**Project 2 – ML- Predicting players rating**

In this project you are going to predict the overall rating of soccer player based on their attributes

such as 'crossing', 'finishing etc.

The dataset you are going to use is from European Soccer Database

(https://www.kaggle.com/hugomathien/soccer) has more than 25,000 matches and more than

10,000 players for European professional soccer seasons from 2008 to 2016.

Download the data in the same folder and run the following command to get it in the environment

About the Dataset

The ultimate Soccer database for data analysis and machine learning

The dataset comes in the form of an SQL database and contains statistics of about 25,000 football

matches, from the top football league of 11 European Countries. It covers seasons from 2008 to

2016 and contains match statistics (i.e: scores, corners, fouls etc...) as well as the team formations,

with player names and a pair of coordinates to indicate their position on the pitch.

+25,000 matches

+10,000 players

11 European Countries with their lead championship

Seasons 2008 to 2016

Players and Teams' attributes\* sourced from EA Sports' FIFA video game series, including the

weekly updates

Team line up with squad formation (X, Y coordinates)

Betting odds from up to 10 providers

Detailed match events (goal types, possession, corner, cross, fouls, cards etc...) for +10,000

matches

The dataset also has a set of about 35 statistics for each player, derived from EA Sports' FIFA video

games. It is not just the stats that come with a new version of the game but also the weekly

updates. So for instance if a player has performed poorly over a period of time and his stats get

impacted in FIFA, you would normally see the same in the dataset.

=====================================================================================

**Code:**

#Import Libraries

import sqlite3

import pandas as pd

from sklearn.tree import DecisionTreeRegressor

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

from math import sqrt

import seaborn as sns

# Create connection.

cnx = sqlite3.connect('C:/Users/srinivas.gannavaram/Downloads/soccer/database.sqlite')

df = pd.read\_sql\_query("SELECT \* FROM Player\_Attributes", cnx)

df.head(2)

cursor = cnx.cursor()

cursor.execute("SELECT name FROM sqlite\_master WHERE type='table';")

print(cursor.fetchall()

df.shape

df.keys()

df.columns

df.describe()

# Null check

df.isnull().sum(axis=0)

df = df.dropna()

# Test for Null after removing

df.isnull().sum(axis=0)

df['overall\_rating'] = df['overall\_rating'].astype('int')

df.corr(method='pearson',min\_periods=1).transpose().sort\_values('overall\_rating', ascending=False)

# Overall correlation using pearson

df['overall\_rating'] = df['overall\_rating'].astype('int')

df.corr(method='pearson',min\_periods=1).transpose().sort\_values('overall\_rating', ascending=False)

df['overall\_rating'] = df['overall\_rating'].astype('int')

display(df.sort\_values('overall\_rating', ascending=False).head(10)[['overall\_rating', 'reactions', 'potential', 'short\_passing', 'long\_passing','ball\_control', 'vision', 'shot\_power']])

feature\_columns = ['reactions', 'potential']

X = df[feature\_columns]

y = df.overall\_rating

lm = LinearRegression()

lm.fit(X,y)

print(lm.intercept\_)

print(lm.coef\_)

from matplotlib.pyplot import figure

import matplotlib.pyplot as plt

#Draw graph for Relationship Between Player Reaction and Overall Rating

figure(num=None, figsize=(12,10), facecolor='G', edgecolor='G')

plt.scatter(df.reactions, df.overall\_rating)

plt.xlabel('Player Reaction')

plt.ylabel('Player Overall Ratings')

plt.title("Relationship Between Player Reaction and Overall Rating")

plt.show()

import statsmodels.formula.api as smf

lm = smf.ols(formula='overall\_rating ~ reactions + potential', data=df).fit()

lm.conf\_int()

lm.summary()

feature\_columns = ['overall\_rating', 'reactions', 'potential', 'short\_passing', 'long\_passing','ball\_control', 'vision', 'shot\_power']

X = df[feature\_columns]

y = df.overall\_rating

lm = LinearRegression()

lm.fit(X,y)

print(lm.intercept\_)

print(lm.coef\_)

lm = smf.ols(formula='overall\_rating ~ reactions + potential + short\_passing + long\_passing + ball\_control + vision + shot\_power', data=df).fit()

#display(lm.rsquared)

#display(lm.resid)

#display(lm.fvalue)

#display(lm.params.Intercept)

#display(lm.params.lstat)

lm.conf\_int()

lm.summary()

=====================================================================================

**Output:**





















