CSE 532 Project Update 2: Soccer Analytics

Akhilesh Soni, soni6@wisc.edu

Github repository: https://github.com/soniakhilesh/soccer-analytics
In the last update, we extracted various features which were based on domain knowledhe of soccer. Some of these features were shots on target, goals scored, fouls committed, etc. We also introduced form feature based on shots of target. We refine our form feature include goals as well as shots on target. $Form = \sum_{i=1}^{5}(STD + 2 * GD)e^{-\alpha i}$. Here STD stands for shots on target difference between the home and away team and GD stands for goal difference between home team and the away team. We give twice the weight to goal difference compared to shots on target difference as goal scored have a bigger impact on the game outcome team morale. We used $\alpha = 0.6$ in the Form expression. This is based on domain knowledge that recent games carry more effect on team's morale and hence performance in next game compared to games they play 2 weeks back.

With these features, we tried the following models on our dataset. We use sklearn to fit the models and GridSearchCV functionality to perform hyper parameter tuning for each model. We used training accuracy to choose the best hyper parameters. Accuracy is the ratio of correctly labeled observations to the total number of observations. We also perform cross validation in the training set to avoid over-fitting.

Note that we decided to use KNN as a base estimator instead of k-means as KNN seemed more intuitive for this problem.

As we are dealing with multi class problem, we use a OneVsRest methodology to handle multiple classes (Win,Draw,Loss) as listed in our previous project update.

- KNN: K nearest neighbors. We keep this as a benchmark. The hyperparamters to tune in this case are number of neighbors and p in ℓ_p norm. We consider $N \in \{5, 20, 50, 200\}$ and $p \in \{1, 2\}$
- SVM: Linear and kernel based: We tried kernel based SVM methods. In particular, we used linear and poly kernels. We also optimized the choice of penalty paramter $\in \{0.1, 1, 10, 100\}$
- Neural Network: We ran Neural Network using Keras library in python. Our input dimension is 41. We tried 6 layers network with 200 nodes in each layer. We used Rectified Linear Unit activation functions in each hidden layer and a softmax in output layer.

Model	**Parameters**	**Accuracy**
KNN	N=200, ℓ_1 norm	55.4%
SVM	Penalty = 100, Kernel=Linear	58.4%
Neural Network	Input dim=41, Layers=6, Nodes in each dense layer=200	46.4%

Table 1: Model comparisons

We observe that SVM performs slightly better better than KNN. However performance of Neural Network seems below par. We believe that Neural network still needs some fine tuning to improve model performance. This will be of the major tasks moving ahead in the project. Next steps of the project:

- Identifying the most important features: We plan to study which features influence the performance of the model most, This will make it easier to interpret the results as currently we have 41 features in the input dataset.
- Fine tuning performance of Neural Network. We believe that Neural network should be able to outperform the other two methods but currently we are getting opposite results. Hence, we plan to spend sometime in refining our network
- Start documenting and preparing the report.