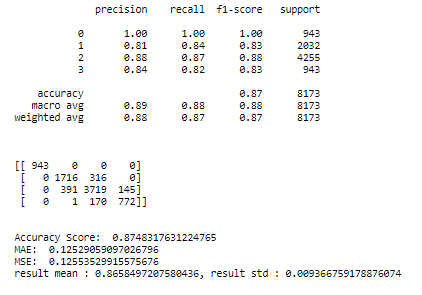
For evaluating the algorithm, I use confusion matrix, classification report, mean absolute error, mean squared error and KFold Cross Validation to check our result and avoid overfit and underfit



And this is Random Forest Result

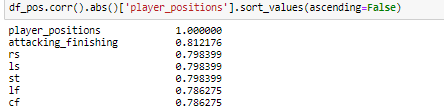


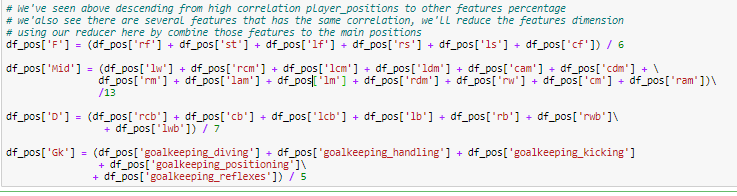
And this is XGBoost Result

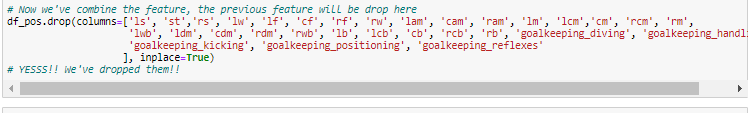


The produce the same accuracy and slightly different on error also the mean and std from Kfold Cross Validation

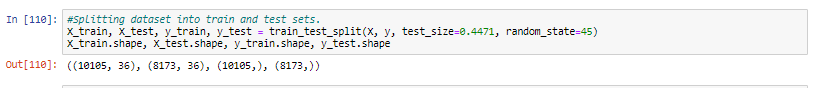
So here, we should think a way to improve the results, and the IDEA is feature correlation or feature importance, we will check the correlation of player positions with other features, low correlation percentage will be drop, the same percentage and high will be combined too into several new feature and the old features will be drop to reduce the dimention, this is what I call manual dimentional reduction so that we can have more insight and not change the semantics of the feature.







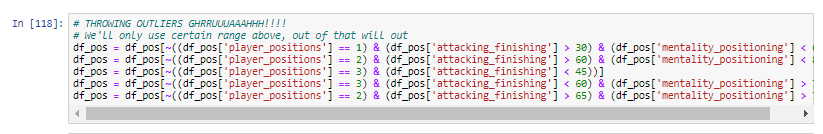
So after we reduce the dimension and drop low correlation features, we retrain the model with the same dimension of data :

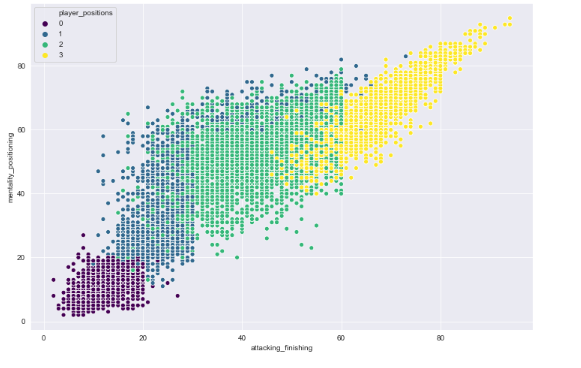


And the results of Random Forest and XGBoost doesn’t give much hope because it’s still the same with the last training, so here I think about outliers check and throw :



There are several outliers to be removed and here is the results:



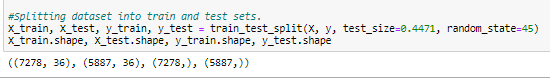


After we throw some outliers, I perform training technique called MULTI-LAYERS TRAINING. Here, i will explain to you that i use methods of multi-layers of training, what does that means? let me tell you a brief example, if i have 100% data, i will use 1/3 for test and the rest is for training and the second splitting down below the 2/3 of training that we have above with :

((10105, 36), (8173, 36), (10105,), (8173,)) dimension shape of data

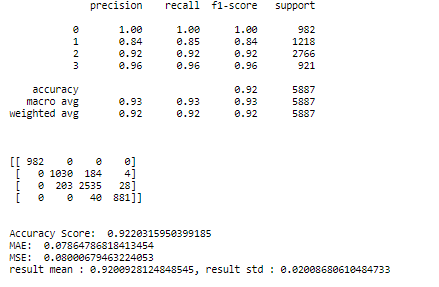
and it will be splitted again from that for test is 1/3 of the 2/3 data that we've left to improve accuracy and make sure OUR MODEL AVOID OVERFITTING and UNDERFITTING, our new data down below will be :

((7278, 36), (5887, 36), (7278,), (5887,)) dimension shape of data

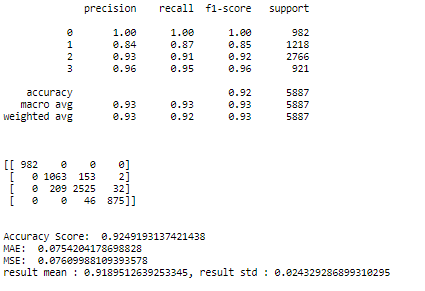


Note : test\_size = 0.4471 help to improve accuracy.

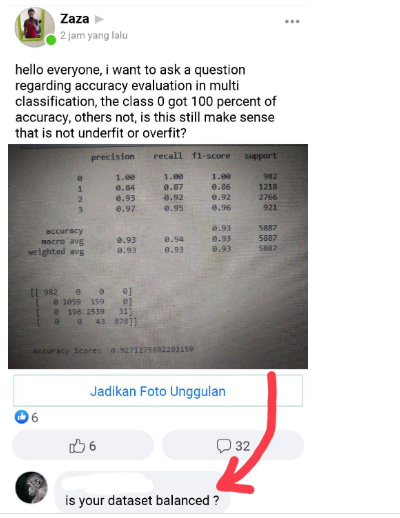
And with the result is increased, this is Random Forest Result :



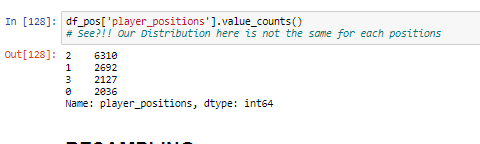
and XGBoost performs better with slightly 0.2 percent of difference :



In this state, I asked about my result in credible forum and I get something or methods to improve more my result, here is the result :

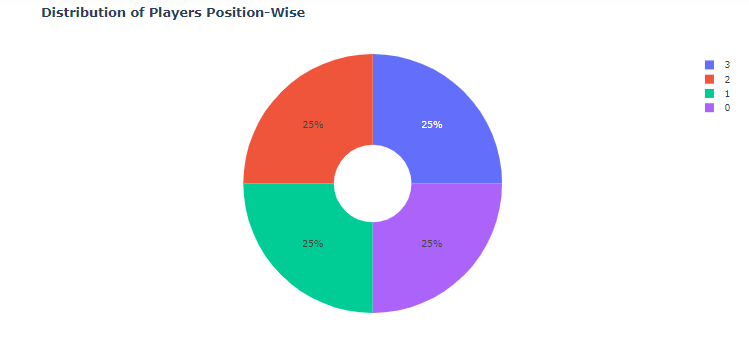


That is new problem, our dataset is not balanced and it could make our accuracy lower, so I use RESAMPLING methods for major player\_positions class , it was not balance before :



And we perform resampling here :

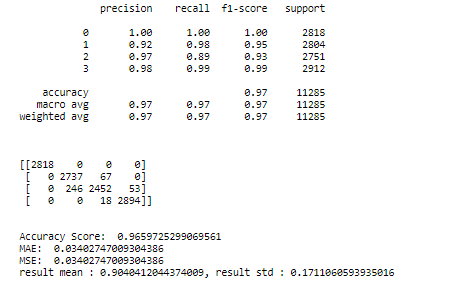


Now our player distribution is balance : 

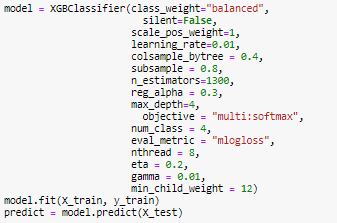
Because i have balanced my dataset, now the numbers of data is increased from originally 18278 to 25240 row of data. The dimension below is :

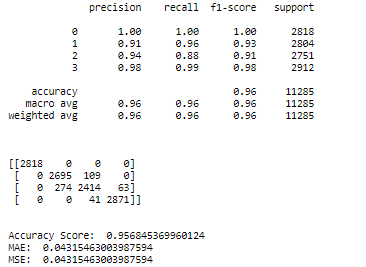
((13955, 26), (11285, 26), (13955,), (11285,)) shape of data.

We re-train our dataset and here is Random Forest result with tuned parameter random state = 1:



And here is XGBoost result with hyperparamter tuning :

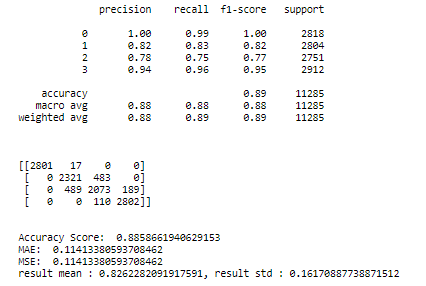




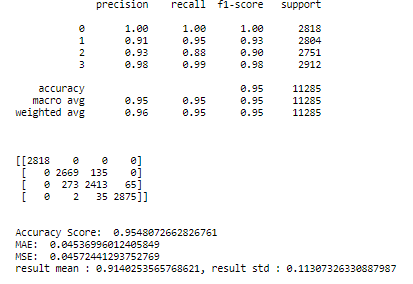
The accuracy is really increased ! because our dataset is balanced.

Note : the balanced dataset is saved into Explored Dataset.csv to use in clustering.

I also add some model to do more experience which is Logistic Regression and Decision Tree, here is LogReg result :



That is not really good for logistic regression in this dataset case, here is Decision Tree result :



It’s good, not much different with Random Forest and XGBoost.

**Evaluation, Conclusion and Solution.**

Now we have reach conclusion that from first training to the last, RandomForest shows good results better than XGBoost

but after re-cut the dimension by split or as we called as multi-layer of training oftenly, shows that the XGBoost is better than RandomForest with 0.2 difference with this scenario

1st training : RF = more than 86%, XGBoost = 86%

2nd training after correlation feature importance phase :

RF and XGBoost results is 87% with slight or small difference where RF is still better

3rd training with Re-cut by split by taking 1/3 of 2/3 data that we have left also throwing out OUTLIERS :

RF = 92.2 %

XGboost = 92.4</h2>

After we balanced the dataset our results is constant which is RandomForest is still better than XGboost with hyperparameter tuning(we add parameter to make more good prediction and it increase more than 0.1 to 0.2 percent), but the accuracy from resampling has increased a lot with good KFold cross validation, our accuracy is :

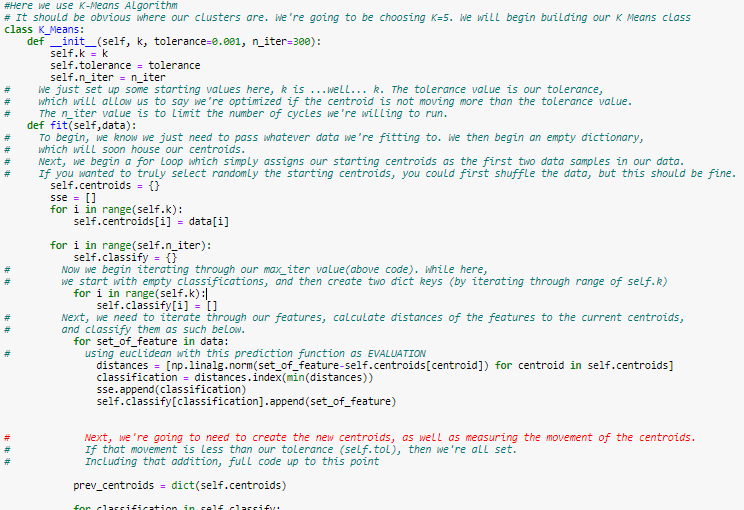
RF = 96% to 97%

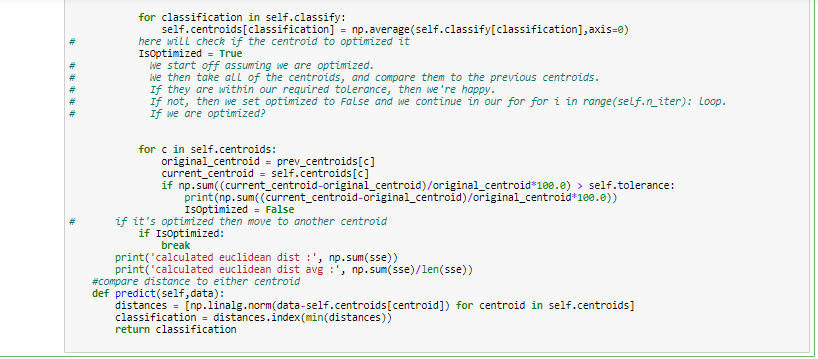
XGBoost = 95% to 96%

Note : i also add 2 model for more prediction, its Decision Tree and Logistic Regression the result is Decision Tree : 95% to 96% which is the same with XGBoost but Logistic Regression give 88% accuracy which is not really good compared to the other 3 model So, for this problem RandomForest is BETTER for MODEL DEPLOYMENT (SOLUTION) and second come up Decision Tree and XGBoost, and i do not recommend Logistic for result of my dataset of fifa.

1. Clustering Task

In clustering, i used the explored and preprocessed dataset so that I would not handle any preprocessing instead of exploration of feature engineering, and I use 2 evaluation metrics to analyse the best feature to cluster which is Euclidean distance and average distance, also the elbow method that I code from scratch, that is why the model that I use is K-Means Clustering Algorithm that I code from scratch consist to 2 model with different feature engineering, here is the brief architecture of the class :

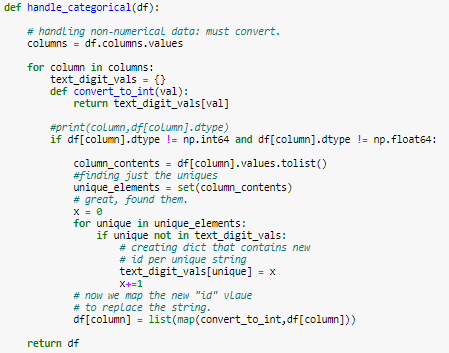




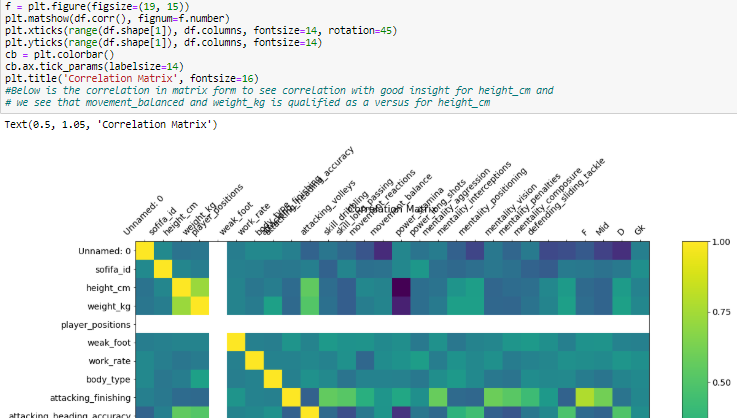
So I just need to explore the dataset again :



I convert categorical feature to numeric with this function :

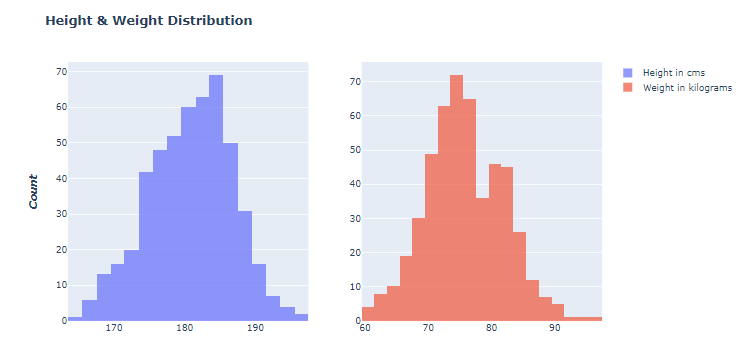


I explore the dataset feature correlation with correlation matrix and sorted correlation

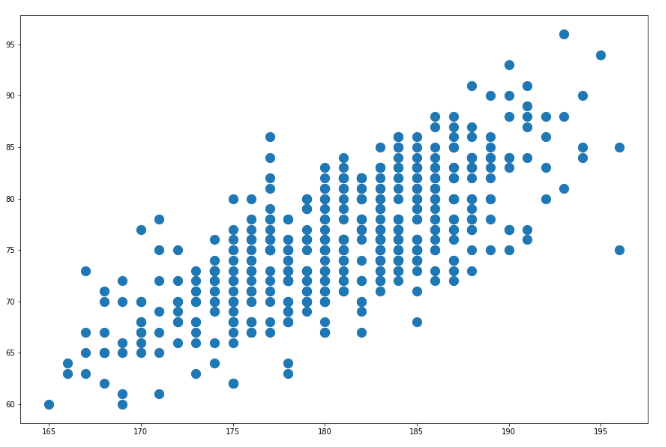


Note : in this case I only use 500 data because my laptop can’t run more than that.

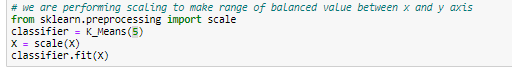
1. For the first experiment I use height\_cm and weight\_kg feature to cluster and here is the distribution :



And this is what the data look like before cluster :



And then I fit and scaled the data using 5 for k value:

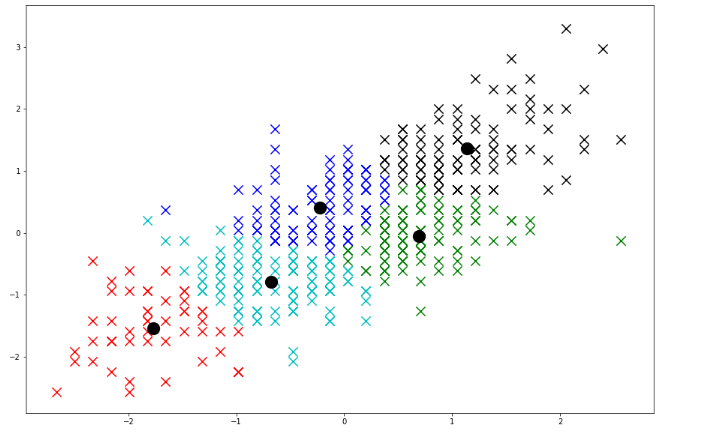


Scaling is important because we need to balance the X and Y axis with certain range of value so the distance calculation could be minimize

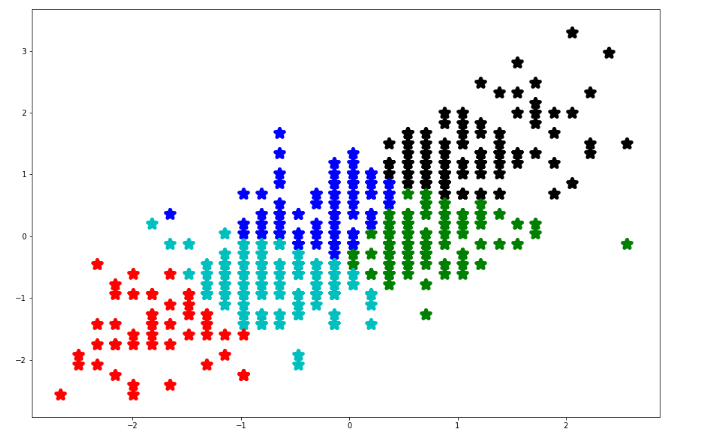
Here is the result of the distance evaluation for this feature :



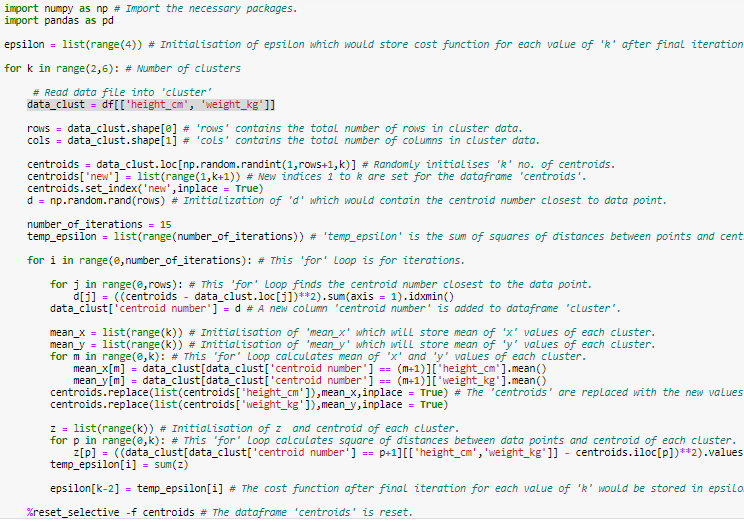
And here is the clustered result with k = 5 :



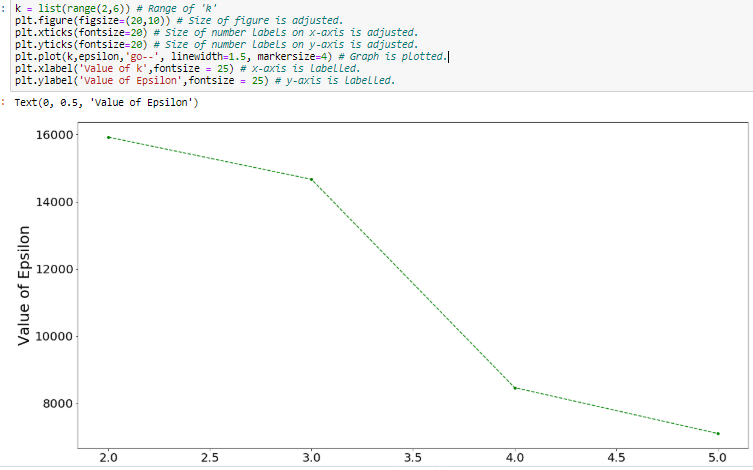
And I used predict function from my K\_means class using Euclidean distance to predict the correct data with their closest centroid and here is the result :



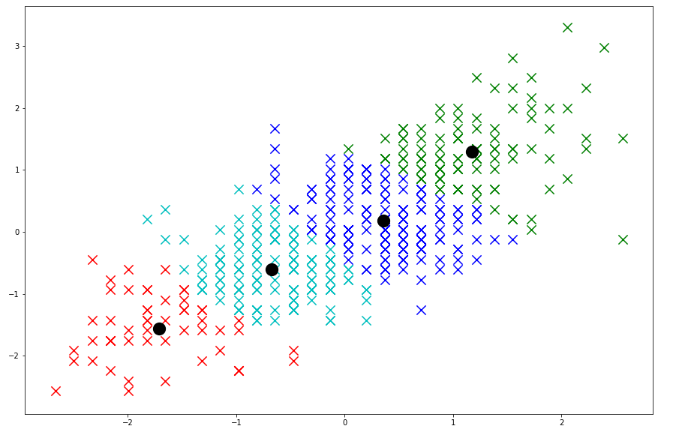
With that I evaluate the features and the model using Elbow method for optimum K value using this methods I code from scratch without library :

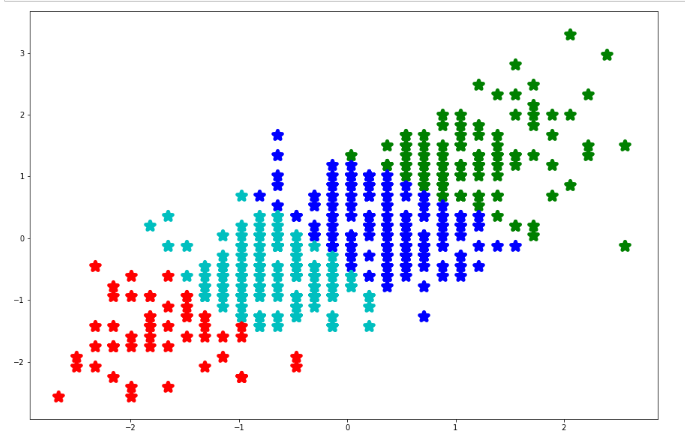


And then I plot the result :

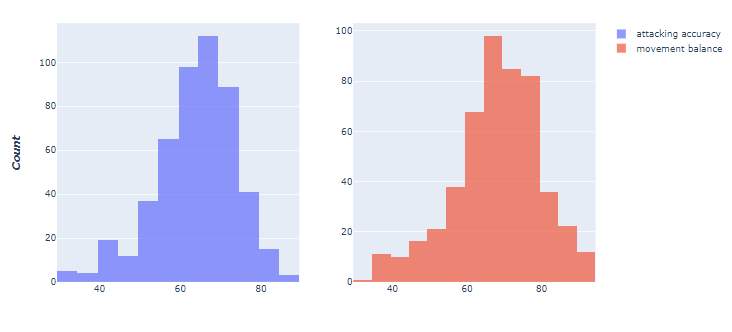
the optimum is at 3 and the best optimum is at 4, so I re-fit the dataset with K=4 and here is the result of distance metric, cluster and the prediction :



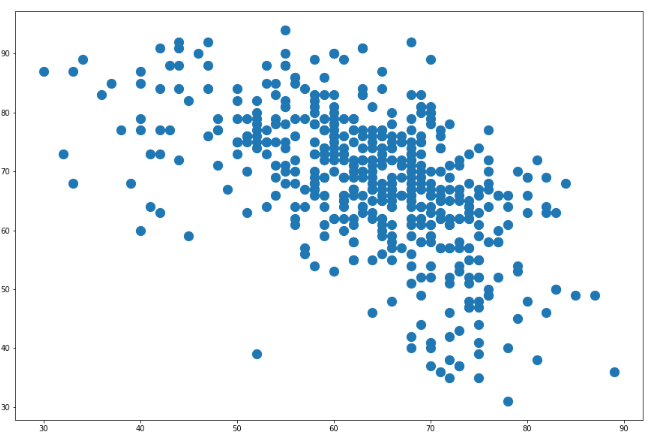




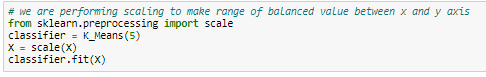
1. For the second experiment, I used the same way of exploration but different feature selection, but same number data and scenario which is movement\_balance and attacking\_headind\_accuracy and here is the distribution :



And here is the data before cluster :

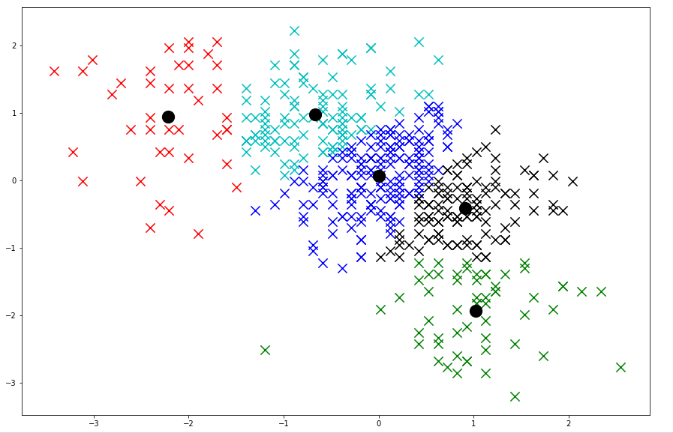


And then I cluster the data using K=5 :

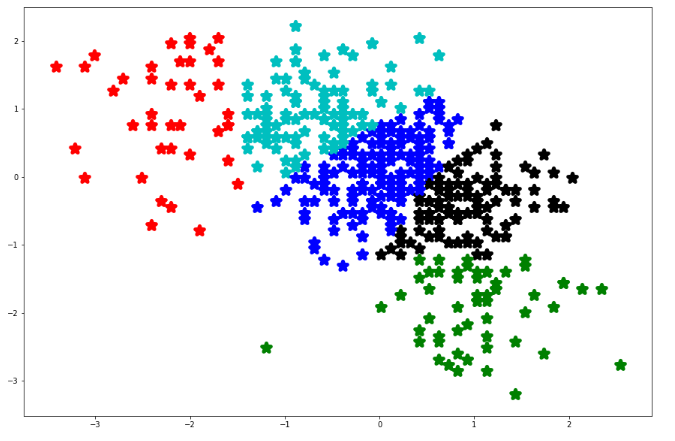


Here is the distance and cluster result :

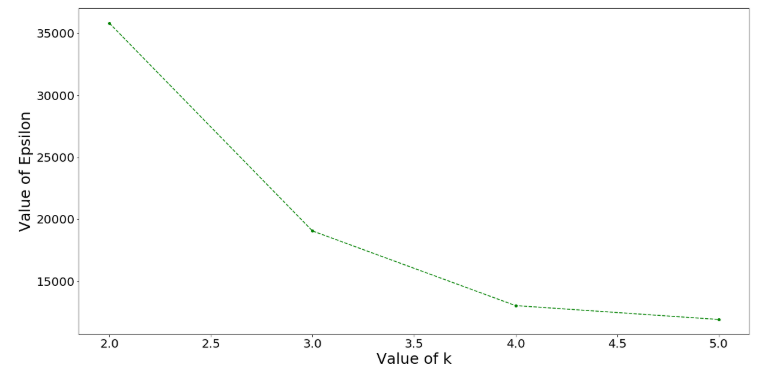




And we predict to perform the best clustered data for this features :

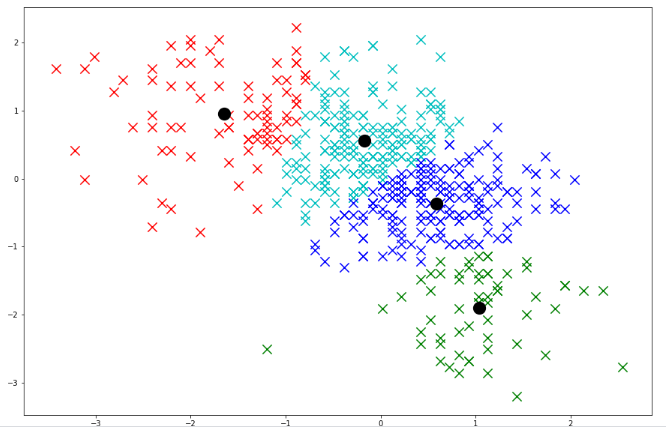


And then I find the best K for this feature using the same code from my first experiment and here is the best K metric :

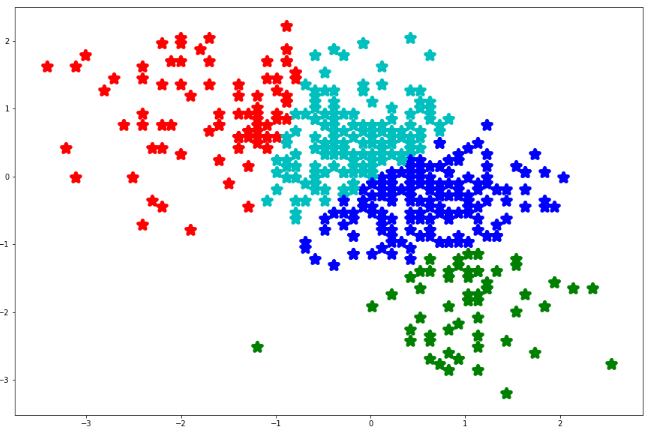


We also got 3 as optimum and 4 as the best optimum K, now we re-fit our data again using K=4 and here is the result of distance and clustered :





and we also predict for the cluster data and here is the result :



**Conclusion, Evaluation and Solution**

As we can see above experiment of first using height\_cm and weight\_cm giving the lowest average Euclidean distance 1.6++ using best optimum K after the evaluation which is K=4, the same goes for the second experiment using movement\_balance and attacking\_heading\_accuracy feature for cluster with the same scenario like the first experiment because the elbow method result giving the same best optimum K which is 4, but the distance is higher, it is 1.9++, so after we evaluate this number of distance and cluster and K number, we can take that the **first** experiment is the best and proposed to be match in clustering and not recommend the **second** experiment.

**Closing**

Inside of this task of classification and clustering consist into the preprocessing, model/algorithms and evaluation with conclusion followed with solution that is match for fifa case dataset, read it carefully to understand the order scenario for both task.

Best Regard.

Muhammad Ihza Mahendra

(1301174682)