STATISTICS FOR DATA SCIENCE

UE19CS203

3rd Semester, Academic Year 2020-21

PROJECT TITLE:

FIFA 20 PLAYER DATASET

PROJECT MEMBERS:

(SECTION -D)

1. KK TARUN KUMAR – PES1UG19CS222

2. KAUSTUBH SHARMA - PES1UG19CS215

3. KOUSHIK VARMA MANDAPATI – PES1UG19CS230

4. MALHAR CHANDRAKANT PATTEKAR – PES1UG19CS255

TABLE OF CONTENTS:

CODE:

```
# -*- coding: utf-8 -*-
"""sds_final.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1m9_9oyaJFMbZarRPAY3r3QUMvhMrArhC
"""

import pandas as pd
import numpy as np
import random as rand
import matplotlib.pyplot as plt
import sklearn
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
import seaborn as sns
from scipy import stats
from random import sample
import statistics

data=pd.read_excel("save4.xlsx",na_values= ["NotAvailable", "na", "-"])

#removing unwanted columns
data=data.drop(['value_eur'],axis=1)

# reseting index column of dataset
data.reset_index(drop=True)
```

```
#removing rows with more than 6 null values
null_new=[]
nullRows = data.isnull().sum(axis=1)
count = 0
for i in range(0,len(nullRows)):
   if(nullRows[i]>6):
      null_new.append(i)
data=data.drop(data.index[null_new])

data.head(20)

# reseting index column of dataset
data.reset_index(drop=True,inplace=True)

#filling null values with the mode of that particular column and
#covert ing everything to upper case
fillna_cols_cate=['preferred_foot','work_rate','body_type','team_position']
for i in fillna_cols_cate:
  data[i]=data[i].str.upper()
  data[i].fillna(data[i].mode()[0], inplace=True)

#filling null values with the mode of that particular column
fillna_cols_num=['skill_moves','weak_foot']
for i in fillna_cols_num:
  data[i].fillna(data[i].mode()[0], inplace=True)

data['wage_eur']=data['wage_eur'].replace(0,np.nan)

data['skill_moves'].sum()
```

```
#sorting columns to find mean and meadian
num_cols_mean=['age','overall','shooting','physic','defending']
num_cols_median=['height_cm','weight_kg','potential','wage_eur','pace','passing','dribbling
']

#calculating mean and median to fill the null values of numeric columns
for i in num_cols_mean:
  temp=data[i].mean()
  data[i].fillna(int(temp), inplace=True)
for i in num_cols_median:
  temp=data[i].median()
  data[i].fillna(int(temp), inplace=True)

#filling remaining null values with 'UNKNOWN'
data.fillna('UNKNOWN',inplace=True)

#checkpoint saved
data.to_excel('save3.xlsx',index=False)

#calculating number of players in a position and making a
#bar graph against 'defending'
unique_pos=data.team_position.unique()
final_position={}
defend_pos=[]
for pos in unique_pos:
  final_position[pos]=data.loc[data['team_position'] == pos]
for df in final_position.values():
  defend_pos.append(df['defending'].mean())
plt.xticks(rotation='vertical')
```

```
plt.bar(unique_pos,defend_pos)
plt.xlabel('team_position')
plt.ylabel('defending')

#calculating number of players in a position and making a bar graph against 'shooting'

unique_pos=data.team_position.unique()
final_position={}
shoot_pos=[]
for pos in unique_pos:
  final_position[pos]=data.loc[data['team_position'] == pos]
for df in final_position.values():
  shoot_pos.append(df['shooting'].mean())
plt.xticks(rotation='vertical')
plt.bar(unique_pos,shoot_pos)
plt.xlabel('team_position')
plt.ylabel('shooting')

#calculating and ploting a piechart to know the percentage of people with different skill move ratings
unique_skill=data.skill_moves.unique()
final_skill={}
shoot_pos=[]
for skill in unique_skill:
  final_skill[skill]=data.loc[data['skill_moves'] == skill]
for df in final_skill.values():
  shoot_pos.append(df.shape[0])
explode = (0, 0, 0, 0.3, 0)
plt.pie(shoot_pos,labels=unique_skill,autopct='%1.1f%%',shadow=True,explode=explode)
plt.title('skill moves')
```

```
#plotting a scatter plot between overall rating of a

#player and his wage before removing outliers

plt.scatter(data['overall'],data['wage_eur'])

plt.xlabel('overall')

plt.ylabel('wage_eur')


data.boxplot(column='passing')


#checking for outliers in categorical data

plt.bar(data.groupby(['body_type']).count().index,data.groupby(['body_type']).count().short
_name)


#replacing values, as "MESSI" is a outlier

data['body_type']=data['body_type'].replace('MESSI','NORMAL')


#calculating and changing values of the outliers of numeric type

num_col_outlier=['overall','potential','pace','shooting','passing','dribbling','defending','physi
c']

for i in num_col_outlier:

  q3=np.percentile(data[i], 75, interpolation = 'midpoint')

  q1=np.percentile(data[i], 25, interpolation = 'midpoint')

  IQR=q3-q1


  median=np.percentile(data[i], 50, interpolation = 'midpoint')

  data[i][data[i]>(q3+(IQR*1.5))]=q3+(IQR*1.5)

  data[i][data[i]<(q1-(IQR*1.5))]=q1-(IQR*1.5)


data.boxplot(column='passing')


#redoing the previous graphs after removing outliers
```

```
unique_pos=data.team_position.unique()
final_position={}
defend_pos=[]
for pos in unique_pos:
  final_position[pos]=data.loc[data['team_position'] == pos]
for df in final_position.values():
  defend_pos.append(df['defending'].mean())
plt.xticks(rotation='vertical')
plt.bar(unique_pos,defend_pos)
plt.xlabel('team_position')
plt.ylabel('defending')

#redoing the previous graphs after removing outliers
unique_pos=data.team_position.unique()
final_position={}
defend_pos=[]
for pos in unique_pos:
  final_position[pos]=data.loc[data['team_position'] == pos]
for df in final_position.values():
  defend_pos.append(df['shooting'].mean())
plt.xticks(rotation='vertical')
plt.bar(unique_pos,defend_pos)
plt.xlabel('team_position')
plt.ylabel('shooting')

unique_skill=data.skill_moves.unique()
final_skill={}
shoot_pos=[]
for skill in unique_skill:
  final_skill[skill]=data.loc[data['skill_moves'] == skill]
```

```
for df in final_skill.values():
  shoot_pos.append(df.shape[0])
explode = (0, 0, 0, 0.3, 0)
plt.pie(shoot_pos,labels=unique_skill,autopct='%1.1f%%',shadow=True,explode=explode)
plt.title('skill moves')

plt.scatter(data['overall'],data['wage_eur'])

#checkpoint saved
data.to_excel('save4.xlsx',index=False)

hdf = data['potential'].plot(kind = 'kde', stacked = False)
plt.show()

#normalizing data
norm_cols=['age', 'height_cm', 'weight_kg', 'overall', 'potential', 'wage_eur','weak_foot',
        'skill_moves', 'pace', 'shooting', 'passing', 'dribbling', 'defending', 'physic']
data[norm_cols] = StandardScaler().fit_transform(data[norm_cols])

hdf = data['potential'].plot(kind = 'kde', stacked = False)
plt.show()

data.to_excel('save_norm.xlsx',index=False)

data=pd.read_excel("save4.xlsx",na_values= ["NotAvailable", "na", "-"])

#hypothesis testing
data_sample=sample(list(data['skill_moves']),270)

mean=sum(data_sample)/len(data_sample)
```

```
sd=data['skill_moves'].std()
var=statistics.variance(data['skill_moves'])
#h0:mean=<2.52
#h1:mean>2.52
#step 1
#null hypothsis=(mean,std/sqrt(n))
data_zscore=stats.zscore(data['skill_moves'])

from scipy.stats import ttest_1samp
_,pvalue=ttest_1samp(data_sample,2.52)

var

mean

pvalue

data_zscore

# p_value>0.05(standard value), hence we fail to reject h0
# this implies h1 is also possible
# hence players with skill_move>2.4 have potential>71.46

#finding the corellation matrix, and plotting a heat map for better vizualization
corelation=data.corr()
print(corelation)
mask = np.zeros_like(corelation, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(corelation, mask=mask, cmap="GnBu", vmax=.3, center=0,
            square=True, linewidths=.7, cbar_kws={"shrink": .7})
```

ABSTRACT:

The Dataset Being studied is of the Player Data in FIFA 2020. It allows multiple comparisons of players and their attributes. Each player has a distinct set of attributes based on their real-life performances. The Following Data has been Processed and cleaned by either Dropping or Imputing the Unknown values by taking Mode, Median or Mean. Multiple graphs have been developed to understand the dataset better and find relations between various Attributes. Then we have analysed the dataset and have formed a Hypothesis by taking H_0 as:**" Players have an average skill move rating lesser than or equal to 2.52".**

INTRODUCTION:

FIFA 20 is a Football Simulation Video Game published by Electronic Arts (EA) as a part of a long running FIFA series. It is the 27th instalment in the FIFA series and was released on the 27th of September 2019. FIFA 20 is one of EA's biggest games and highest earner.

It includes real life football players representing their Football clubs and Nations. These players have been scouted for multiple years and have been given attribute ratings based off the scout's notes. This game is a massive hit amongst football fans all over the world as it gives them a chance to control and play their favourite teams and players and indulge in the world of Football.

We have tried to exploit the Power of Data Science to attain a deeper understanding of how the players are rated and what are the attributes which play massive roles in a players overall rating.

Problem Statement: Analysis of FIFA 20 players and their attributes

DATASET INFORMATION:

The Following dataset has been uploaded on Kaggle by Stefano Leone and includes the following attributes:

- sofifa_id: It is a unique identification number assigned to every player
- short_name: It denotes the name of the player
- age: It denotes the age of the player
- height_cm: It denotes the height of the player in centimetres
- weight_kg: It denotes the weight of the player in kilograms
- nationality: It denotes the Nationality of the player
- club: It denotes the club a player is associated to
- overall: It is a rating given to each player based on all other numerical attributes
- potential: It is highest overall rating a player can achieve as and when he grows in attributes (in game)
- wage_eur: It denotes players income in euros
- preferred_foot: It denotes whether a player is left footed or right footed
- weak_foot: It is a rating out of 5 which denotes how good a players weak foot is
- skill_moves: It is a rating out of 5 which denotes how good a player is at executing skills
- work_rate: It denotes a player's involvement in attacking an defending scenarios
- body_type: It denotes the type of in-game body a player has
- team_position: It denotes the position a player plays a majority of his matches
- team_jersey_number: It depicts a players jersey number for his specific team
- pace: It denotes how fast a player is
- shooting: It denotes how good a player is while taking a shot at goal
- passing: It denotes how good a player is while making a pass
- dribbling: It denotes how good a player is at dribbling
- defending: It denotes how good a player is at defending
- physic: It denotes a players strength

PREPROCESSING OR DATA CLEANING:

Data Cleaning is the process of detecting and correcting, inaccurate records from a record set, table or database. This also refers to identifying incomplete, irrelevant parts of the data by following certain methods. There are multiple effective ways to clean a particular data set. Few of them are as follows:

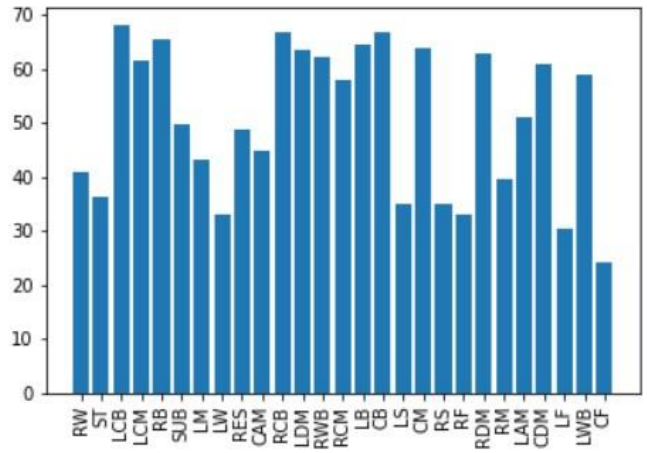
- Imputing: This is the process of replacing missing data with substituted values.
- Modifying: Modification of Incorrect Data from the data set
- Deletion: The process of deleting/dropping of irrelevant or unwanted data from the data set

We have systematically gone through our Dataset and have replaced missing data with the mean or the mode of that particular attribute. We have dropped 8 columns because the contained unnecessary and irrelevant information. Like-wise we have also dropped the players who had 6 or more than 6 missing values in their attribute. We also capitalized the string values to prevent any incorrect capitalization.

EXPLORATORY DATA ANALYSIS:

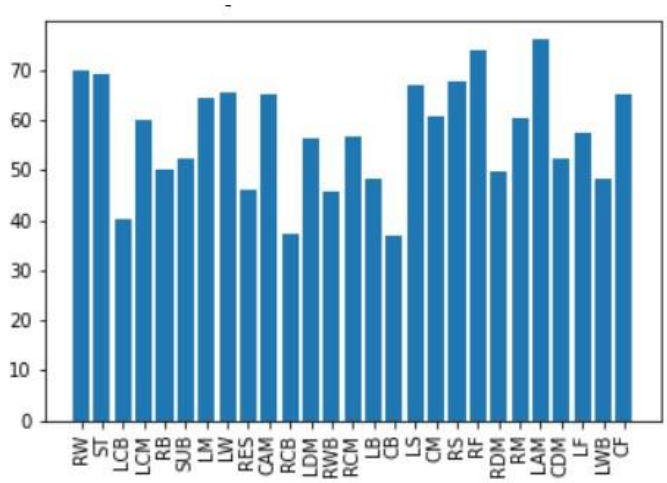
Data Visualization:

BAR GRAPH: POSITION VS DEFENDING RATING



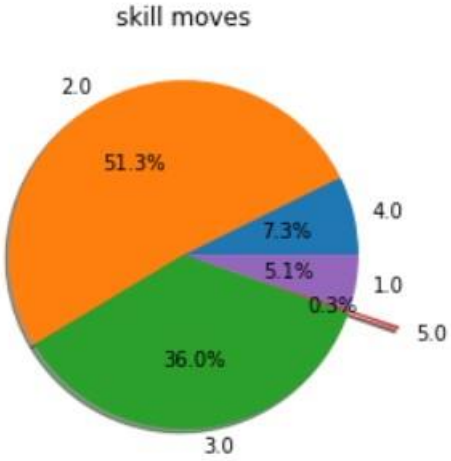
Inference: As seen in the graph players who have defensive positioning (LB, RB…) have higher defending rating than the forward players (RW, ST…)

BAR GRAPH: POSITION VS SHOOTING RATING



Inference: As seen in the graph players who have attacking positioning (LW,ST,RW…) have higher shooting rating than the defensive players.

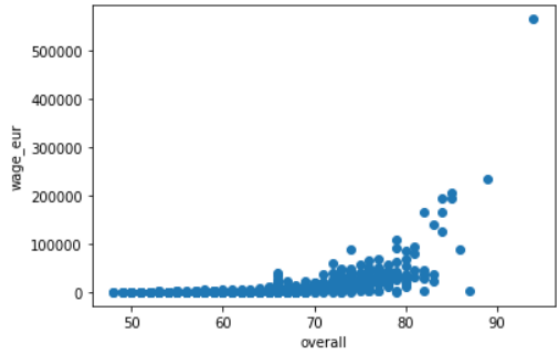
PIE CHART: DISTIBUTION OF SKILL RATINGS



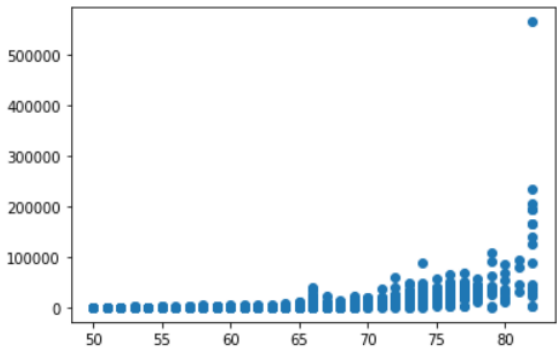
Inference: As seen in the pie chart 2 skill rating is most common amongst players, with 51.3% of the players having 2-star skills while the 5-skill rating is the least common amongst players with only 0.3% of the players having 5-star skills.

SCATTER PLOT: WAGE_EUR VS OVERALL RATING

Before Outlier Removal:



After Outlier Removal:



Inference: As seen in the above scatter plot it is inferred that players with higher overall rating (Better playing in real life) get paid higher wages.

BOX PLOT: PASSING

IMAGE 1: BEFORE REMOVAL OF OUTLIERS

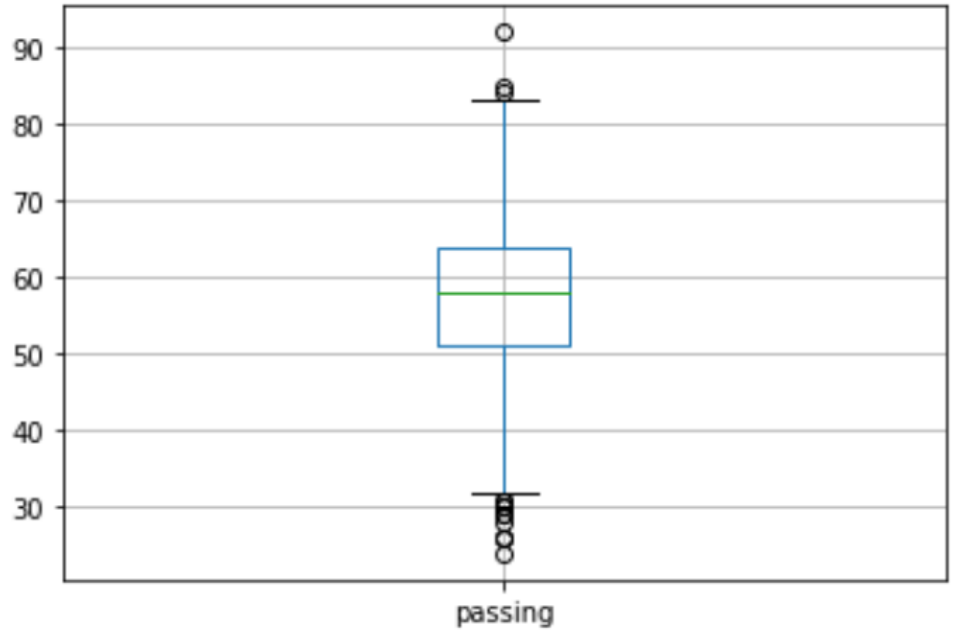
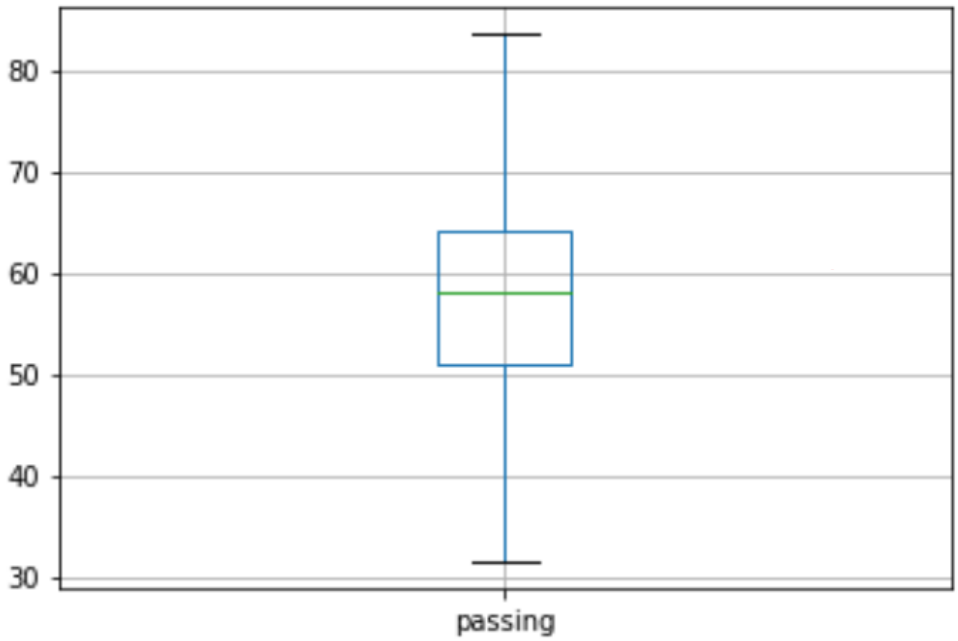


IMAGE 2: AFTER REMOVAL OF OUTLIERS



Inference: As seen from the above boxplot, we infer the passing ratings of the players. From the Box Plot it is inferred that the median of the passing rating lies between the interval 50 and 60. This Box Plot also contains outliers which can be removed using conventional methods. An Outlier is an observation that lies an abnormal distance from other values in a random sample from a population. They are a set of results which are extremely bigger or smaller than the nearest data point.

Outlier Treatment: We use the Inter Quartile Range concept to identify the outliers in the dataset and eradicate the outliers by decreasing them.

Normalization and Standardization:

We normalized the data using the fit transform function, which is imported from sklearn.preprocessing, such that the mean and the variance are close to 0 and 1 respectively.

HYPOTHESIS TESTING:

From the selected dataset, the improvement of a player's potential is heavily dependent on their skill moves. Hence, it was only appropriate that our research was based on the improvement of a player's potential.

STEP 1:

Our Research Hypothesis is:

H_1: Conclude that Players with skill move > 2.52.

The contradictory Null Hypothesis is:

H_0: Players have an average skill move rating lesser than or equal to 2.52.

Here a Sample of 270 players is taken from a population of 1080 players randomly using the random module. The sample mean of the attribute "skill_move" has been considered and concluded that it is 2.52. Standard deviation and variance of the attribute "skill_move" is also calculated using the functions from in-built modules.

STEP 2:

Assume Null Hypothesis is True.

STEP 3:

Null distribution:

n = 270, Mean = 2.52, variance=0.488, standard deviation= 0.698

$\bar{x}$~N (mean, variance/n) = X'~N (2.52, 0.0.488)

$\bar{x}$~N (mean, standard deviation/sqrt(n)) = X'~N (2.52, 0.698)

STEP 4:

Obtaining of z-score:

The z-score is obtained from the following equation $\frac{\bar{x}-mean}{standard\ deviation}$. Using the calculated z-score we get the p-value. Obtained z-score = 0.52

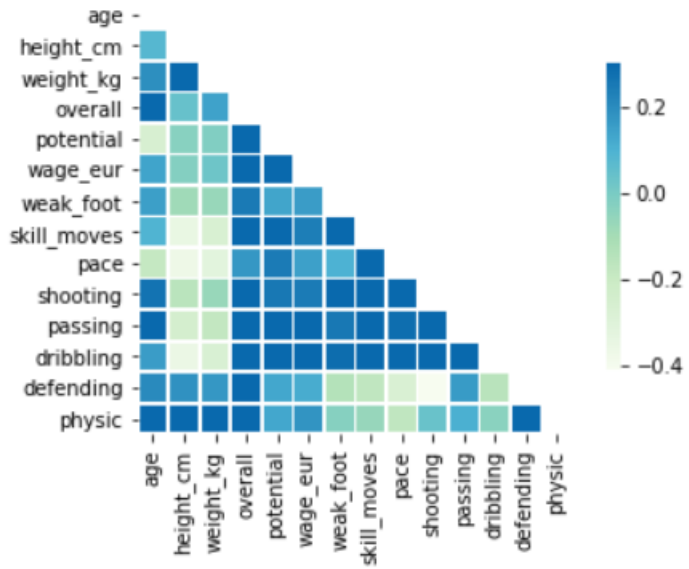
Obtaining of p-value:

The P-value is obtained by using a function from the inbuilt module known as scipy.stats. We use the p-value of the sample data set and compare it with the significance level(alpha) which has a standard value 0.05. Obtained p-value = 0.699

STEP-5:

Comparing the p-value with the significance level we realize that p-value is greater than the standard value of the significance level. (0.699>0.05). This implies that we failed to reject H_0. If we fail to reject H_0, H_1 is also plausible. Hence, we conclude that both H_0 and H_1 are plausible and this case the sample has 69.9% of probability disagreeing with H_0.

HEAT MAP: CORRELATION BETWEEN ALL THE ATTRIBUTES



Inference: As seen from the above Heat map, we extrapolate from the fact that darker a particular region of the heat map more is the relation among the corresponding attribute. For example, a player with more height will probably have more weight and thus is represented by a darker region.

Correlation Co-efficients:

	age	height_cm	weight_kg	...	dribbling	defending	physic
age	1.000000	0.083007	0.192875	...	0.160876	0.208325	0.391349
height_cm	0.083007	1.000000	0.742966	...	-0.350384	0.188213	0.472462
weight_kg	0.192875	0.742966	1.000000	...	-0.266508	0.170330	0.528792
overall	0.440207	0.044894	0.142431	...	0.581951	0.308700	0.488236
potential	-0.259270	-0.030886	-0.013102	...	0.439255	0.130162	0.131661
wage_eur	0.141377	-0.020741	0.029073	...	0.332628	0.117733	0.182933
weak_foot	0.154585	-0.086341	-0.067424	...	0.290125	-0.134275	-0.025814
skill_moves	0.096911	-0.346286	-0.263909	...	0.619670	-0.168094	-0.060448
pace	-0.182417	-0.366217	-0.312708	...	0.541277	-0.267100	-0.167898
shooting	0.270051	-0.151496	-0.065871	...	0.712482	-0.407740	0.038773
passing	0.316667	-0.243857	-0.177503	...	0.791591	0.164152	0.111260
dribbling	0.160876	-0.350384	-0.266508	...	1.000000	-0.149310	-0.034509
defending	0.208325	0.188213	0.170330	...	-0.149310	1.000000	0.476849
physic	0.391349	0.472462	0.528792	...	-0.034509	0.476849	1.000000

RESULTS AND CONCLUSION:

This Project is centred on the subject Data science and we have Analysed a data set which is based on a Football simulation video game known as FIFA 20. The main motive behind performing the above Hypothesis is to come to a conclusion on how much a player's potential is affected by their skill move rating. The aforementioned hypothesis is performed to conclude the fact that players with skill move greater than 2.52 which implies that players skill rating plays a big role in their potential. We have used multiple Modules and functions such as pandas, numpy, random, matplotlib.pyplot, etc. to analyse the data. We have also used the process of data visualization for viewing the data through an easier perceptive and efficiently interpret the data. The different types of graphs used here are Boxplots, Bar graphs, Heat Map, Scatter plot. These graphs allow us to interpret various attributes of a particular player. In conclusion, we believe that the project was very insightful in representing how players ratings are calculated in game.