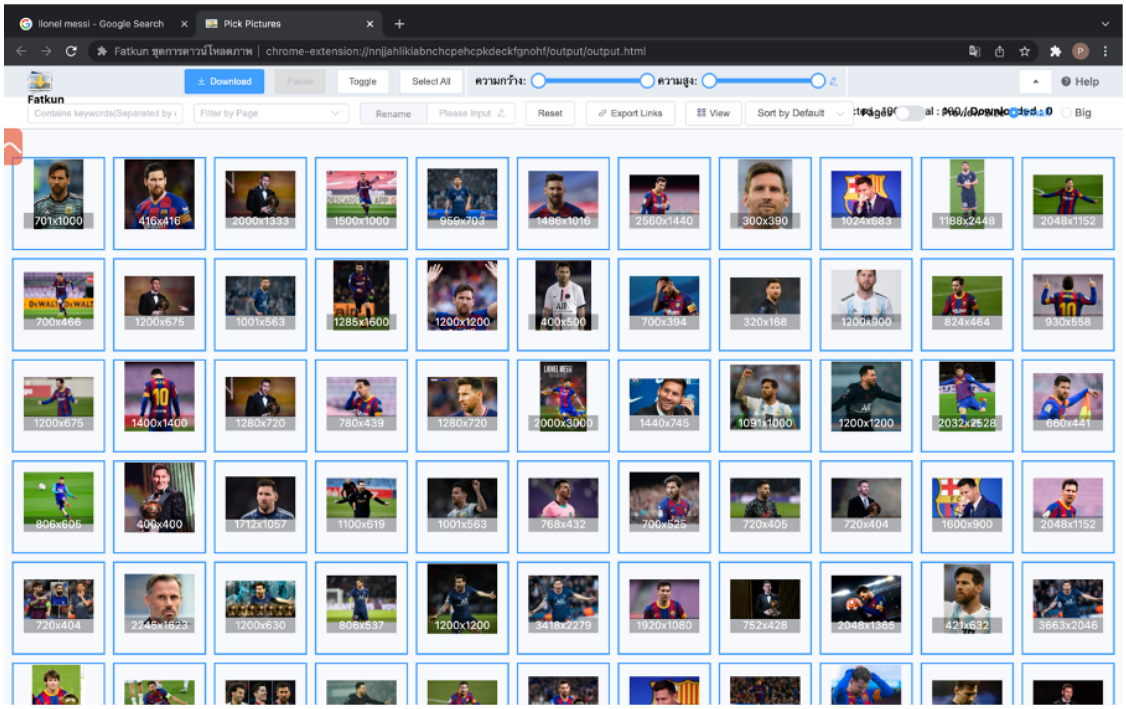
Image Classification of Football players project

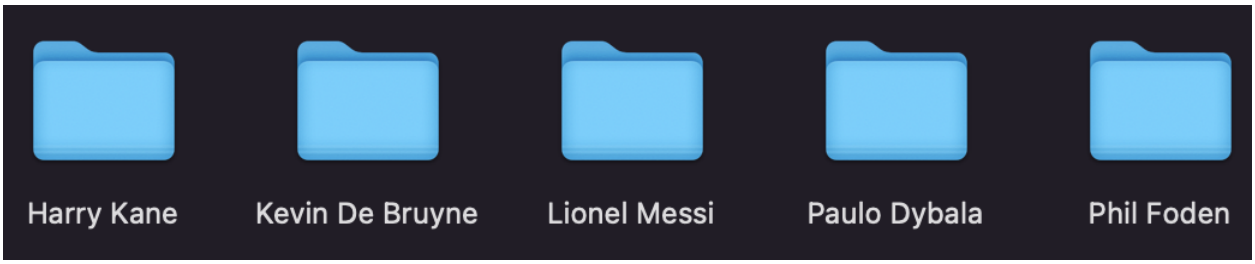
Pattadon Naksuwan

Main Objective: To predict which football player is from a given picture

Data set: Automatically download many photos using Fatkun from google photos



Since the datasets are pictures, the statistical descriptions can't not be represented, we have to transform from pictures into numbers first. There are 5 football players used in this project, which are Harry Kane, Lionel Messi, Paulo Dybala, Kevin De Bruyne and Phil Foden (5 classes). After the pictures of all 5 players are downloaded, they will be automatically kept in different 5 folders, each folder containing about 80-100 pictures of each player.

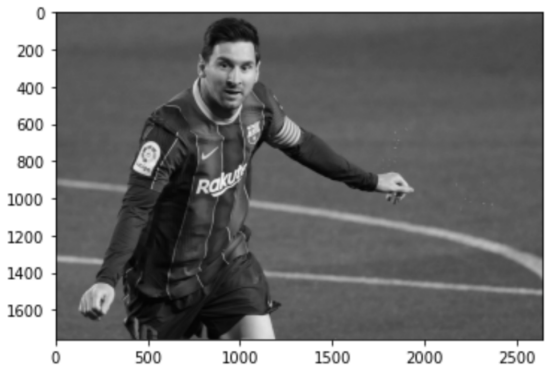


Data cleaning and features engineering:

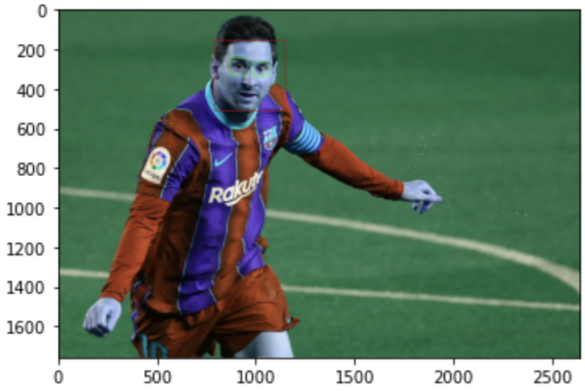
When downloading footballer pictures, the pictures can be selected manually to be downloaded. Therefore, the pictures that do not contain the footballer will not be selected. To create models with sklearn, the data need to be numbers. So, the pictures need to be transformed into numbers, and in this case the opencv library, Haar Cascade classifier and wavelet transformation are used to detect footballers in the pictures and extract the features.

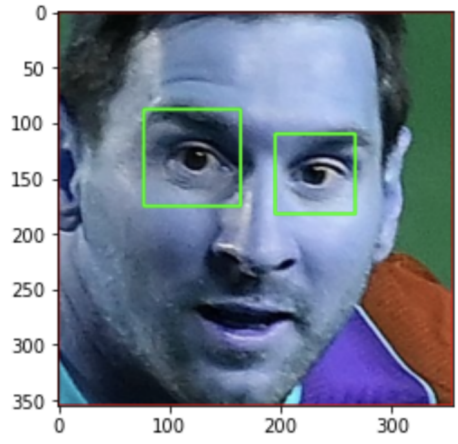
Steps of data cleaning and features engineering:

1. Use cv2(from opencv) to read to pictures from the downloaded files and change the color to gray color.

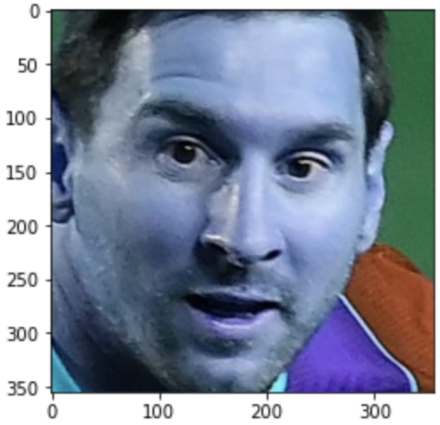


2. Use Haar Cascade face detection and Haar Cascade eye detection to detect the faces and eyes from the pictures.

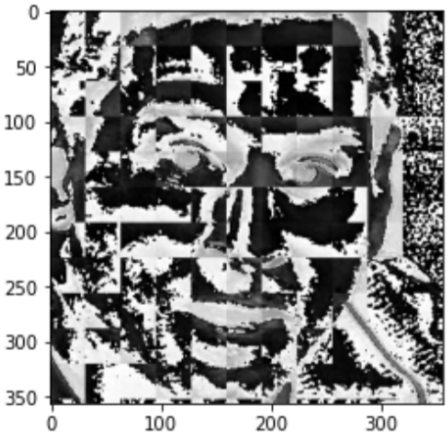




3. After faces and eyes are detected, now the pictures will be cropped to only faces of the players, which means other parts of the pictures will not be used. Some pictures cannot detect the faces of the players, so those pictures will also not be used.



4. The cropped pictures will now be transformed to arrays containing numbers by using wavelet transformation. (Wavelet Transformation will give the important features in the faces as numbers.)



5. The x variable will be the raw images read from cv2(will now be array) and the array from wavelet transformation stacked together. The x variable now will be an 2d array that contains 218 elements (218 pictures of all players) and each element will have 4096 features. So the dimension of the array is (218, 4096).
6. The y variable will be a list of 0-4 (5 classes/players) corresponding with the pictures.
7. After checking that the data is not imbalance data, the data will be splitted to x_train, x_test, y_train, y_test using train_test_split. Moreover, the x variable needs to be scaled, therefore the standard scaler will be applied. And now, the data is ready to be trained.

Classification Models:

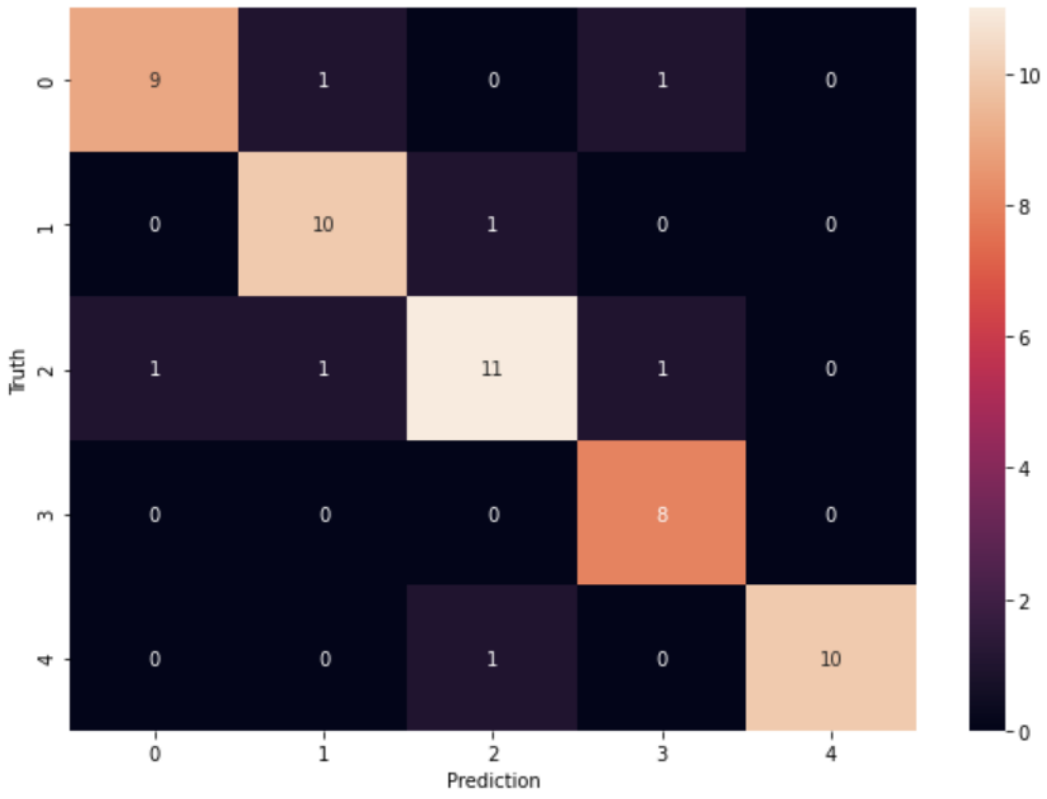
In this project, three classification models, namely Support Vector Machine, K-Nearest Neighbors and Logistic Regression, are used.

Support Vector Machine(SVM):

GridSearchCV will be used for hyperparameter tuning (for every model), in this case there are three hyperparameters to consider, which are C, gamma and kernel(linear or rbf). Eventually, we have the best SVM model with C = 0.0005, gamma = 0.0001 and kernel = 'linear'.

This is the performance of the model from classification_report and confusion_matrix.

	precision	recall	f1-score	support
0	0.90	0.82	0.86	11
1	0.83	0.91	0.87	11
2	0.85	0.79	0.81	14
3	0.80	1.00	0.89	8
4	1.00	0.91	0.95	11
accuracy			0.87	55
macro avg	0.88	0.88	0.88	55
weighted avg	0.88	0.87	0.87	55

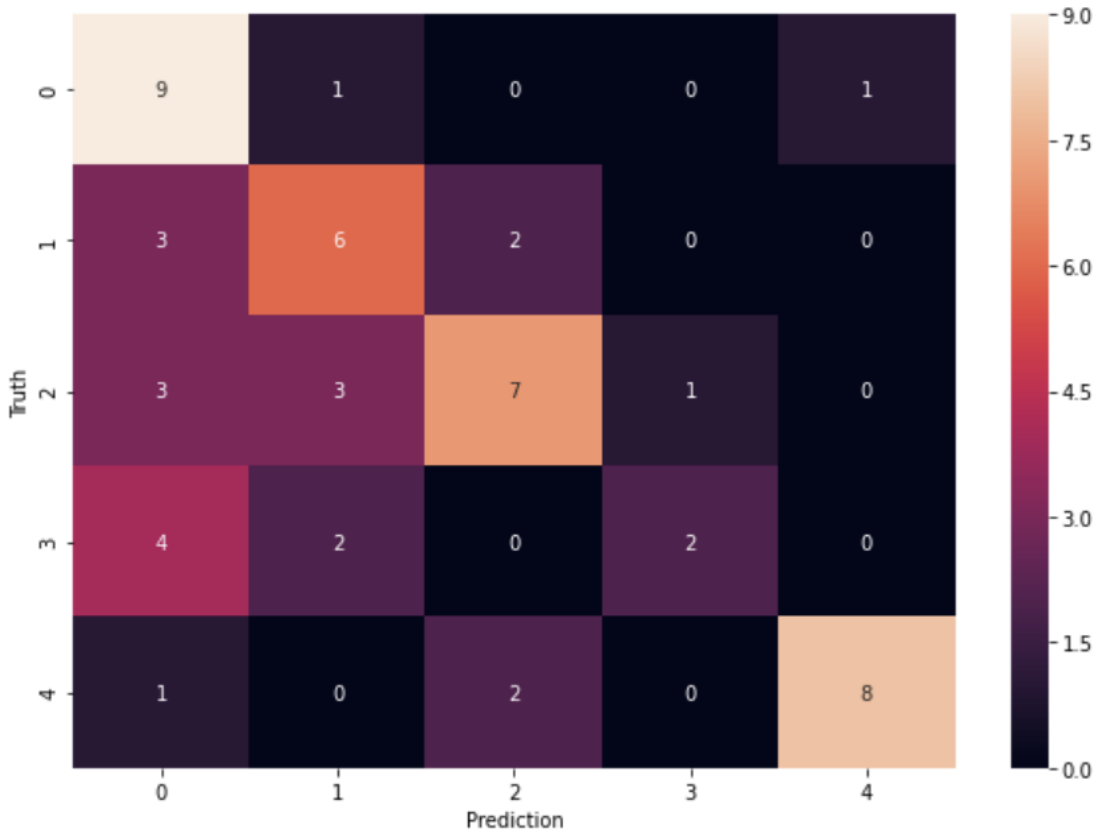


K-Nearest Neighbors(KNN):

Now, the hyperparameters are k value. In this case, k is the value between 1-30. And the result is we have the best KNN model with k=4.

This is the performance of the model from classification_report and confusion_matrix.

	precision	recall	f1-score	support
0	0.45	0.82	0.58	11
1	0.50	0.55	0.52	11
2	0.64	0.50	0.56	14
3	0.67	0.25	0.36	8
4	0.89	0.73	0.80	11
accuracy			0.58	55
macro avg	0.63	0.57	0.57	55
weighted avg	0.63	0.58	0.58	55

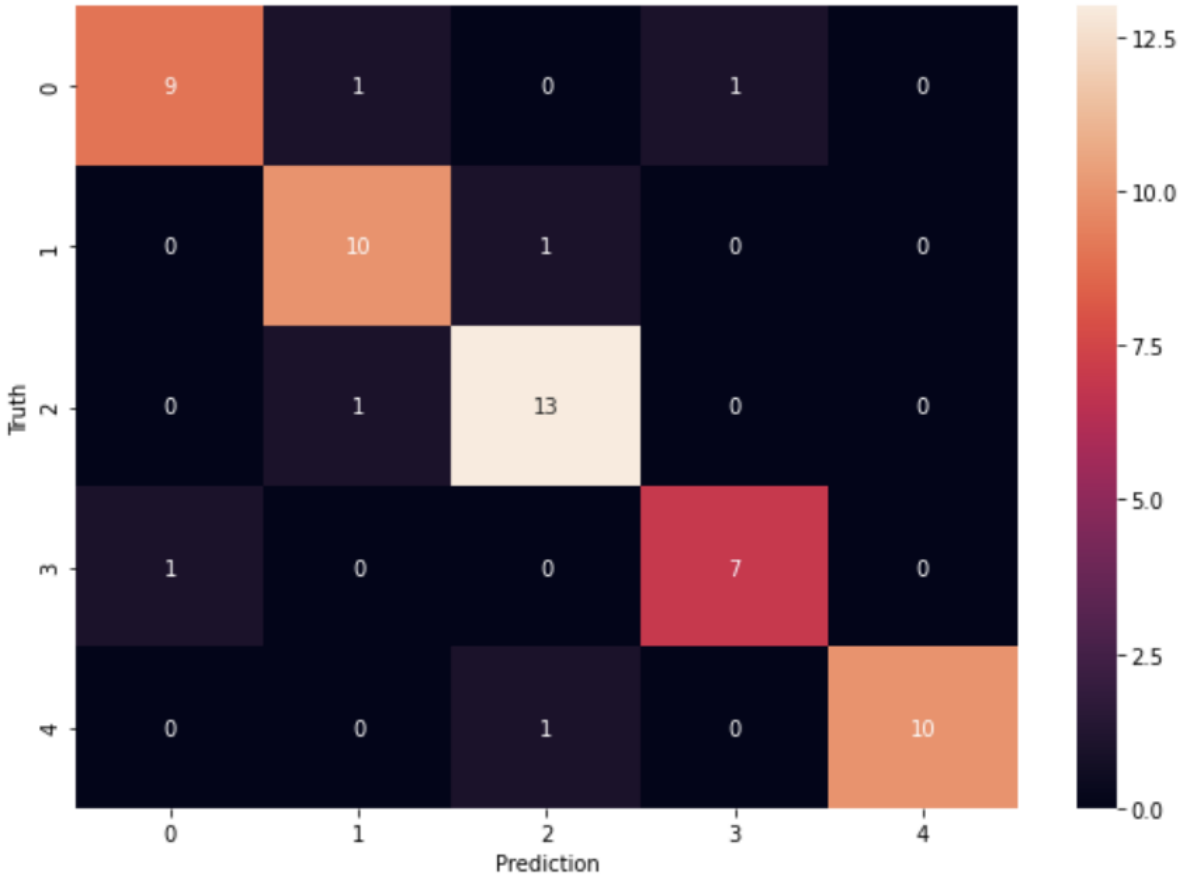


Logistic Regression(LogReg):

The hyperparameters are C(values from -5 to 5) and penalty (l1 or l2). The result is we have the best LogReg model with C = 46.42 and penalty = 'l2'.

This is the performance of the model from classification_report and confusion_matrix.

	precision	recall	f1-score	support
0	0.90	0.82	0.86	11
1	0.83	0.91	0.87	11
2	0.87	0.93	0.90	14
3	0.88	0.88	0.88	8
4	1.00	0.91	0.95	11
accuracy			0.89	55
macro avg	0.89	0.89	0.89	55
weighted avg	0.89	0.89	0.89	55



Model Selection:

Models	precision	recall	f1-score	accuracy
SVM	0.88	0.87	0.87	0.87
KNN	0.63	0.58	0.58	0.58
LogReg	0.89	0.89	0.89	0.89

According to the picture above, we can see that Logistic Regression has the best performance in all aspects, but it is not far away from the Support Vector Machine model. For K-Nearest Neighbors, the performance is not good enough to be a classification model for this project. Since Logistic Regression has the best performance, it is chosen to be this project's classifier.

Summary:

Since the objective of this project is to predict which football player is, given a picture, the best score model/Logistic Regression model is chosen. Both Logistic Regression and Support Vector Machine predict very well with the f1-score of 0.89 and 0.87 accordingly, but K-Nearest Neighbors does not predict the outcome well with the f1-score of only 0.58.

Suggestion:

To improve the project, a web that can be used to insert the picture that we want to predict will be interesting. Moreover, using the stacking technique can help improve the model with even higher score.