Machine Learning in Circus Maximus

Ali Asghar Aamir
LUMS
21100124@lums.edu.pk

Ali Hassan Bukhari
LUMS
21100187@lums.edu.pk

Abstract

Predictions involving the outcome of football matches and leagues is a common phenomenon amongst Machine Learning and football enthusiasts of today. In the present study, we build on the same approach but also demonstrate a novel method using relatively popular and robust machine learning methods such as neural networks and support vector machines. We simultaneously contrast their results with lazy learners, such as the K-nearest neighbors algorithm as well as the logistic regression and Bayes classifier, to predict the results of the most popular football league in the world, the English Premier League (EPL). We also train multivariate models in the hopes of identifying the most optimum algorithm(s) with the highest accuracy and lowest error to achieve this task. Our models are based on multiple resources, long time series, and globally distributed datasets.

1 Introduction

Predictive models are used in a variety of domains from sports outcomes and TV ratings to technological advances and corporate earnings. These models are built from "experience", which constitutes data acquired from actual cases. The data can be pre-processed and expressed in a set of rules, such as it is often the case in knowledge-based expert systems, or serve as training data for statistical and machine learning models. Among the options in the latter category, the most popular models in medicine are logistic regression (LR) and artificial neural networks (ANN).These models have their origins in two different communities (statistics and computer science), but share many similarities.

In this article, we show that logistic regression and artificial neural networks along with other popular machine learning models such as the Bayes classifier, K-nearest neighbors, and support vector machines can be used to make our desired predictions and a draw a comparison between these models to conclude which model is the best suited for such tasks.

There are now several implementations of predictive modeling algorithms readily available, both as free and commercial software. The quality of the results obtained using these models mainly depends on three factors: the quality of the data set employed in model-building, the care with which adjustable model parameters were chosen, and the evaluation criteria used to report the results of the modeling process.

Overall, the data set consisted of 20 year-wise datasets that were amalgamated into a final data set with the processed data. This final data set was further used to make predictions and run the aforementioned machine learning models.

2 Preliminaries

2.1 K-nearest Neighbors

In this method the Euclidean distance between each data point and all training data points is calculated, and the test data point is assigned the class label that most of the K closet training data points have. The KNN algorithm assumes that all the data corresponds to points in the N-dimensional space. Let the test data point x_i be represented by the feature vector $[x_1^i, x_2^i, x_3^i, ..., x_N^i]$, where x_k^i denotes the value of the kth attribute of the test data point x_i, and $x_i^{'}$ is the transpose of x_i. the distance between x_i and x_j is defined as $d(x_i, x_j) = \sqrt{\sum_{k=1}^{N}(x_k^i - x_k^j)^2}$. If the number of training data is n, the n such distances will be calculated, and the closet K training data are identified as neighbors. If K = 1, then the class label of the test data point is equal to the closet training data point. If K > 1, then the class label of the test data point is equal to the class label that most of the neighbors have. If there is a tie, then the tie is resolved arbitrarily.

2.2 Bayes Classifier

The Naive Bayes classifier is based on the simplifying assumption that the attribute values are conditionally independent given target value. In other words, the assumption is that given the target value of the instance, the probability of observing the conjunction $a_1, a_2.....a_n$ is just the product of the probabilities for the individual attributes:

$P(a_1, a_2.....a_n|v_j) = argmax_{vj \epsilon V} \prod_i P(a_i|v_j)$

So, basically the Naive Bayes Classifier ignores the possible dependencies, namely, correlations, among the inputs and reduces a multivariate problem to a group of univariate problems. It is noticed that in a Naive Bayes classifier the number of distinct $P(a_i|v_j)$ terms that must be estimated from the training data is just the number of distinct attribute values times the the number of distinct target values- a much smaller number than if we were estimate the $P(a_1, a_2.....a_n|v_j)$ terms as needed for Bayesian theory.

2.3 Support Vector Machine (SVM)

These models are algorithmic implementations of ideas from statistical learning theory, which concerns itself with the problem of building consistent estimators from data: how can the performance of a model on an unknown data set be estimated, given only characteristics of the model, and performance on a training set? Algorithmically, support vector machines build optimal separating boundaries between data sets by solving a constrained quadratic optimization problem.

By using different kernel functions, varying degrees of non-linearity and flexibility can be included in the model. Because they can be derived from advanced statistical ideas, and bounds on the generalization error can be calculated for them, support vector machines have received considerable research interest over the past years. Performances on par with or exceeding that of other machine learning algorithms have been reported in the medical literature.The disadvantage of support vector machines is that the classification result is purely dichotomous, and no probability of class membership is given.

2.4 Logistic Regression and Artificial Neural Network

These models differ from the three algorithms above in the sense that they both provide a functional form and parameter vector α to express $P(y|x)$ as

$P(y|x) = f(x, \alpha)$.

The parameters α are determined based on the dataset D, usually by maximum-likelihood estimation. As the functional form of f differs for logistic regression and artificial neural nets, the former is known as a parametric method, whereas the latter is sometimes called semi-parametric or non-parametric. This distinction is important because the contribution of parameters in logistic regression (coefficients and intercept) can be interpreted, whereas this is not always the case with the parameters of a neural network (weights).

3 Implementation, Training, Testing

In this section, an overall description of the data set is provided, multiple classifiers including the KNN, logistic regression, Bayes classifier, SVM, and neural networks are implemented, trained using the training data, and tested using the testing data. The initial dataset is cleaned for training purposes, certain attributes are dropped, and others are engineered based on preference. The training, validation and, testing split are done based on research experience, the data is then split 90/10 for the training and test split. The training data is further split into an 80/20 ratio for the training and validation splits.

3.1 The Dataset

The data set consists of features such as League Division, Match Date, Full Time Result, Full Time Home Team Goals, etc. There are a total of 162 such features after successfully combining the data set which results in a larger data set (a combination of 20 data sets. Overall, there are a total of 7303 features on the initial combination of the data sets.

3.2 Preprocessing

What is interesting to note is that python arranges the features according to the number of NaN values in them; therefore, there are a lot of NaN values in the columns that follow the 23rd column. So all the columns after the 23rd one have been dropped. Additionally, the data points that contain NaN values have also been dropped. After this preporcessing the data set we obtain has a dimension of 7260 x 23.

3.3 Feature Engineering

For the feature engineering, all the columns that have string values, namely, "HomeTeam," "AwayTeam," "Refree," "FTR," "HTR" have been encoded with unique integer values, so that the data is in a form that can be used for the application of different machine learning models.

Additionally, features such as Div, and Date have been dropped by mere eyeballing because they are not correlated with the results of a match. Moreover, features such as FTAG and FTHG are also dropped because they represent the same thing as the final score of a match.

The features are also normalized using the standard deviation and the mean of those particular features to remove and unwanted bias from particular features.

3.4 Implementation of Different Models

Three models have been implemented for this portion of the project. These models include: The Naive Bayes Classifier, Logistic Regression, and, The KNN Classifier.

3.4.1 Naive Bayes

The Naive Bayes classifier results in an accuracy of 63% with the validation data and an accuracy of 60% with the test data.

3.4.2 KNN Classification

Moreover, the KNN classifier results in an accuracy of 52% with the validation data and almost the same accuracy with the testing data as well.

3.4.3 Logistic Regression

Moreover, the logistic regression results in an accuracy of 63% with the validation data and 61% acuracy with the testing data.

[0]The code is available at: `https://github.com/aliasghar211/EPL-MachineLearning.git`