PREDICTION OF NUMBER OF GOALS WILL BE SCORED BY A PLAYER IN A MATCH

Lab Grp No : A2

ID : 14.02.04.034

Introduction

Football is the game of champions. It is one of the most popular games in the world. The players work hard to score a single goal. Now scoring a goal depends on many attributes. In our project we have tried to predict how many goals a player will score in a match. In this problem we have made a custom dataset which have different needed information. Our aim is to predict the goal number a player will score. We have built it like a classification problem. We have selected 4 classes.

Dataset Description

Our dataset is made up of 6 attributes or columns. In which the first 5 are features and the 6th or the last attribute is target class. To predict goals some features are important like the player's rate, the opponent team's rate etc. the features and their description is given below.

- **POSITION:** In which position a player plays. Goal scoring possibilities increases if a player is a striker. We have denoted with 0, 1, 2 or 3. 0 means striker,1 means left wing , 2 means right wing , 3 means central attacking midfielder.
- **FINISHING:** Finishing accuracy. Which is less than 100. The more accurate the finishing power the higher the rating.
- **RECENT_SCORING_FROM:** Recent scoring form denoted by 0 or 1. 0 means good form and 1 means bad form.
- **OPPONENT_TEAM_RATING:** Opponent team rating which denoted by 0 or 1. If 0 then bad team and if 1 then good team.
- **HOME_OR_AWAY:** Home or away match. Home match is 0 and away is denoted by 1.
- **GOALS:** : It is the target class. Denoted by 0, 1, 2 or 3. It is mainly describing the goal number a player scores or should score.

Models

Random Forest Classifier

Random Forest Classifier is an ensemble algorithm. Ensemble algorithms are those which combines more than one algorithms of same or different kind for classifying objects. Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object. This works well because a single decision tree may be prone to a noise, but aggregate of many decision trees reduce the effect of noise giving more accurate results In our project we have used the following code to run random forest classifier. clf = RandomForestClassifier(n_estimators=10) here , n_estimators: Number of trees in the forest. Default is set to 10. It means this classifier will create 10 decision trees from randomly selected subset.

Multi-layer Perceptron

A MLP is a feed-forward artificial neural network model that maps a set of input data onto a set of appropriate outputs. An MLP is a network of simple neurons called perceptron. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. Mathematically this can be written as where W denotes the vector of weights, X is the vector of inputs,

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

b is the bias and is the activation function. A signal-flow graph of this operation is shown in the below figure. A typical MLP network consists of a set of source nodes forming the input layer, one or more

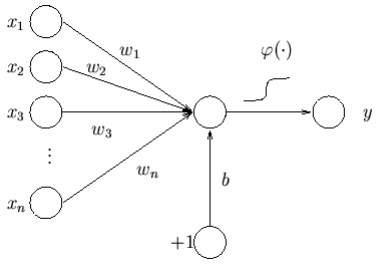


Figure 1: A Signal-Flow Graph

hidden layers of computation nodes, and an output layer of nodes. The input signal propagates through the network layer-by-layer. The signal-flow of such a network with one hidden layer is shown in Figure2

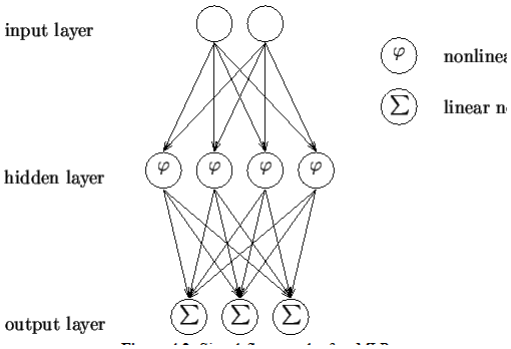


Figure 2: Signal-Flow Graph

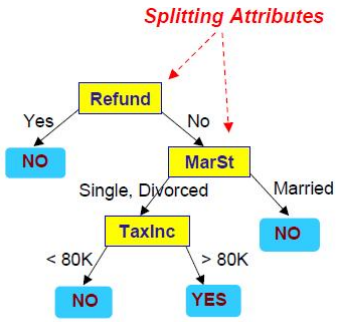
Decision Tree

Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receive an answer, a follow-up question is asked until a conclusion about the class label of the record is reached. For an example, we can think of a training data to determine if a person cheats. In the decision tree, the root and internal nodes contain attribute test conditions to separate records that have different characteristics. All the terminal node is assigned a class label Yes or No. Once the decision tree has been constructed, classifying a test record is straightforward. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test.
Advantages of this classification

- Simple to understand, interpret, visualize.
- It can handle both numerical and categorical data.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

Disadvantage of this classification

- Decision trees can be unstable because small variation in the data might result in completely different tree regenerated.

Adaboost

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire. It can be used in conjunction with many other types of learning algorithms to improve performance. Ada-boost, like Random Forest Classifier is another ensemble classifier. Ada-boost classifier combines weak classifier algorithm to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier. Each weak classifier is trained using a random subset of overall training set. After training a classifier at any level, ada-boost assigns weight to each training item. Misclassified item is assigned higher weight so that it appears in the training subset of next classifier with higher probability.After each classifier is trained, the weight is assigned to the classifier as well based on accuracy. More accurate classifier is assigned higher weight so that it will have more impact in final outcome. The equation for the final classifier, ht(x) is the output of weak classifier t for input x alpha_t is weight assigned to classifier. After

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

weak classifier is trained, we update the weight of each training example with following formula Adaboost like random forest classifier gives more accurate results since it depends upon many weak classifier for final decision. One of the applications to Adaboost is for face recognition systems.

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

K-Nearest Neighbors

K-nearest neighbor algorithm is one of the simplest classification algorithm and it is one of the most used learning algorithms. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. KNN Algorithm is based on feature similarity. How closely out-of-sample features resemble our training set determines how we classify a given data Example of k-NN classification. The test sample

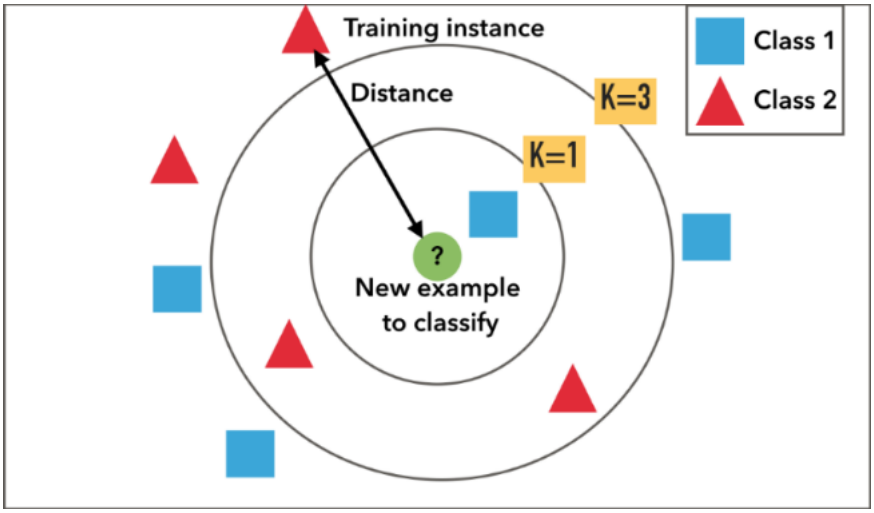


Figure 3: KNN

(green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3 (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. To calculate distance we use many equations like Euclidian distance.

Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. Support Vector Machine" (SVM) is a

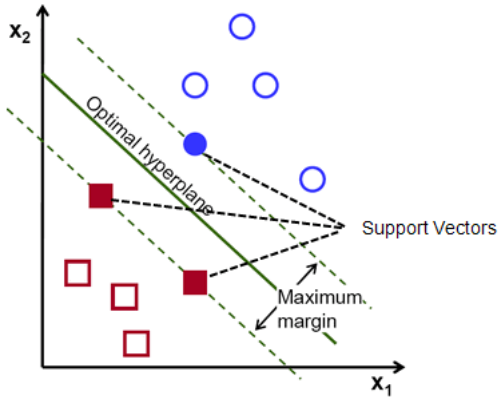


Figure 4: SVM

Model name	Train Score	Test Score
Random Forest Classifier	0.977528	0.782609
Stochastic Gradient Decent	0.820225	0.869565
Support Vector Machine	0.876404	0.739130
K-Nearest Neighbors	0.921348	0.782609
Decision Tree	0.977528	0.913043
Multi-layer Perceptron	0.966292	0.782609
AdaBoostClassifier	0.966292	0.782609

Table 1: Accuracy After Train Test Split

supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning. SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing.
The advantages of Stochastic Gradient Descent are:

- Efficiency.
- Ease of implementation (lots of opportunities for code tuning).

The disadvantages of Stochastic Gradient Descent include:

- SGD requires a number of hyperparameters such as the regularization parameter and the number of iterations.
- SGD is sensitive to feature scaling.

SGD is a very popular numerical procedure to find the local minimum of a function (in this case, the loss function, which measures how far every instance is from our boundary). The algorithm will learn the coefficients of the hyperplane by minimizing the loss function.

Comparison Of Different Models

For testing the performance of different models or classifiers we have chosen 4 performance metrix. Different comparison is given below.

Model name	Accuracy	Precision	Recall	F1 Score
Random Forest Classifier	0.813438735178	0.664915084915	0.657150997151	0.647778269661
Stochastic Gradient Decent	0.794466403162	0.650742296919	0.693668091168	0.631397232397
Support Vector Machine	0.85652173913	0.677455969956	0.681502849003	0.659171640465
K-Nearest Neighbors	0.795652173913	0.629203805454	0.631780626781	0.60275351181
Decision Tree	0.885375494071	0.747738095238	0.746944444444	0.735762489626
Multi-layer Perceptron	0.813438735178	0.640705128205	0.648297720798	0.620073319926
AdaBoostClassifier	0.793675889328	0.581595441595	0.592489316239	0.559018290255

Table 2: Accuracy After K-Fold Cross Validation

Discussion

In the 7 methods we have used in this project decision tree gives us the best result. Decision tree is a simple to understand classification problem. . Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receive an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.