Final Project
Introduction to Data Science 2020/21

FOOTBALL RESULTS PREDICTION USING 'K-NEAREST NEIGHBORS' ALGORITHM

Niccolò Parodi S4668271

Emanuele Prella S4636470

May 31, 2020

CONTEXT AND MOTIVATIONS:

Datasets:

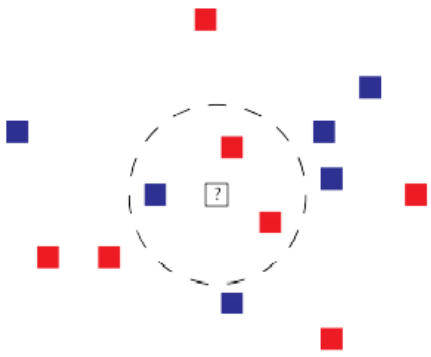
- **Fifa 21 Complete Player Dataset** from **keggle.com**
- **European Leagues 2019-20 Match Results** from **football-data.co.uk**
- **Champions League 2019-20 Match Results** from **fixturedownload.com.**

The report aims at predicting the 2020-2021 rankings of the 5 major European leagues (Serie A, Premier League, LaLiga, Ligue 1, BundesLiga) and the 2020-21 Champions League competition rounds, starting from the results returned by our model (using the previously mentioned datasets) and using the classification algorithm 'K-Nearest Neighbors' .

Analysis:

Using the values of the players taken from the FIFA21 dataset and grouping them by team, we obtain an evaluation (divided by fields, for example: age, attack, defense, passing, dribbling, physicality and many others) of each team. Players whose role is 'Goalkeeper' are considered separately because their statistics are different from those of other roles.
The use of this dataset allows to have an updated, consistent and real evaluation of the actual performance and "strength" of the teams, based on the average statistics of the players that compose it.
K-Nearest-Neighbors (KNN), one of the most versatile lazy algorithms, is used.

The KNN algorithm is a classification algorithm, but it is also widely used for predictive analysis. KNN is based on the idea that starting from classified data points, it is possible to classify a new one by looking at the K data points closest to it. This means that the algorithm uses an instance-based learning, where the training data set is used to classify the unknown data point. As can be seen from the figure below, the colored squares (red and blue) are classified, while the white square is the one to be classified. In this case, the K chosen is three and by majority vote the white square will be classified as red.



To determine which points are the closest to the one to be classified, the Euclidean distance between all classified and unclassified data points is measured.

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

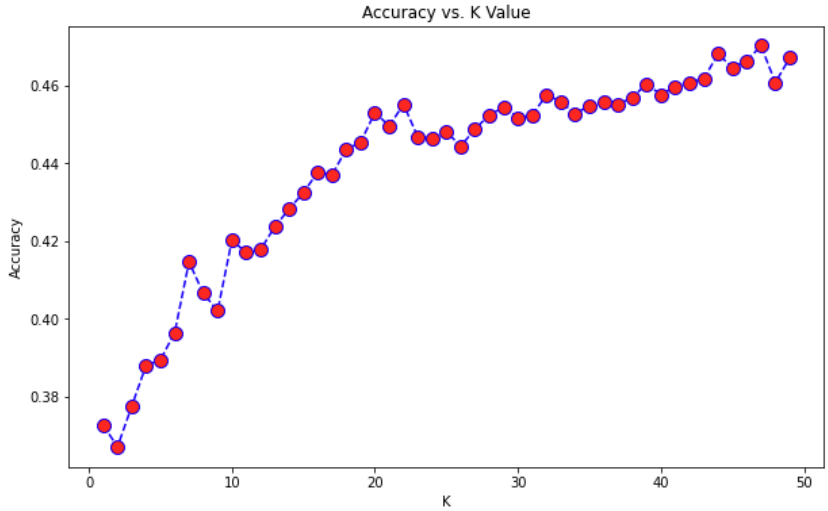
Large K-values are generally more accurate than small K-values because they eliminate noise-generating factors, but they are computationally more expensive.

Before making actual predictions, it is always a good idea to resize features so that all can be evaluated uniformly, so the MinMaxScaler function from the Scikit-Learn library is imported.
MinMaxScaler normalizes variables in a standard range [0,1], the range in which floating-point values have the highest precision.

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

The scaler is then fit with the .fit() function that reads the training data and makes an estimate of the maximum and minimum values that may be encountered.

The program has at its disposal 3400 results of matches of 2019/20 between leagues and international cups on which to train the model, after several tests, the best choice is to use 85% of the data as training and the remaining 15% of the data as tests.
As shown in the graph the ideal number of K to increase the accuracy of the model is



K=47, an odd number is chosen so as not to have situations of parity of votes. The

program arrives so to guarantee a 55% of accuracy of the result, percentage in line with that of other studies[1].

	precision	recall	f1-score	support
0	0.27	0.07	0.11	87
1	0.56	0.81	0.66	194
2	0.58	0.49	0.53	148
accuracy			0.55	429
macro avg	0.47	0.46	0.44	429
weighted avg	0.51	0.55	0.51	429

The F1 score takes into account test precision and recovery, where precision is the number of true positives divided by the number of all positive results, and recovery is the number of true positives divided by the number of all tests that should have tested positive (i.e. true positives plus false negatives). The F1 is calculated by the harmonic mean of precision and recovery:

$$F_1 = \frac{2}{\frac{1}{\mathrm{r}} + \frac{1}{\mathrm{p}}} = 2 \cdot \frac{\mathrm{p} \cdot \mathrm{r}}{\mathrm{p} + \mathrm{r}}.$$

For each league, after adding the newly promoted teams and eliminating the relegated ones, all the matches (round-trip) between the participants are created. 3,1,0 points are assigned for victory, draw or defeat respectively. The value "0" indicates a tie, while "1" and "2" win the home or away team.

The Champions League groups and participating teams are imported from an external .csv file. The program calculates the rankings of all 8 groups, then the two highest scoring clubs in each group are selected, i.e. the 16 teams that are guaranteed passage to the knockout stage.

PROTOTYPE DESCRIPTION:

Language used:

Python Notebook (.ipynb)

Libraries used:

- **Numpy, Pandas** for dataframes management
- **Matplotlib** for graph creation
- **Sklearn** for predictive model creation and training

Interesting code excerpt:

Model Creation and Training

```
# from sklearn.model_selection import train_test_split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.15,random_state=6)

# Import the scaler and fit train and test data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

# Train the model using an initial random k value (3)
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,confusion_matrix
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train,y_train)
pred = knn.predict(X_test)
```

Testing to find the best K

```
# Accuracy can be improved, so we try to get Error Rate for k values up to 50
# Where we see error rate gets flattened, we can choose on an optimal k value for our model

from sklearn.model_selection import cross_val_score
error_rate = []

# Check the k values up to 50 and see how error rates are changing
for i in range(1,50):

    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_i = knn.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))

plt.figure(figsize=(10,6))
plt.plot(range(1,50),error_rate,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')

k_scores = []

# Check the k values up to 50 and see how accuracy is changing
for i in range(1,50):

    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_i = knn.predict(X_test)
    k_scores.append(cross_val_score(knn, X, y, scoring='accuracy', cv = 5).mean())

plt.figure(figsize=(10,6))
plt.plot(range(1,50),k_scores,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Accuracy vs. K Value')
plt.xlabel('K')
plt.ylabel('Accuracy')
```

Achievements:

Below are two of the results obtained from the prediction, one related to the Italian league and one related to the Champions League.

2020-21 A Series

HomeTeam	Total Points
Inter	111
Juventus	108
Milan	91
Roma	90
Lazio	88
Napoli	87
Atalanta	84
Sampdoria	51
Udinese	51
Parma	49
Torino	46
Genoa	44
Hellas Verona	42
Cagliari	41
Sassuolo	36
Benevento	33
Bologna	31
Fiorentina	29
Spezia	13
Crotone	5

The model predicts Inter as the winner after 9 consecutive championships obtained by Juventus. Inter will then actually be champion of Italy for the 2020-2021 season, thus confirming the goodness of the prediction.

Champions League

The model succeeds in predicting 13 of the 16 teams past the Champions League group stages (2020-21) with a percentage of 81.2 percent

Group	HomeTeam	Total Points
H	Paris Saint-Germain	18
	Manchester United	12
G	FC Barcelona	15
	Juventus	15
F	Borussia Dortmund	18
	Lazio	7
E	Sevilla FC	18
	Chelsea	12
D	Ajax	15
	Liverpool	15
C	Manchester City	18
	FC Porto	9
B	Real Madrid	18
	Inter	12
A	FC Bayern München	18
	Atlético Madrid	12

Knowing the finalists of May 29 and using the *result()* function, which takes as input the names of two teams and returns the hypothetical result, Manchester City is favored over Chelsea. Prediction in line with that of the bookmakers.

HomeTeam	AwayTeam	Results
Manchester City	Chelsea	1

Man City - Chelsea	1.91	3.43	4.35

DIVISION OF LABOR:

Both group members participated in the entire creation of the project.

BIBLIOGRAPHY / SITOGRAPHY:

[0] **FOOTBALL RESULT PREDICTION USING SIMPLE CLASSIFICATION ALGORITHMS, A COMPARISON BETWEEN K-NEAREST NEIGHBOR AND LINEAR REGRESSION - PIERRE RUDIN**

[1] **Imperial College London - Predicting Football Results Using Machine Learning Techniques**

[2] Dr Saed Sayad - K Nearest Neighbors - Classification