**Title:**

**Soccer Analytics using Deep learning**

**Abstract:**

Our work proposes use of deep learning techniques for soccer match winner prediction and also how we can use deep learning for building a soccer team using deep learning.

**Dataset:**

For soccer match winner prediction, we have taken a dataset from <http://football-data.co.uk/data.php>. For this purpose we have taken data of three sessions 2021/2022, 2020/2021, 2019/2020 from France ligue le championtat.

For building a soccer team using deep learning, we have taken a dataset from kaggle (FIFA 21 complete player dataset). Link:<https://www.kaggle.com/stefanoleone992/fifa-21-complete-player-dataset>

**Experiment:**

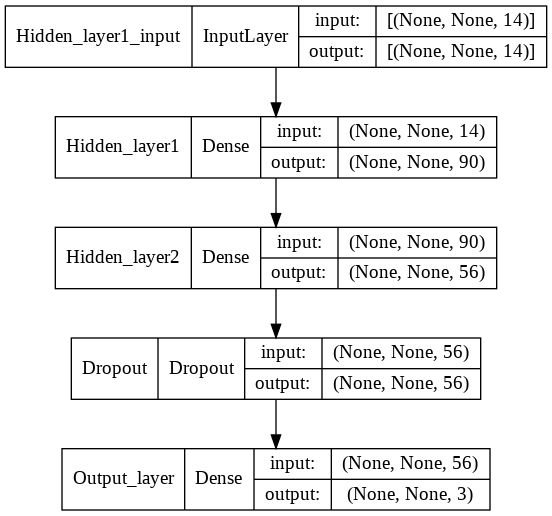
For soccer match winner prediction, we first follow the basic machine learning pipeline steps like dropping redundant columns which are not useful for further analysis. The dataset contained 104 features. There are 889 different teams. As these are club leagues so either the soccer game is played at home or different venues. Home means the venue the club originates from. The target variable is which team will win the game either the home team, away team or the match will be draw. The dataset contained 337 instances that the match will result in a draw, 302 instances that the match will be won by the home team, 250 instances that the match will be won by the away team. As we have checked multicollinearity between the different features using Variance inflation factor (VIF). Keeping the threshold of VIF as 10, below 10 multicollinearity is absent otherwise multicollinearity is present. High multicollinearity affects the outcomes of the algorithms. So we applied various feature selection techniques to select the best features and also to keep the multicollinearity between the features as low as possible. The important features after feature selection are:

HomeTeam,HF = Home Team Fouls Committed, AF = Away Team Fouls Committed, HS = Home Team Shots, HC = Home Team Corners, AwayTeam

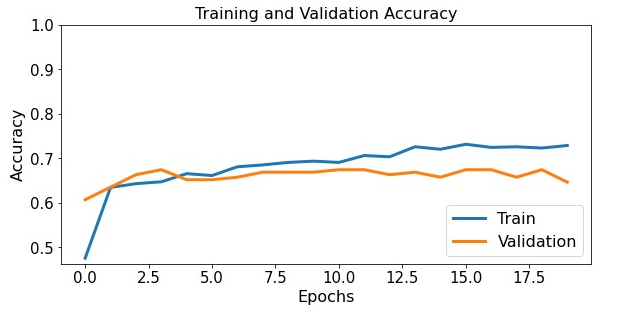
,AvgA = Market average away win odds, AS = Away Team Shots, AvgH = Market average home win odds, AST = Away Team Shots on Target

,HTAG = Half Time Away Team Goals, HST = Home Team Shots on Target, HTHG = Half Time Home Team Goals, HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win).Target variable is FTR (Full Time Result) (H=Home Win, D=Draw, A=Away Win). We have encoded the categorical variables with the help of LabelEncoder of scikit-learn library. We have also scaled the features to the same scale using StandardScaler from scikit-learn library. As no separate testing data is available so we will keep 20 % of the data for testing purposes.

We have built a deep learning model with the help of Tensorflow library for training on the above dataset. The deep learning model architecture is shown below (Fig1). The model architecture consists of two hidden dense layers, one dropout layer and an output layer. Softmax activation function was used as the activation function in the output layer. The loss function used was categorical cross entropy loss. The optimizer used was Adam and the metric was accuracy. The model was trained for 20 epochs. The model architecture contained 6,617 trainable parameters. A callback was used while training the model that saves the best weights after every epoch if the validation set accuracy increases. Here the test set was used as a validation set while training the deep learning model.

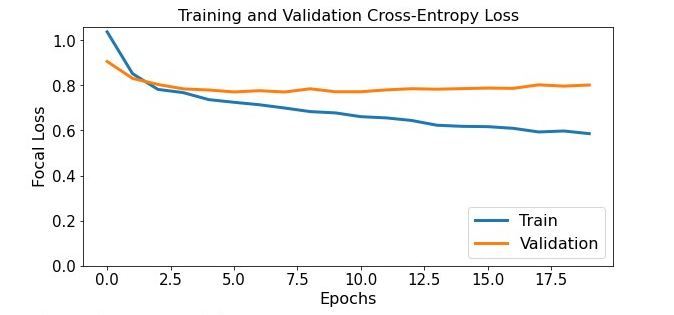
 Fig 1

The model was able to obtain 72.85% train accuracy and an accuracy of 67.42% on the test set. The plot of training and validation accuracy while training for each epoch is shown below (Fig 2).

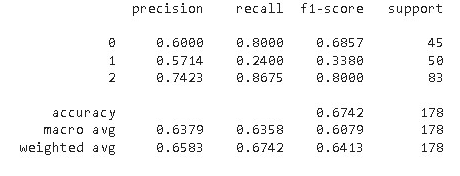
Fig 2

As we can see from the above plot, validation accuracy is decreasing near 20 epochs or the loss is increasing, that is a sign of overfitting so we will stop the training at 20 epoch. Overfitting leads to bad performance of the deep learning model, it prevents the model from generalizing.

The plot of training loss and validation loss while training for epoch can be seen below(Fig 3).

 Fig 3

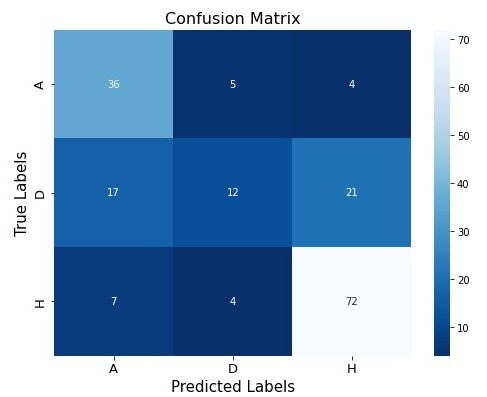
The classification report when the model was used to predict on the test set can be seen below. (Fig 4)

 Fig 4

Here 0 stands for A: Away Win, 1 stands for D: Draw and 2 stands for H: Home Win. As it can be seen from the confusion report that f1-score for class 1 is very low. As soccer is an unpredictable game and predicting which team will win is a very difficult task and out of the most difficult is to predict whether a match will result in a draw.

The confusion matrix for the model when it was used to predict on the test set.

The confusion matrix can be seen below. (Fig 5)

 Fig 5

Here A: Away Win, D: Draw, H: Home Win

The frontend for soccer match winner prediction is the file Final\_UI\_smp.png.

We are building a deep learning model that can classify whether the player will be offered club membership or not based on various factors so that cost remains within the potential budget and also provides market gains to investors.

For selecting a certain player a lot of variables come into effect as a player who plays in forward position cannot be judged by his defending skills or vice versa.

For building a soccer team using deep learning, the dataset contained 106 features and the no of instances was 18944. The features that are present in the dataset are: age, height\_cm, weight\_kg, nationality, wage\_eur etc. The dataset contains profiles of players from 162 nations. We will follow the basic machine learning project steps such dropping redundant features which will not be useful for further analysis. We have kept the ‘overall’ column as the target variable. After manual inspection and feature selection techniques only 54 features were selected. Among the best selected features, some features are age, height\_cm, mentality\_vision, movement\_reactions etc.

We have encoded the categorical variables with label encoder from scikit-learn library. We have also applied feature scaling to bring all the features to the same scale using Standardscaler from scikit-learn library.

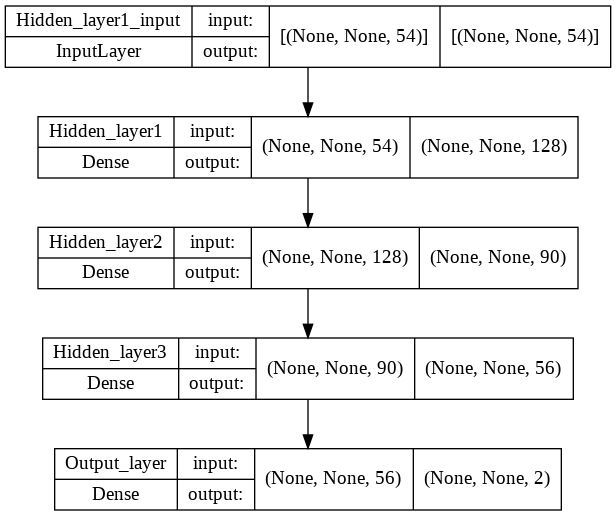
In the target variable, 0 stands for the player will not be selected or the player will not be a club member. 1 stands for the player will be selected or the player will be a club member. Class 0 contained 13547 instances and Class 1 contained 5347 instances. So we can see that classes are slightly imbalanced. As no separate training set was available so 20% of the dataset was kept as a test set.

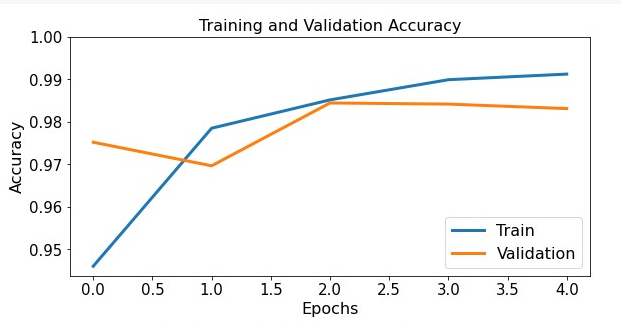
15155 instances were kept as training set and 3789 instances were kept as test set.

We have built a deep learning model using tensorflow library. The model architecture contained three hidden layers and an output layer. Softmax activation function was used as activation function for the output layer. The loss function used was categorical cross entropy loss. The optimizer used was Adam and the metric was accuracy. The model contained 23,860 trainable parameters. A callback was used while training the model that saves the best weights after every epoch if the validation set accuracy increases. Here the test set was used as a

validation set while training the deep learning model. Model architecture (Fig 6)

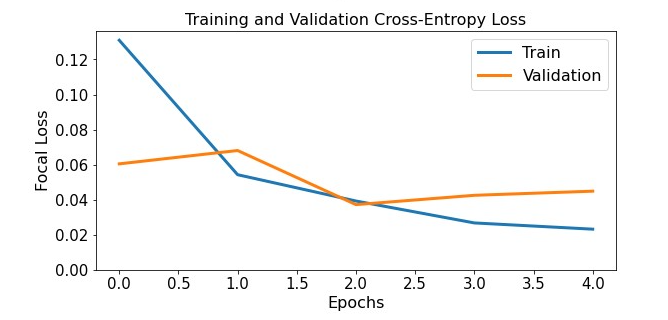
Fig 6



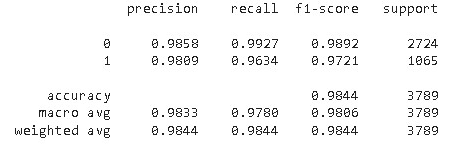
Fig 7

The plot of training loss and validation loss while training for epoch can be seen below(Fig 8).

As we can see, validation accuracy starts to decrease or validation loss starts to increase near the 5th epoch so we trained the deep learning model for only 5 epochs. Because training it more will lead to overfitting, which is not desirable for deep learning models as it prevents the model from generalizing on unseen data.

Fig 8

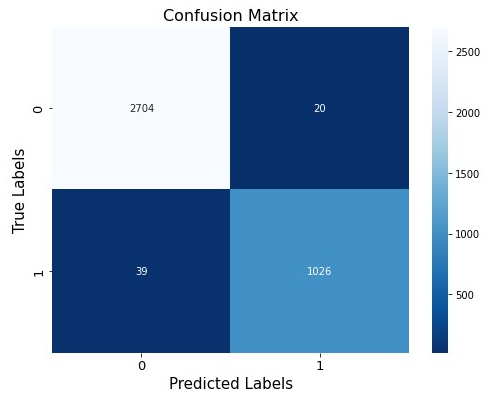
The classification report when the model was used to predict on the test set can be seen below. (Fig 9). From the classification report, we can see that we were able to achieve 98.44 % accuracy and also before we have seen that the classes were imbalanced, but from the classification report we can see that f1\_score(macro\_avg) is 98.06 % which is high. This means the model was able to prevent the class imbalance problem.

 Fig 9

Here 0 stands for the player will not be selected or the player will not be a club member. 1 stands for the player will be selected or the player will be a club member.

The confusion matrix for the model when it was used to predict on the test set.

The confusion matrix can be seen below. (Fig 10)

Fig 10

The Frontend UI for building a soccer team using deep learning is the file Final\_UI\_bst.png.