# Predictive Analysis of FIFA World Cup 2022.

Data Science and Machine Learning Project

Chaitanya Bachhav (csb152), Ritesh Dutta (rd969), Hasnain Gandhi (hg410)

***Abstract*—The project is based on analysing the international matches from 2011 till present and reason whether we are able to predict the outcome of a future match. The first part of the project involves selecting the correct data set, and then perform Exploratory Data Analysis (EDA) on the data to select those data which would help us in predicting the matches better. The second module involves implementing different Machine Learning algorithms to calculate the accuracy and confidence with which we can predict a particular result. The final module discusses the results along with an explanation on the uncertainty in the predictions and what the model doesn't incorporate.**

***Index Terms*—FIFA World Cup, Exploratory Data Analysis, Logistic Regression, Predictive modelling, Neural networks**

## I. Project Overview

In any real life scenario, predicting the future constitutes an important part of the study which a lot of statisticians and computer scientist are interested as it helps them to study trends and help the society. Some common examples here includes weather forecast, hurricane prediction. Several businesses are dependent on stock market and traders study the market to predict the profit and loss for particular stocks. So this problem has a vast use in the real domain. We have approached a problem which is more fun than life saving, where we plan to predict the outcomes of matches played in the **FIFA Football World Cup** which is the most prestigious football tournament in the world. The championship has been awarded every four years since the start of the tournament in 1930.

The current format involves a qualification phase, which takes place over the preceding three years, to determine which teams quality for the tournament. In the tournament, 32 teams, including the host nation, compete for the title at different stadiums in the host country. We test whether our model is good enough to account for the uncertainties that games produce and learn from those.

The next few sections explain the database we are using along with the different methods to process it before passing it through a machine learning model. We also briefly discuss about the libraries we are using in the project. These analysis gives us a clearer picture on the data and which features are useful in the predictions.

We then use different ML models and train our input data and find the accuracy on the test set. The last part of the report discusses about these results and future work on how to scale this to increase the accuracy.

## II. Tools and Technologies

The implementation of the project is done in python using google colab. The reason for using python is due to its ability to seamlessly use libraries similar to R but R is more suited for Statistical data analysis whereas it is easier to use Python for machine learning. The important libraries used for the project are

1) **NumPy:** Library used for manipulating arrays. Arrays are very frequently used in data science, where speed and resources are very important, and numpy arrays are lot faster than python lists.
2) **Matplotlib:** Used to plot and visualize results. Consists of important functions like plot(), scatter-plot().
3) **Plotly:** Plotly's Python graphing library makes interactive, publication-quality graphs similar to matplotlib. Difference between the libraries is the ease of use for matplotlib over the sophistication of plots.
4) **Seaborn:** Uses matplotlib as a wrapper and helps in visualizing random distributions.
5) **Scikit-learn:** Main library used for machine learning, has important functions like Logistic regression, MLP, random forest.
6) **Tensorflow:** Tensorflow is a very useful f deep-learning library. Tensorflow was used to create a feedforward neural network in the project.
7) **Pandas:** Pandas is a python library for data analysis. We used pandas to manipulate data for processing and analysis, pandas stores data in the form of a DataFrame which is a type of data-structure.

## III. Data Set Description

The dataset(FIFA World Cup) for this project is taken from Kaggle. The dataset contains 25 exploratory variables also called as parameters. We first go through the original data explaining each column and its data type. Table **??** shows the dataset.

- **date**: (date) Date of Match
- **home_team**: (String) name of home team because of ground where the game is played.
- **away_team**: (String) name of away team because of ground where the game is played.
- **home_team_continent**: (String) The continent of the home team.
- **away_team_continent**: (String) The continent of the away team.
- **home_team_fifa_rank**: (integer) The FIFA rank of the home team when the match is played
- **away_team_fifa_rank**: (integer) The FIFA rank of the away team when the match is played
- **home_team_total_fifa_points**: (integer) Total fifa points of home team when the match is played.
- **away_team_total_fifa_points**: (integer) Total fifa points of away team when the match is played

- **home_team_score**: (integer) Full-time home team score including extra time, not including penalty-shootouts.
- **away_team_score**: (integer) Full-time away team score including extra time, not including penalty-shootouts.
- **tournament**: (string) Name of the tournament
- **city**: (string) host nation city
- **country**: (string) host nation country
- **neutral_location**: (boolean) If match played at neutral venue.
- **shootout**: (boolean) If match included penalty shootout.
- **home_team_result**: (string) Match result of home team.
- **home_team_goalkeeper_score**: (integer) Game score of highest ranked goalkeeper of home team.
- **away_team_goalkeeper_score**: (integer) Game score of highest ranked goalkeeper of away team
- **home_team_mean_defense_score**: (integer) Average FIFA game score of the 4 highest ranked defensive players of the home team.
- **away_team_mean_defense_score**: (integer) Average FIFA game score of the 4 highest ranked defensive players of the away team.
- **home_team_mean_offense_score**: (integer) Average FIFA game score of the 3 highest ranked attacking players of the home team.
- **away_team_mean_offense_score**: (integer) Average FIFA game score of the 3 highest ranked attacking players of the away team.
- **home_team_mean_midfield_score**: (integer) Average FIFA game score of the 4 highest ranked midfield players of the home team.
- **away_team_mean_midfield_score**: (integer) Average FIFA game score of the 4 highest ranked midfield players of the away team.

## IV. Dataset Preparation

The data includes matches details from 1993 on-wards, as a result a lot of entries are irrelevant due to the change in the formats, players etc. So we implement the following changes to trim the data-set.

- First drop the columns which we believe are not required for analysis. So we remove the following - home_team_continent, away_team_continent, neutral_location, country, city, tournament, shoot_out
- Next our focus is to predict the teams playing the current FIFA world cup, so we only work with entries of 32 teams currently playing the world cup.
- The column names **home_team** and **away_team** will be renamed to **team_A** and **team_B** since the home and away prefixes are irrelevant with respect to the World Cup.
- Lastly we also create a country wise data which will be useful in analysis of data.

## V. Exploratory Data Analysis

The dataset, contains a lot of important inherent characteristics that need to be explored before we start running the Machine learning model.

- We start with creating a scatter plot for the important features.
  We observe from figure 1 that there seems to be some correlation between different parameters. We see that as the FIFA rank increases, the total points decreases which is logical as it means the team is performing worse.

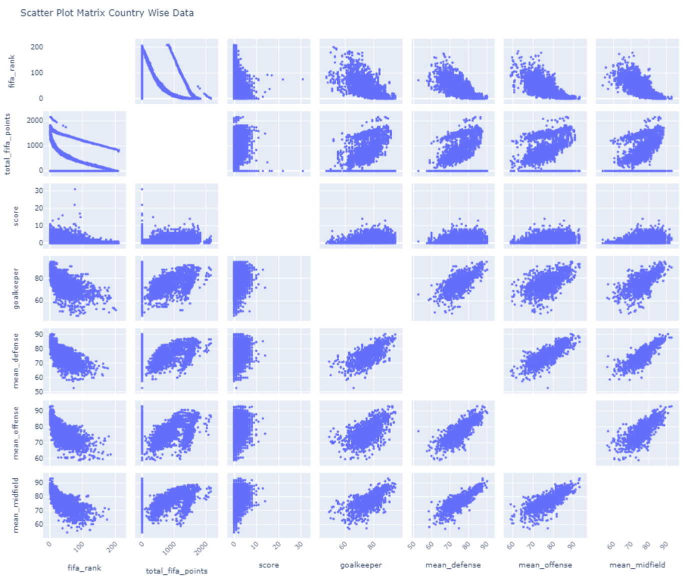


Fig. 1: Pair Plot of each of the features

- To make our understanding better let us now create a heatmap to get an idea on the exact correlation between the different features. From fig 2 we get a clear understanding that it seems that the team details are correlated strongly which basically means that goalkeeper's, midfield, offense and defense are correlated. Also notice that rank of the team or country is also negatively correlated to the scores, so the rank difference would be a better alternative.
- We now take the data of only 4 countries which reached semifinal to analyse better the values of their goalkeepers and midfield score. These 4 countries are France, Argentina, Morocco and Croatia. Figures 3 and 4 showcases the value for goalkeepers and midfield score.
- Another important metric is goal difference for these teams. This is important because goal difference basically includes information of the entire team. It denotes team work in the sense a team which can score as well as defend and keep clean sheets. Figure 5 shows that and we see that France has one of the best all round performances.
- Then we analyse the win rate with the rank difference. This metric will give us clear correlation between them. Two teams having more rank difference clearly tells us that one with smaller rank is bound to win.
- Upon completing the exploratory analysis, we can decide what columns/features should be used to predict the outcome of the matches. The prediction will depend on these variables. The model that

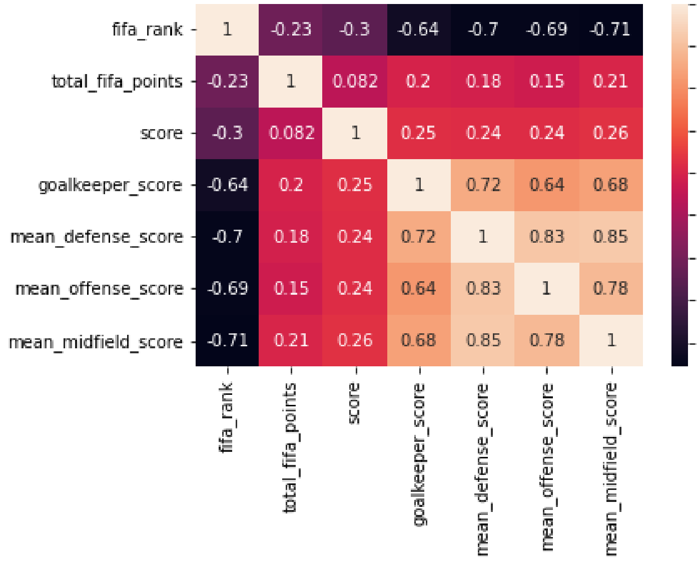


Fig. 2: Heat-map for correlation between country wise columns.

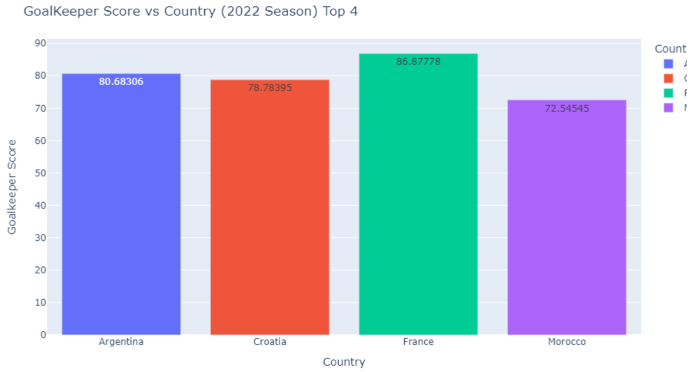


Fig. 3: Goalkeeper scores for 4 countries

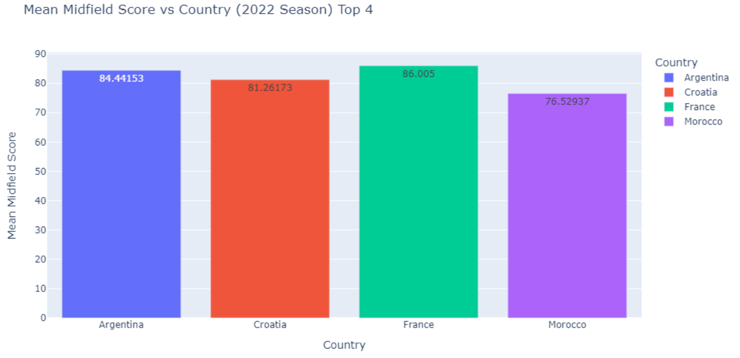


Fig. 4: Midfield scores for 4 countries

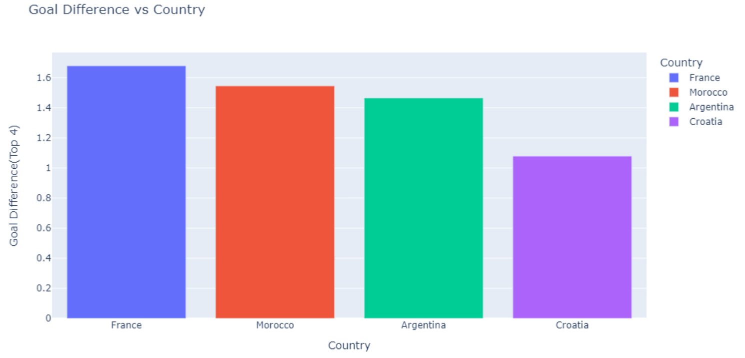


Fig. 5: Goal difference for top 4 countries

we used contains the following features: rank_diff: Difference between ranks of the two teams. We calculate this from the original kaggle dataset. points_diff: Difference between the points of the two teams, we calculate this from the original kaggle dataset, team_A: Team/Country Name, team_B: Team/Country Name, team_A_goalkeeper_score, team_B_goalkeeper_score, team_A_mean_midfield_score, team_B_mean_midfield_score

## VI. ML Algorithms Description

We have used the following algorithms to first classify the data and then use it to predict the value of a future match. Below we have mentioned briefly the algorithm details and then the implementation we have used.

- Logistic Regression: Logistic regression is a multi-variable method devised for binary outcomes. It calculates the probability of a certain outcome and rounds off that probability to 0 or 1. This can also be used for more than two outcomes. The cost function (Eq.1) calculates the cost the algorithm pays if it predicts a value h$_\theta$(x) while the actual cost label turns out to be y. By using this function we will grant the convexity to the function the Gradient Descent algorithm needs to process.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^i), (y^i)) \tag{1}$$

$$Cost(h_\theta(x), y) = -log(h_\theta(x)) \text{ if y=1} \tag{2}$$

$$Cost(h_\theta(x), y) = -log(1 - h_\theta(x)) \text{ if y=0} \tag{3}$$

In Eq.2, the output, or the cost to pay, approaches 0 as h$_\theta$(x) approaches 1. Conversely, h$_\theta$(x) approaches 0 as the cost to pay grows to infinity. This is a desirable property; we want a greater penalty when the algorithm predicts something farther away from the training value. If label y is equal to 1 but the algorithm predicts h$_\theta$(x) equal to 0, the result is wrong and a greater penalty should be applied. The same intuition continues in Eq. 3 where the result is vice-versa. We have used the scikit-learn implementation of Logistic Regression as a linear model for binary classification. We have used the default second-derivative solver as our data set is quite small.
- **Random Forest**: Random Forest Classifier is an ensemble algorithm. Ensembled algorithms are those which combine more than one algorithm of the same or different

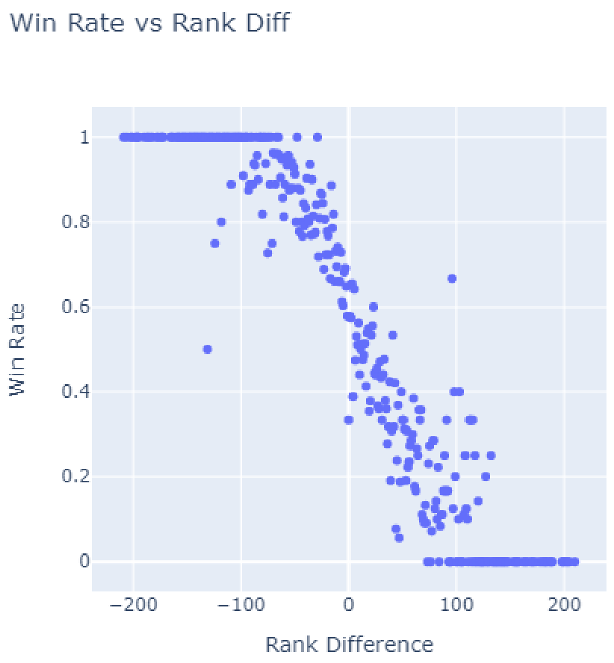


Fig. 6: Win rate versus the rank difference for all teams

kind for classifying objects. For example, running prediction over Naive Bayes, SVM, and Decision Tree and then taking a vote for final consideration of class for a test object. A random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It then aggregates the votes from different decision trees to decide the final class of the test object. Random forest consists of a large number of individual decision trees. We have used the scikit-learn implementation of the Random Forest Classifier for binary classification. The parameters used in this model are:

n_estimators: The number of times bootstrapping is done to generate trees

class_weight: This is used to automatically adjust weights inversely proportional to class frequencies to make the predictions more ”balanced”

- **Multi-Layer Perceptron**: A Perceptron is a single-layer neural network, which has an input layer, an output layer, and weights that are updated to classify/predict the labels. It is a linear classifier algorithm as the activation function is a linear combination of weights and input and we can get good accuracy with limited training data. Incorporating a layer of hidden neurons into our input and output layers makes our model a Multi-Layered Perceptron. This can be shown in figure 7 below. In our implementation, we have used an in-built Keras perceptron with 4 hidden layers of sizes = (512,256,256,128) along with an input and output layer. We have used relu as an activation function in all the layers except the output layer where we have used a sigmoid function to output categorical values.

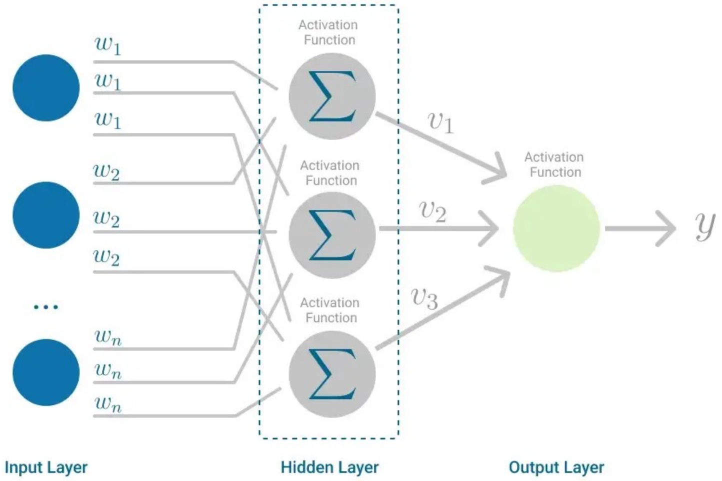


Fig. 7: Illustration of a general MLP architecture.

An Adam gradient descent optimizer is used along with a binary-cross-entropy loss function. We have also included features like balanced weights, checkpoint saving and early stopping.

## VII. Analysis and Results

Based on the above algorithms we have trained the model and got the following results:

1) Logistic Regression: For our model the confusion matrix is given by 8. We can see that the true values across diagonal are high. The accuracy of the model came at **75%**

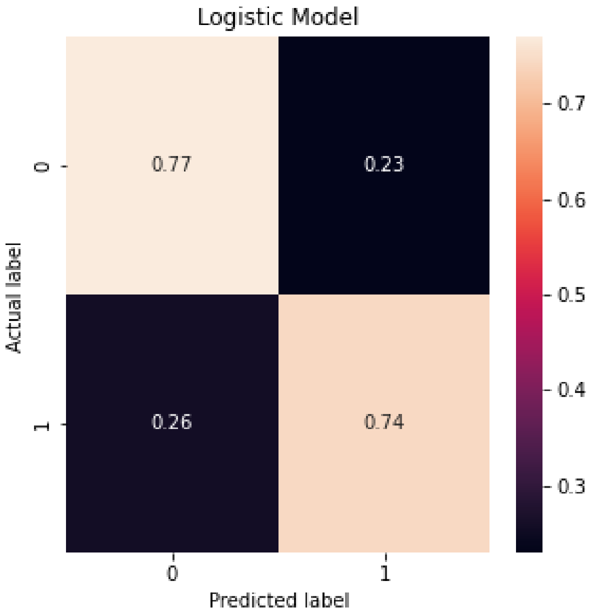


Fig. 8: Confusion Matrix for Logistic Regression

2) Random Forest: For our model the confusion matrix is given by 9. We got an accuracy of **72%**.
3) Neural Network: For the model, our results are given by figures 10,11, 12. The accuracy for MLP is **72%**

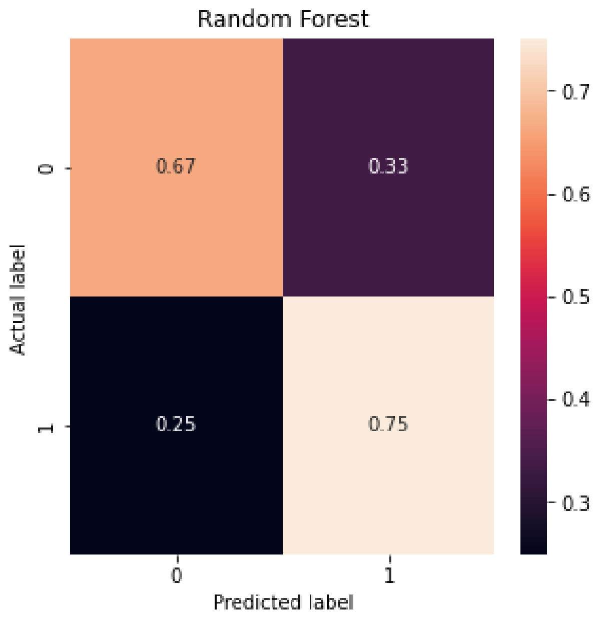


Fig. 9: Confusion Matrix for RF

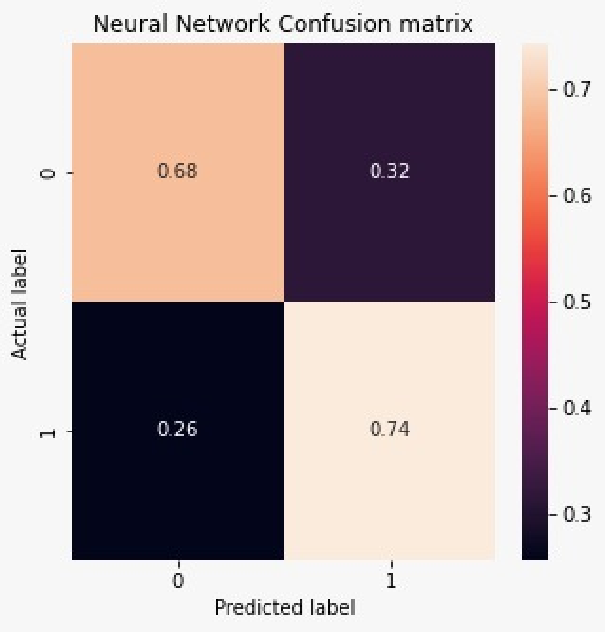


Fig. 10: Confusion Matrix for Feedforward Neural Network.

## VIII. Prediction and Understanding the Answer

After training the model on our testing data, we then created a visual tool where the user can enter 2 countries and based on the trained model, it predicts the winner. There is some uncertainty associated with the data, which is attributed to the fact that an upset is not anticipated by the user.
Image 13 shows the input as Argentina and France and using our model predicted the answer as **Argentina** in 2 models whereas a **France** victory for 1 model.
An important observation we got was that for teams having similar FIFA rank and points, we see that there would be difference in the prediction amongst different models.

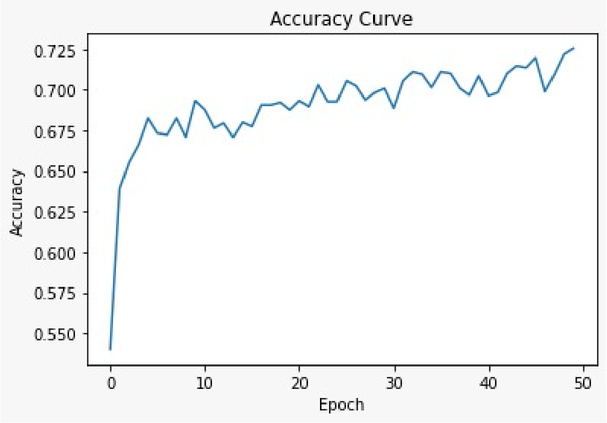


Fig. 11: Accuracy trend for Feedforward Neural Network

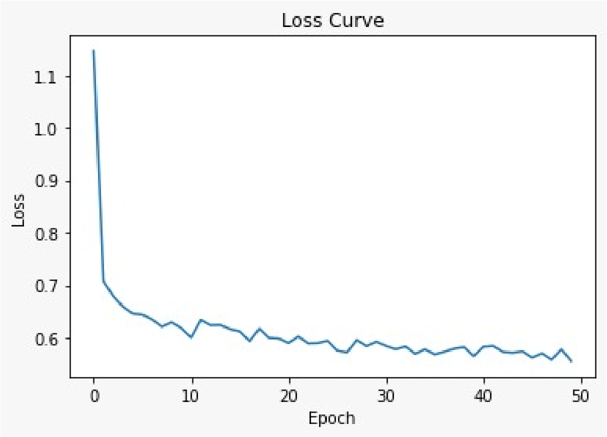
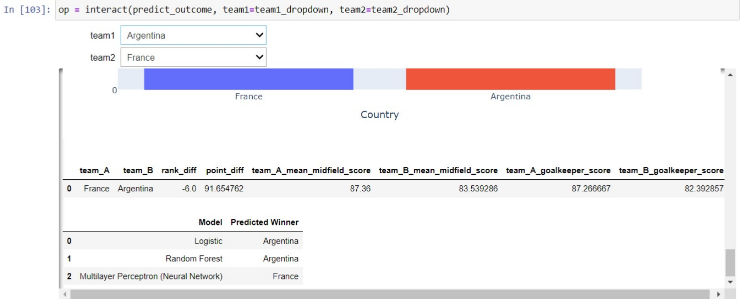


Fig. 12: Loss function of Feedforward Neural Network

Whereas for teams having a significant difference in their ranks, the model predicts similar results giving the win to a stronger side.



```
In [103]: op = interact(predict_outcome, team1=team1_dropdown, team2=team2_dropdown)
```

team1: Argentina

team2: France

| | team_A | team_B | rank_diff | point_diff | team_A_mean_midfield_score | team_B_mean_midfield_score | team_A_goalkeeper_score | team_B_goalkeeper_score |
|---|---|---|---|---|---|---|---|---|
| 0 | France | Argentina | -6.0 | 91.654762 | 87.36 | 83.539286 | 87.266667 | 82.392857 |

| | Model | Predicted Winner |
|---|---|---|
| 0 | Logistic | Argentina |
| 1 | Random Forest | Argentina |
| 2 | Multilayer Perceptron (Neural Network) | France |

Fig. 13: UI for team and predicted data for each model.

## References

[1] Brenda Loznik. Fifa world cup 2022, 2022.
[2] Python machine learning module.
[3] Python plotting library.
[4] Python data analysis library.