FIFA-19 The Biggest League

yunus salman shaik, MSc Big Data Analytics, LYIT

***Abstract*—FIFA which is defined as ’Federation Internationale de Football Association’ is the greatest league on the planet. The information in the data-set is colossal as the players enrolled in the database are almost 18k+ and the extraordinary qualities for the players are more than 17k. As the data is gigantic in the data-set, playing out the activities for this data-set is the greatest undertaking. The amazing players like Messi, Ronaldo, and Pogba are notable around the world.FIFA-19 data-set contains 85 sorts of qualities for every single player, as no other player is like one another, these properties separate the player from different players Be that as it may, if any of the players get harmed in this game, the substitution for them is an enormous undertaking.For this data-set processing and visualization we have setup a environment in databricks and used two algorithms for performing the operations, linear regression and K-Nearest Neighbor algorithms are utilized. We have used python and pyspark language for our coding with utilizing the libraries.**

***Index Terms*—FIFA, Data-set, Linear regression, K-Nearest Neighbor Algorithm, Visualization,python**

I. Introduction

Sports have a wide scope of advantages including fitness promotions, entertainment and rivalry purposes. Among different games, football is a famous game around the world over and numerous apprentices are beginning to learn it. Football the sport of the world which many people refer as, the game is huge and fans are insane for this sport. FIFA-19 is the biggest gaming league ever released with thousands of players included in the league from all over the world and all the famous football legends are participating in the league.Many players like Lionel Messi, Christiano Ronaldo, Paul Pogba have been included in the league for better championship and tough competition for all other teams.

Monitoring all the players attributes and their strengths and weaknesses are major consideration for other teams. So data analysis and data visualization plays a major role in these aspects. For better analysis and visualization we have been opted an cloud environment in databricks.com which will help in executing the algorithms and all other purposes. Without the algorithms, we cannot find the solution for our questions, so we have opted for two algorithms which are linear regression and K-Nearest Neighbour for performing the essential operations on the colossal data-set.The programming language we have opted is python and pyspark, which plays a major role as it is very flexible and libraries are very handy for all the tough tasks.

Environment is crucial aspect whenever we are performing any operation with any type of data-set with any number of algorithms. A environment need to be pretty flexible with all type of programming languages and libraries. Databricks.com website allows to create any type of cluster in spark for performing operations using the algorithms.Python used as a coding barrier for algorithms as it has a variety of libraries which can perform any type of optimization and performance analysis on the data-set.

II. Related Work

A colossal amount of papers have been submitted on football analysis and its injury analysis, their have been papers also on how youth soccer which comes under-17 age group will impact the game analysis on professional football.A interesting paper submitted on how a altitude of ground will impact the performance of the players.we will just have a glance on the papers submitted by different authors on these aspects.

A paper submitted by J.H.williams on ”Relative age effect in youth soccer: analysis of the FIFA U17 World Cup competition” explains about how the youth with the age group of under-17 has been a good impact on the professional football tournaments as the youth are very well built and they are quite young so that the performance and the sprint speed is high compare to professionals. The major drawback has been their game play, as they are very young the skills they maintain are very less compare to professionals, as the senior players are much more skilled and the goal accuracy is very high than youth soccer.[2]

A paper submitted by Nassis, George p on ”Effect of Altitude on Football Performance: Analysis of the 2010 FIFA World Cup Data” explains a significant aspect of the FIFA world cup that held in 2010 in the country South Africaa. The paper succeeded on how a altitude of the ground shows a significant impact on the players performance. As we know that south Africa is very popular for its high altitude football grounds, as the altitude increases the level of oxygen decreases, as the players take less oxygen it will effect the Adenosine Triphosphate(ATP) which will impact on the uptake of the oxygen and it will directly impact on the performance.[1]

A bunch of papers and technical reports have been submitted on the external aspects of the sport, but the player internal performance and their attributes haven’t been fully analysed in any of these papers and reports.So this technical report will be mainly focused on the player performances.

III. SYSTEM OVERVIEW

The data-set picked for the technical report is ’FIFA-19’ gave by ’karan gadiya’ from the kaggle site. Ascribed attributes for every player took a crack at the latest arrival of FIFA 19 database. The data-set not just contains the names and ages of the players. The data-set is pre-occupied with 89 unique sections and every segment has been loaded up with various kind of attributes. All these 89 attributes are concentrating on every single player.For, an occurrence we can exhibit that an individual player has the traits like left foot or right foot, straight forward or center forward, has a country with India or France, Playing for a club or for a nation, a player is a penalty kicker or an goal keeper.

The data-set appropriately bode well, since when playing out the data analysis and data cleaning on these qualities, a specific data analyst can’t just take away(drop) any attributes from the data-set. As their will be no examination between and goal keeper and a penalty kicker. So the obligation is significantly centered around the data analyst who playing out the examination and perceptions on the huge data. The attributes are only limited but the rows of data in these columns are very huge, their is nearly 18000+ rows of data of all the different players, as we can simply mention that the players registered in the FIFA-19 data-set is more than 18k+. So when any performance analysis performed on these big data is most complicated job. so opting for best possible algorithms are important aspect.

* **MY github link**

***FIFA-19 Dataset**

IV. IMPLEMENTATION

The implementation has been done in the form of stages where each and every implementation has been done in a flow,we have followed the ETL process in the implementation, where extract the data-set from the source and transform the data-set, transformation includes all the aspects like, data analysis, data cleaning, visualization and modelling and at-last pushing it into the algorithms.[5]

* Stage-1 Import data-set from the external source (Kaggle) and push it into the cloud based environment databricks.com.

Databricks builds up an online stage for working with Spark, that gives robotized bunch the board and IPython-style notebooks. Notwithstanding building the Databricks stage, the organization is co-arranging monstrous open online courses about Spark and runs the biggest meeting about Spark - Spark Summit.

After importing data-set we need to setup a environment in the cloud based platform, we have chosen the environment as which is a run-time version ”6.3 Machine learning(Scala 2.11, spark 2.4.4) and then create a python notebook to upload data-set.

* Stage-2 Data transformation is the second part of our implementation.

After importing the data-set in the python-notebook we will just check the data-set by using the command ’data.head()’ or ’df.head()’ this command will display all the attributes and their values (data) of your data-set.

After the analysis part you will find out what are the essential attributes you need for performing the algorithms and finding out the predictions and solutions for your questions raised.In the data cleaning by using the command ’data.drop([’column’],axis=1,inplace=True)’, this command will drop the columns which you don’t need.

A simple hack in the databricks environment is , whenever the error displays about the library in the code which we are performing we can simply import the library in the command prompt itself and get rid of the error, the command is ’percentagesh pip install library’.All the libraries can be installed in this way without much messy in the coding part.

The dropping of attributes has been done then we will perform the cleaning of null values from the data, as we all know that the null values from the columns wont perform any operations as it will always shows the error value.The following command will work to find out the null values ’data.isnull().any()’.By performing this command it will display the null values in your attributes,if it displays false their is no null, if it displays true their is null values in the attributes.

we will analyse the data of the attributes, if the columns with the null attributes are useful for the data-set we will change the null values into false as assigning any integer value to it. If the attributes are not useful we will simply drop the columns from the data.

* Stage-3 Data analysis is the third part of the implementation.

In this stage we will perform the data analysis, as the name indicates analysing the data, as we have dropped the unnecessary columns and unnecessary null values from our data, we will perform the analysis on the refined data.The analysis is much more complicated that it looks.

As compare to other data-sets, the fifa-19 data-set has 89 attributes or columns, which are very hard to compare. so, we will perform analysis only upon our requirements, so the comparison will be done on limited number of attributes and then we will push the analysed data into testing and training purpose.

*Stage-4 Visualizing the data is the fourth implementation.

Visualising is the good aspect for the data, as we will see what are the attributes are performing in a visualization mode. Comparing the two attributes and display the results in the form of scatter plots, mat-plots, bar graphs and maps are the process we will do in the visualisation. Their is a problem with the databricks environment, whenever we are performing the data visualisation as it will just execute the command and displays nothing, for proper visualisation of any image we need to put a ’display()’ command at the end of every code for data visualisation.

*stage-5 Training and testing the data is the fifth implementation.

After all the four stages we need to perform the testing and training our data-sets as we need to know with how much precision our algorithm will perform its operations. Usually the testing and training will be done in 20:80 or 30:70 approach (test:train).We have opted for the train:test for 80:20 approach and saved the results for much more precision’s.The final stage of the implementation is to performing our algorithms.

V. Methodology

We have used python language for our two algorithms and we have performed pyspark programming for linear regression algorithm. PySpark Programming. PySpark is the joint effort of Apache Spark and Python. Apache Spark is an open-source cluster computing system, worked around speed, convenience, and gushing examination while Python is a broadly useful, significant level programming language.

There are few libraries that need to be used in our algorithms in python that need to be installed before writing the code, they are ELI5 is a Python package which helps to debug machine learning classifiers and explain their predictions. It provides support for the following machine learning frameworks and packages: scikit-learn and from pyspark libraries.'Missingno' is the missing data visualization packages for python. We have to import 'math' libraries as we need to perform some permutations in our algorithms.

In the methodology we have opted for two algorithms to perform operation on our data-set.They are

*__Linear Regression__

*__K-Nearest Neighbour__

A. Linear Regression

Linear Regression is a machine learning algorithm dependent on supervised learning. It plays out a relapse task. Regression models, objective prediction esteem is dependent on independent variables. It is generally utilized for discovering the connection between variables and determining. Diverse regression models vary dependent on – the sort of connection between the reliant and independent variables, they are thinking about and the number of independent variables being utilized.

Linear regression plays out the errand to predict a dependent variable worth (y) in light of a given independent variable (x). Along these lines, this linear method discovers a regression connection between x (input) and y(output). Subsequently, the name is Linear Regression.[3]

Linear regression algorithm is best for making the predictions and outcomes based on our data-set, in this algorithm, we have used several mathematics calculations like linear algebra, root mean square and permutations. All these calculations help us to and the important features of the data-sets which gives the predictions of player ratings.

we have performed mathematical operations in the linear regression algorithm, the maths operation are (a) Coefficient of determination or R-squared is a factual proportion of how close the information is to the fitted regression line. It is otherwise called the coefficient of determination, or the coefficient of numerous determination for multiple regression.100 percent demonstrates that the model clarifies all the changeability of the reaction data around its mean.

R-squared = Explained variation / Total variation

(b)Root-mean-square deviation or root-mean-square error is an as often as possible utilized proportion of the contrasts between values predicted by a model or an estimator and the qualities watched.

we have performed the permutation importance model and found the top 3 important attributes of the data-set and found the 'Linear Prediction of Player Rating'.

B. K-Nearest Neighbor

The closest neighbor's vote tallies more than that of more distant away, K is a parameter of the calculation. Separation count is utilized in the working of K-Nearest Neighbor. KNN is the calculation that utilizes the entire data-set to foresee and settle on a choice. Speculation of psychological oppressor interpersonal organizations, planning recommend-er framework are the utilization's of the OSN's and the KNN.

The k-nearest neighbors (KNN) algorithm is a basic, simple to-execute supervised ML algorithm that can be utilized to tackle both grouping and regression issues. supervised ml are utilized to take care of arrangements or regression issues.[4]

The KNN is expected that comparative things exist in closeness. As it were, comparative things are close to one another. The KNN is depended on this supposition that being genuine enough for the algorithms to be helpful. KNN catches the possibility of likeness (at times called separation, nearness, or closeness) with some science we may have learned in our adolescence—ascertaining the separation between focuses on a chart.

In the technical report on the knn algorithm we have performed the **(a) K nearest classifier and (b) k nearest regression** and perform the regression curves on the algorithm.we have also performed the **under-fitting and over-fitting** on the training score.Under-fitting implies the model doesn't fit, at the end of the day, it doesn't predict, the training information well overall. Then again, over-fitting implies that the model predicts the training information excessively well. Ordinarily, under-fitting suggests high bias and low variance, and over-fitting infer low bias however high variance.

The data-sets which we have opted for are the biggest game on the planet, so the players are powerful, for performing the substitutes for these players is hard. So, with the use of this algorithm and finding out the essential attributes of the players and the players which can substitute another player.

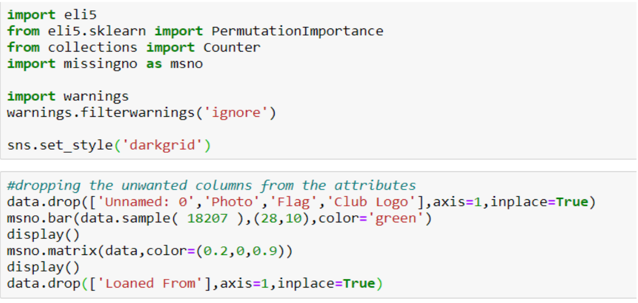
In the technical report we have chosen the knn for finding the substitute player for the football legends like L messi, Christiano Ronaldo and Neymar Jr. For finding the substitute we will choose the k as the main player and by comparing all the attributes which suits the player we will match this attributes with the remaining 18000 rows of data and whoever top 5 players are nearest to the chosen player, it will display those top 5 players.

Performing this operations without KNN is impossible, because Knn performs the nearest neighbor operation full potentially.So for these 3 top legends of the game we have found the substitutes who can replace them whenever a injury or concussion occurs.For the Knn we need a 'standardScaler' library which can match all those players attributes.Knn is the best possible algorithm for the data-set like FIFA-19 where the attributes are unique and data in those attributes are colossal

VI. Software Package Description

The software package consists of linear regression coding and knn coding with data analysis, data cleaning and algorithms modeling.

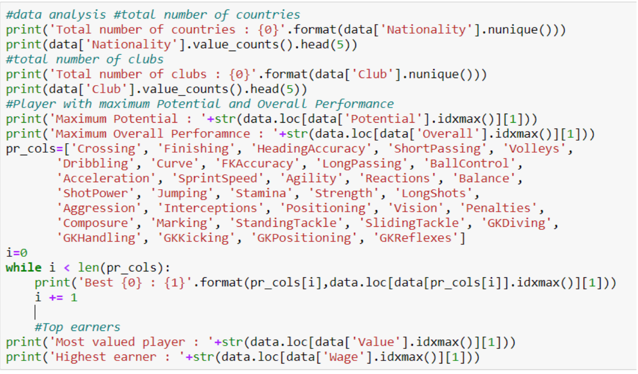
A. Linear regression



```
import eli5
from eli5.sklearn import PermutationImportance
from collections import Counter
import missingno as msno

import warnings
warnings.filterwarnings('ignore')

sns.set_style('darkgrid')
```

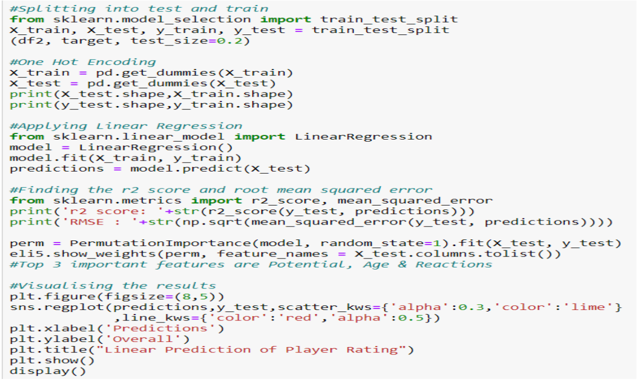
```
#dropping the unwanted columns from the attributes
data.drop(['Unnamed: 0','Photo','Flag','Club Logo'],axis=1,inplace=True)
msno.bar(data.sample( 18207 ),(28,10),color='green')
display()
msno.matrix(data,color=(0.2,0,0.9))
display()
data.drop(['Loaned From'],axis=1,inplace=True)
```

Fig. 1. code 1. Importing librariesdropping attributes in LR



```
#data analysis #total number of countries
print('Total number of countries : {0}'.format(data['Nationality'].nunique()))
print(data['Nationality'].value_counts().head(5))
#total number of clubs
print('Total number of clubs : {0}'.format(data['Club'].nunique()))
print(data['Club'].value_counts().head(5))
#Player with maximum Potential and Overall Performance
print('Maximum Potential : '+str(data.loc[data['Potential'].idxmax()][1]))
print('Maximum Overall Perforamnce : '+str(data.loc[data['Overall'].idxmax()][1]))
pr_cols=['Crossing', 'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys',
       'Dribbling', 'Curve', 'FKAccuracy', 'LongPassing', 'BallControl',
       'Acceleration', 'SprintSpeed', 'Agility', 'Reactions', 'Balance',
       'ShotPower', 'Jumping', 'Stamina', 'Strength', 'LongShots',
       'Aggression', 'Interceptions', 'Positioning', 'Vision', 'Penalties',
       'Composure', 'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving',
       'GKHandling', 'GKKicking', 'GKPositioning', 'GKReflexes']
i=0
while i < len(pr_cols):
    print('Best {0} : {1}'.format(pr_cols[i],data.loc[data[pr_cols[i]].idxmax()][1]))
    i += 1
    |
    #Top earners
print('Most valued player : '+str(data.loc[data['Value'].idxmax()][1]))
print('Highest earner : '+str(data.loc[data['Wage'].idxmax()][1]))
```

Fig. 2. code 2. Data analysis in Linear regression!



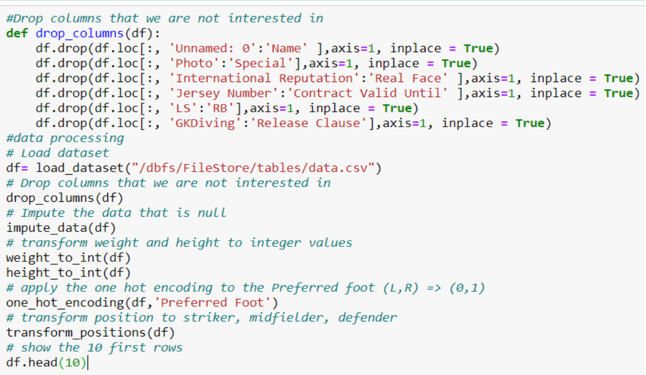
```
#Splitting into test and train
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split
(df2, target, test_size=0.2)

#One Hot Encoding
X_train = pd.get_dummies(X_train)
X_test = pd.get_dummies(X_test)
print(X_test.shape,X_train.shape)
print(y_test.shape,y_train.shape)

#Applying Linear Regression
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)

#Finding the r2 score and root mean squared error
from sklearn.metrics import r2_score, mean_squared_error
print('r2 score: '+str(r2_score(y_test, predictions)))
print('RMSE : '+str(np.sqrt(mean_squared_error(y_test, predictions))))

perm = PermutationImportance(model, random_state=1).fit(X_test, y_test)
eli5.show_weights(perm, feature_names = X_test.columns.tolist())
#Top 3 important features are Potential, Age & Reactions

#Visualising the results
plt.figure(figsize=(8,5))
sns.regplot(predictions,y_test,scatter_kws={'alpha':0.3,'color':'lime'}
            ,line_kws={'color':'red','alpha':0.5})
plt.xlabel('Predictions')
plt.ylabel('Overall')
plt.title("Linear Prediction of Player Rating")
plt.show()
display()
```

Fig. 3. code 3. Linear regression Model

we haven't added all the code regarding the linear regression model, as data cleaning, data analysis and data visualisation has been done in any type of algorithms. All the coding has been done in python and as well as pyspark which will be added with the supplementary files to the technical report. The next coding will be on the knn algorithm

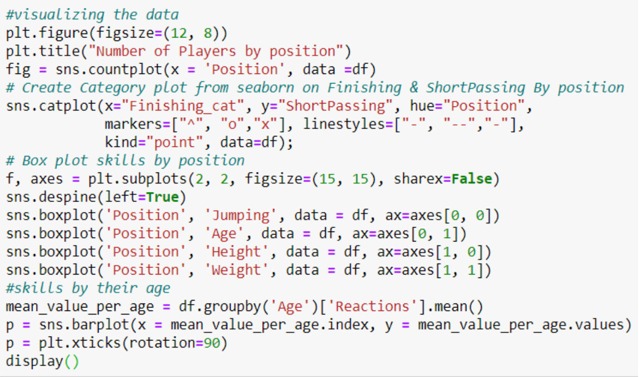
B. K-Nearest Neighbor

we have been opted our second algorithm as knn which we have perform the knn classifier and regression and found out the under-fitting and over-fitting.



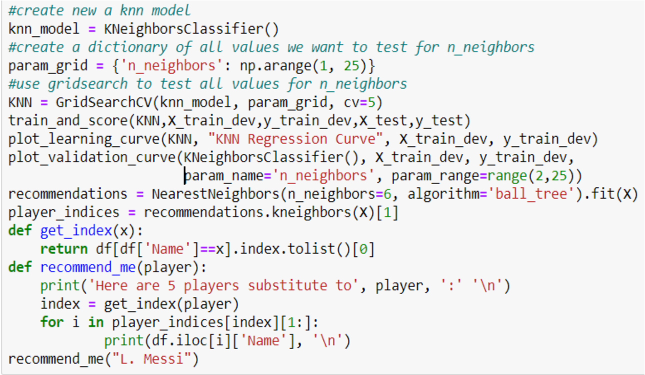
```
#Drop columns that we are not interested in
def drop_columns(df):
    df.drop(df.loc[:, 'Unnamed: 0':'Name' ],axis=1, inplace = True)
    df.drop(df.loc[:, 'Photo':'Special'],axis=1, inplace = True)
    df.drop(df.loc[:, 'International Reputation':'Real Face' ],axis=1, inplace = True)
    df.drop(df.loc[:, 'Jersey Number':'Contract Valid Until' ],axis=1, inplace = True)
    df.drop(df.loc[:, 'LS':'RB'],axis=1, inplace = True)
    df.drop(df.loc[:, 'GKDiving':'Release Clause'],axis=1, inplace = True)
#data processing
# Load dataset
df= load_dataset("/dbfs/FileStore/tables/data.csv")
# Drop columns that we are not interested in
drop_columns(df)
# Impute the data that is null
impute_data(df)
# transform weight and height to integer values
weight_to_int(df)
height_to_int(df)
# apply the one hot encoding to the Preferred foot (L,R) => (0,1)
one_hot_encoding(df,'Preferred Foot')
# transform position to striker, midfielder, defender
transform_positions(df)
# show the 10 first rows
df.head(10)|
```

Fig. 4. code 4. Cleaning data in KNN



```
#visualizing the data
plt.figure(figsize=(12, 8))
plt.title("Number of Players by position")
fig = sns.countplot(x = 'Position', data =df)
# Create Category plot from seaborn on Finishing & ShortPassing By position
sns.catplot(x="Finishing_cat", y="ShortPassing", hue="Position",
            markers=["^", "o","x"], linestyles=["-", "--","-"],
            kind="point", data=df);
# Box plot skills by position
f, axes = plt.subplots(2, 2, figsize=(15, 15), sharex=False)
sns.despine(left=True)
sns.boxplot('Position', 'Jumping', data = df, ax=axes[0, 0])
sns.boxplot('Position', 'Age', data = df, ax=axes[0, 1])
sns.boxplot('Position', 'Height', data = df, ax=axes[1, 0])
sns.boxplot('Position', 'Weight', data = df, ax=axes[1, 1])
#skills by their age
mean_value_per_age = df.groupby('Age')['Reactions'].mean()
p = sns.barplot(x = mean_value_per_age.index, y = mean_value_per_age.values)
p = plt.xticks(rotation=90)
display()
```

Fig. 5. code 4. Cleaning data in KNN



```
#create new a knn model
knn_model = KNeighborsClassifier()
#create a dictionary of all values we want to test for n_neighbors
param_grid = {'n_neighbors': np.arange(1, 25)}
#use gridsearch to test all values for n_neighbors
KNN = GridSearchCV(knn_model, param_grid, cv=5)
train_and_score(KNN,X_train_dev,y_train_dev,X_test,y_test)
plot_learning_curve(KNN, "KNN Regression Curve", X_train_dev, y_train_dev)
plot_validation_curve(KNeighborsClassifier(), X_train_dev, y_train_dev,
                      |param_name='n_neighbors', param_range=range(2,25))
recommendations = NearestNeighbors(n_neighbors=6, algorithm='ball_tree').fit(X)
player_indices = recommendations.kneighbors(X)[1]
def get_index(x):
    return df[df['Name']==x].index.tolist()[0]
def recommend_me(player):
    print('Here are 5 players substitute to', player, ':' '\n')
    index = get_index(player)
    for i in player_indices[index][1:]:
            print(df.iloc[i]['Name'], '\n')
recommend_me("L. Messi")
```

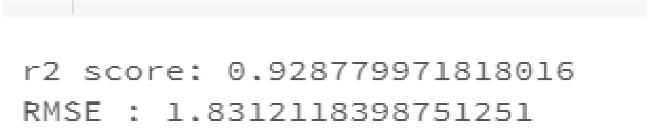
Code 6. Knn modeling and prediction

All the knn algorithm code haven't been added in the technical report as many commands and libraries are universal which can be implemented anywhere in any type of algorithms. Only the most important aspects of the linear regression and k-Nearest Neighbor algorithms have been show in the software package. And importing the sklearn libraries are very useful and importing 'pyspark' libraries are little complicated as pyspark doesnt have all the libraries for machine learning algorithms so some of them need to be directly imported from the scikit learn packages.

VII. Experimental Results

As we have taken FIFA-19 data-set we have performed all the data cleaning, data analysis and data visualization at the end we have pushed the data into the algorithms by creating models and performed the testing and training the data and with the accuracy values we are performing the predictions onto our data-set.

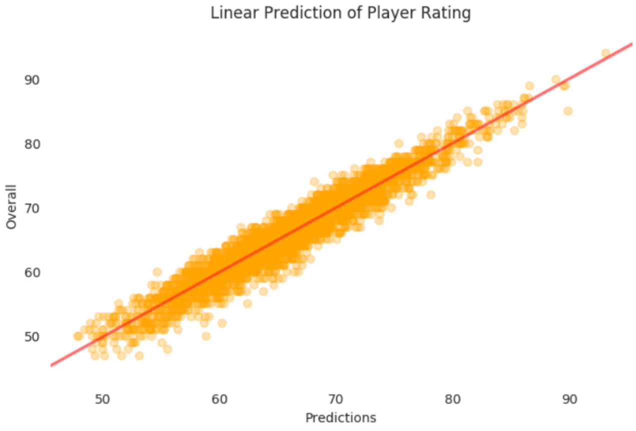
* **Linear regression:** After performing the mathematical calculations on the linear regression model we got the r2 score and root mean squared error rate as:



```
r2 score: 0.928779971818016
RMSE : 1.8312118398751251
```

Result 1: r2 and rmse score of lr

The r2 score of 0.92 is very good for predictions and the mean error of 1.8312 is also very good for further predictions.



Result 2: Prediction of player ratings

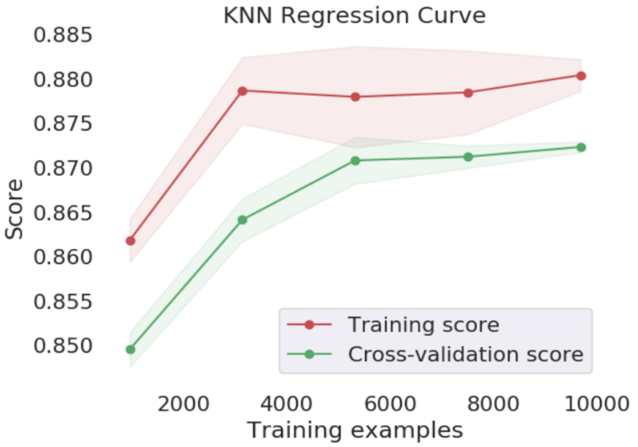
so we have performed the predictions of players based upon the ratings.All these prediction are performed based upon the potential permutation on all the attributes, by knowing the top 3 attributes of the data-set we have performed the prediction and find out the linear prediction of player ratings.

By performing the permutation importance on the attributes we found the top 3 attributes, those attributes are (a) Potential (b) Age (c) Reactions.

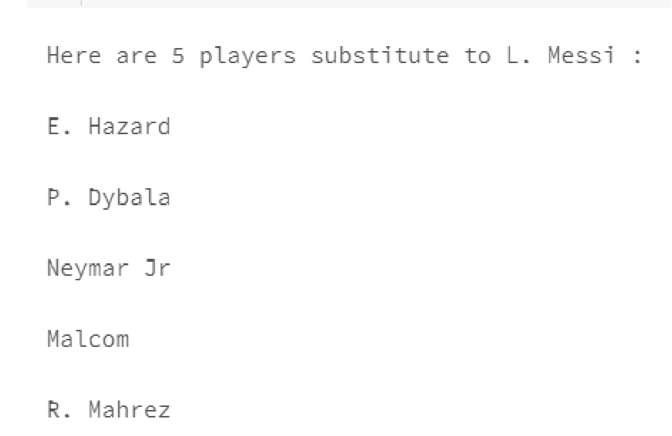
* **K-Nearest Neighbor:**

On the K-Nearest neighbor we have found the classification and regression on the knn models and found the training score and cross validation score of more that 0.88 which is 88 percent accuracy of the knn algorithm on the data-set so further we can perform the predictions on to the algorithm and find out the optimal results.

After finding out the accuracy we have performed the operation on algorithm for knowing the top 5 substitute players for the particular player ’L.messi’ as he is the football legend finding the substitute for him is most complicated, so only knn algorithm can perform these operations, where we can find the nearest neighbors based upon the clusters.We have also find the substitutes for the ’christiano ronaldo’ and ’Neymar jr’. By getting the good amount of under-fitting and over-fitting score we can predict the solutions with higher accuracy.



Result 3: Training,cross-validation score on regression curve



```
Here are 5 players substitute to L. Messi :

E. Hazard

P. Dybala

Neymar Jr

Malcom

R. Mahrez
```

Result 4: Top 5 Substitute players for L.Messi

comparing the two different algorithms is quite injustice, but we will analyze their accuracy scores, linear regression got the 92.8 percent accuracy and knn got the 88 percent accuracy so performing predictions on linear regression is much more effective.

VIII. Conclusion

Performing the operations on FIFA-19 Data-set is a crucial task, as missing a single attribute from the huge data-set will impact the accuracy and predictions of the algorithms.So performing all the data cleaning, analysis and visualisation was the huge task to perform professionally as well as changing the attributes data into a single integer type is important as attributes with different value types wont perform any result. We have opted for databricks, the cloud based platform for running the machine learning algorithms was a good option as we don’t need any other tools to install, which can be accessed from anywhere in the world. But the cluster creation is very awful as two hours of inactivity will terminate the cluster and their are much more restrictions in the cloud based platform. Doing coding in python and pyspark is not easy task as their are much more complications in the form of dataframes and libraries. The accuracy of both algorithms are more than 88 percent which are pretty good for performing any predictions and operations on our data-set. But while comparing both the algorithms the linear regression have 92 percent accuracy which is higher than knn’s 88 percent.

References

[1] Nassis, G.P. (2013) ‘Effect of Altitude on Football Performance: Analysis of the 2010 FIFA World Cup Data’, The Journal of Strength Conditioning Research, 27(3), 703–707.

[2] Williams, J.H. (2010) ‘Relative age effect in youth soccer: analysis of the FIFA U17 World Cup competition’, Scandinavian Journal of Medicine Science in Sports, 20(3), 502–508.

[3] When to Use Linear Regression, Clustering, or Decision Trees - DZone AI [online] (2020) dzone.com, available: https://dzone.com/articles/decision-trees-vs-clustering-algorithms-vs-linear

[4] Harrison, O. (2019) Machine Learning Basics with the K-Nearest Neighbors Algorithm [online], Medium, available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 [accessed 23 Feb 2020].

[5] Prado Prandini Faria, M., Maria Silva Julia, R., Bononi Paiva Tomaz, L. (2019) ‘Evaluating the Performance of the Deep Active Imitation Learning Algorithm in the Dynamic Environment of FIFA Player Agents’, in 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Presented at the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 228–233.