­­­­­­­­­

**Rashtriya Raksha University**

**School of Information Technology, Artificial Intelligence and Cyber Security (SITAICS)**

**B. Tech CSE with specialization in Cyber Security**

**5th Semester**

**Project Report**

**Subject:** Design Project

**Submitted by:**

**Name:** Harish Agarwal

**Enrolment no.-** 200031101611025

**FIFA QATAR PREDICTION**

**Introduction:**

The FIFA World Cup is an international soccer tournament held every four years. The 2022 FIFA World Cup is scheduled to take place in Qatar from November 21 to December 18, 2022. This will be the first time that the World Cup will be held in the Middle East, and the first time that it will take place in November and December, rather than June and July. A total of 32 teams will participate in the tournament.

**Objectives:**

* The main objective is to predict the winner of 2022 FIFA world cup with the help of data of all the previous matches played till now.
* The second objective is to create a sustainable or reliable algorithm to predict the result of any tournament in any sport with its training data set.

**Data Description:**

The data contains the different data related to a previous FIFA world cup. The detailed data dictionary is given below.

**Data Dictionary:**

**date:** The actual date when the match was played.

**home\_team:** home team is the team which played in its own country.

**away\_team:** away team is the team which played matches in any country except its country.

**home\_score:** Score made by home team.

**away\_score:** score made by away team

**tournament:** There are different types of tournament like FIFA, friendly, British tournament etc.

**city:** The city in which the match was played.

**country:** Country in which the match was played.

**neutral:** Result of the match.

**Libraries used in Project:**

**Pandas:**

Pandas is a popular open-source data analysis and data manipulation library for Python. It provides data structures for efficiently storing large datasets and tools for working with them. The two main data structures in Pandas are the Series (1-dimensional) and the DataFrame (2-dimensional). Pandas makes it easy to perform operations such as aggregation, filtering, and transformation on large datasets.

**Numpy:**

NumPy is a library for the Python programming language that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It is a fundamental package for scientific computing with Python, and is widely used in machine learning, data science, and scientific computing communities. It provides high performance multidimensional array objects, which allow users to perform mathematical operations on these arrays with ease, including linear algebra, random number generation, and Fourier transform.

**Seaborn:**

Seaborn is a data visualization library for Python, based on Matplotlib. It provides a high-level interface for creating statistical graphics and offers a more concise and aesthetically pleasing syntax for producing charts, histograms, scatter plots, etc. Seaborn also has built-in support for visualizing statistical models and includes functions for estimating and visualizing linear regression models, generalized linear models, and categorical models.

**Matplotlib:**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. Matplotlib provides a high-level interface for producing static, animated, and interactive visualizations in Python. It has a variety of tools for plotting and visualizing data, including line plots, scatter plots, bar plots, error bars, histograms, bar charts, pie charts, box plots, and more.

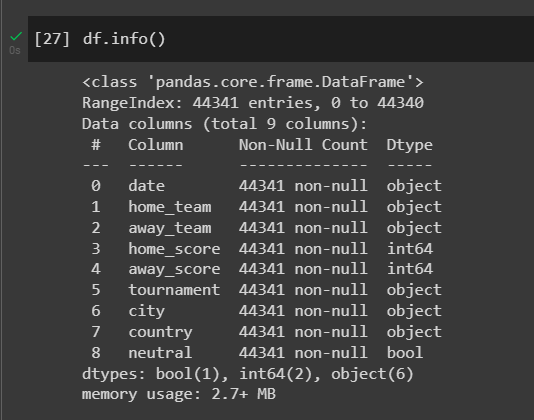
**Random:**

The random library is a module in Python's standard library that implements pseudo-random number generators for various uses. It provides a suite of functions for generating random numbers from various continuous and discrete distributions, including uniform, Gaussian, exponential, Poisson, and others. The library also provides functions for generating random numbers from a seed, shuffling sequences randomly, and picking random elements from a sequence. The random library is useful for a wide range of applications, including simulations, games, cryptography, and more.

**Dataset Image:**

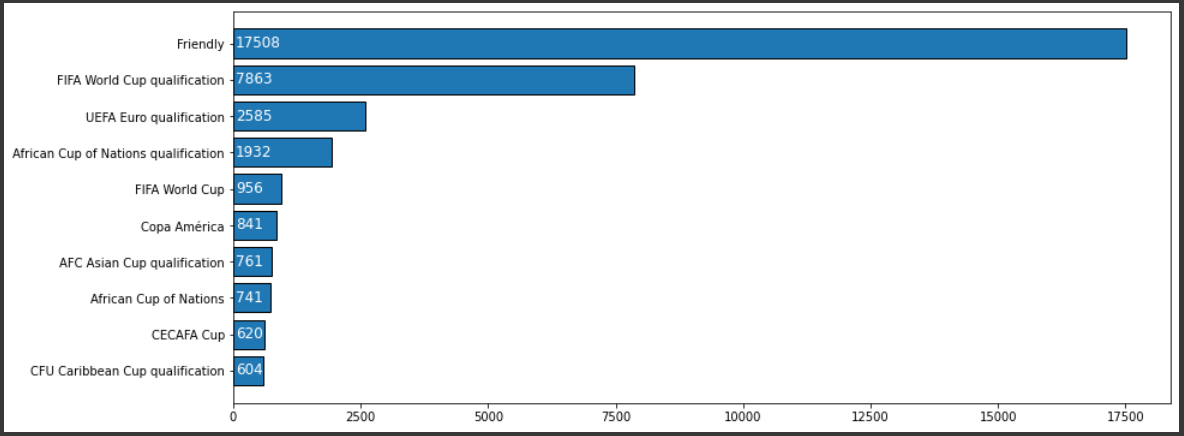
****

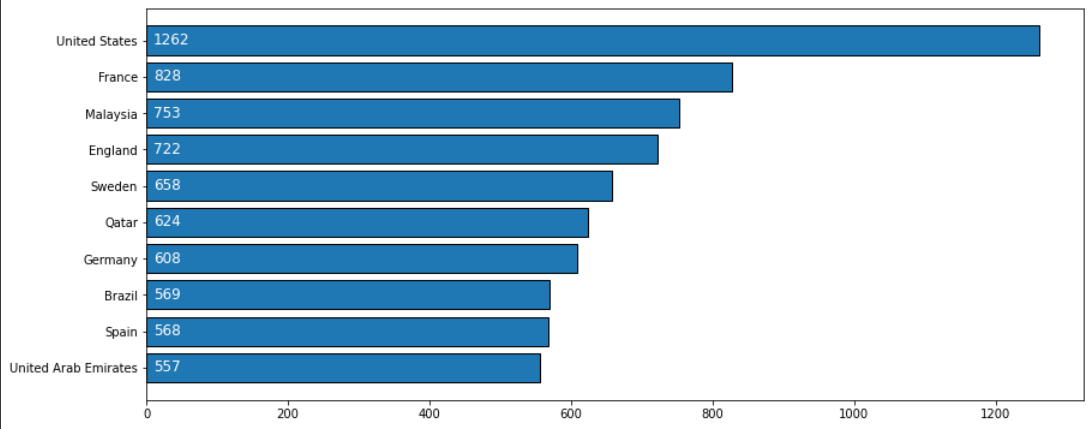
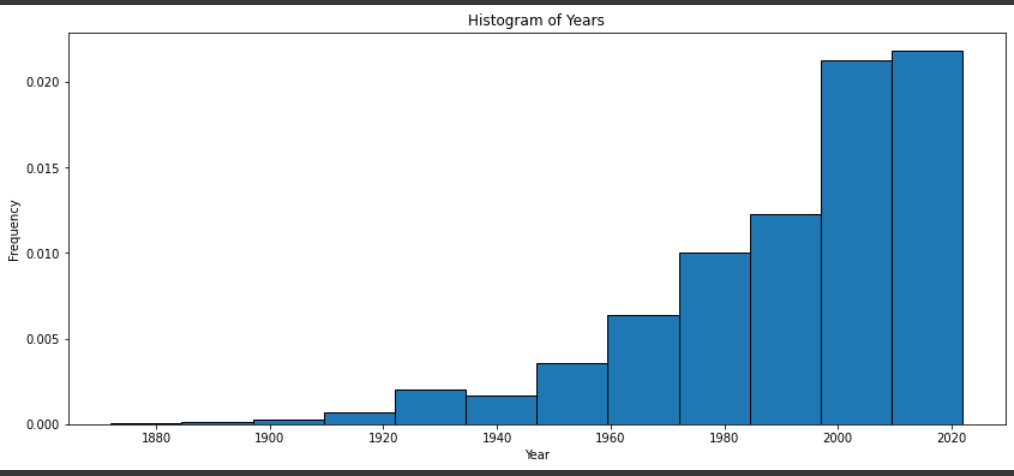
* The DataFrame has 9 columns as mentioned in the Data Dictionary. Data in each row corresponds to the match played by the teams.
* The data frame has 44341 rows and 9 columns.



We have 1 bool data type, 2 integers and 6 objects.

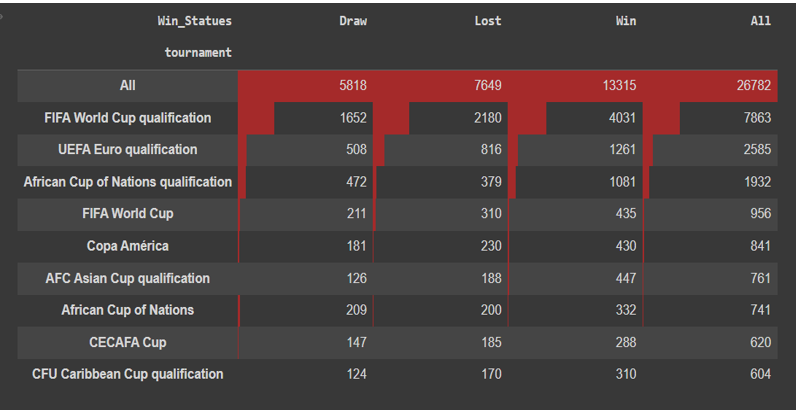
**Visual representation of all tournaments ever played:**

****

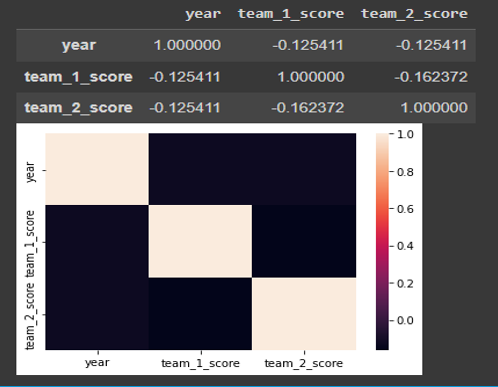
**** ****

Most of the matches were played after 1960.

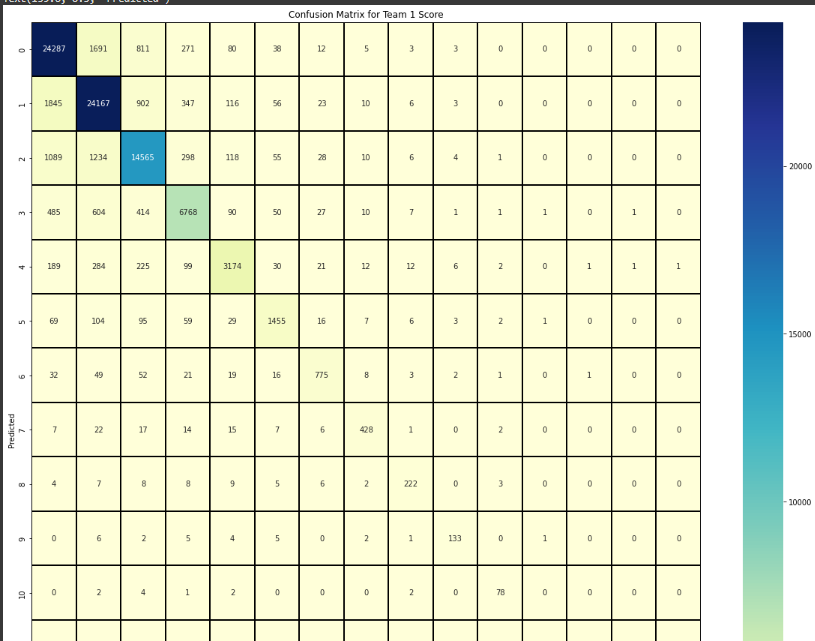
**Performance of home and away team:**

****

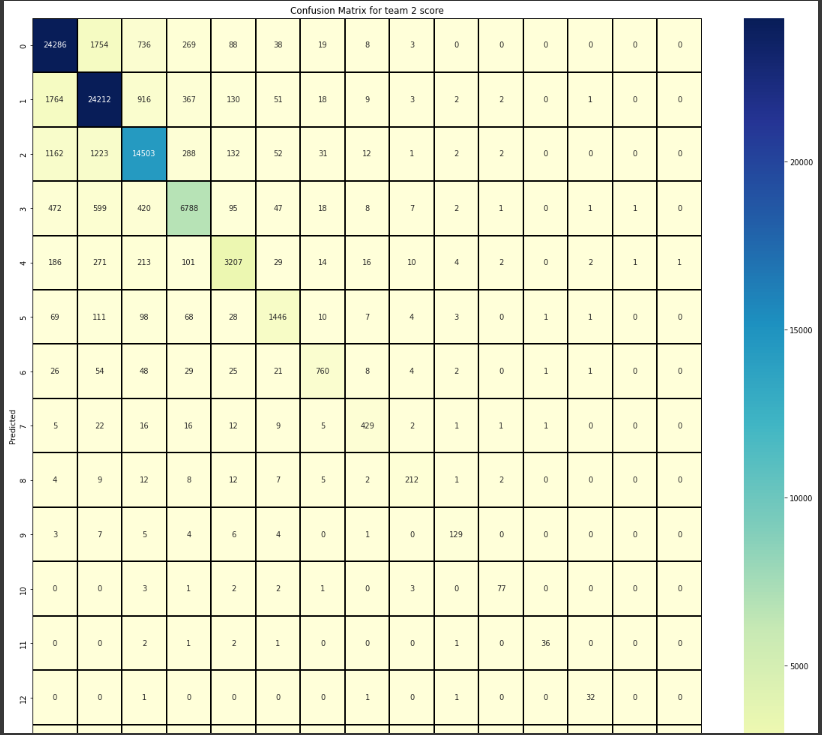
**Correlation:**

****

**Confusion matrix for team 1:**

****

**Confusion matrix for team 2:**

****

**Methodology:**

In this project, I used `MultiOutputRegressor` present in scikit-learn.

**Code for importing the above algorithm:**

X=categorized\_data

from sklearn.multioutput import MultiOutputRegressor

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

model = MultiOutputRegressor(RandomForestClassifier())

model.fit(X,Y)

MultiOutputRegressor in scikit-learn is a class that allows you to fit a multi-output regression problem, i.e., a problem where multiple response variables are predicted given one or more predictor variables. It does this by wrapping a single-output regression model and training it on each output column in the data. It allows you to fit regression models on multi-output datasets and provides a unified interface for all regression models in scikit-learn.

All the teams which have played atleast 20 matches in a year are selected and year wise I have created a pair of two two teams and then with the help of confusion matrix, their accuracy, precision, and score is calculated.

A confusion matrix is a table used to evaluate the performance of a classifier. It is a summary of the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions made by a classifier. It is used to calculate a number of metrics including accuracy, precision, recall, and F1 score, which are useful for understanding the performance of a classifier and how it's making predictions. The elements of the confusion matrix are defined as follows:

TP: The number of instances correctly classified as positive by the classifier.

FP: The number of instances incorrectly classified as positive by the classifier.

TN: The number of instances correctly classified as negative by the classifier.

FN: The number of instances incorrectly classified as negative by the classifier.

**Algorithm implementation:**

In fifa 2022, total 32 teams are going to compete and they will compete in in 8 groups.



First of all, best two teams will be selected from each group on the basis of prediction and then they will send to round of 16 competition and again two teams from each group will compete and finally we get total 8 teams and then they will compete in quarter finals.

Same process will applied again and 4 teams got selected for semi-finals and then two teams will be selected for finals and then again with the help of prediction, here are the results:

First place: Argentina

Second place: Russia

Third place: Poland