Report

Group Number: B1
Lab Group Number: 05

Submitted By:
Ahmed Shahriar Adnan (14.02.04.077)
Biozid Rahman Niloy (14.02.04.070)

1 Problem

An attempt to predict the outcome of the football matches played by Liverpool F.C..

2 Problem Description

Liverpool Football Club is a professional football club in Liverpool, England, which competes in the Premier League, the top tier of English football.
In this problem, we have a data-set containing the performance of this club in previous matches, by studying their previous performance and result we are trying to predict the outcome of the matches such as win,loss or draw.

3 Data-set Description

The data-set consists of 8 columns among which 7 are features and the last column is the target class. The features are data collected from performance showed by Liverpool Club and some features are attributes of the opponents they faced. For the target we assumed 1 for a win, 2 for a loss, 3 for a draw.

Data Description

Variable	**Definition**
TeamRating	Opponent team rating
Shots	Total shots taken by Liverpool
ShotsTarget	Total shots taken on target
PassSuccess	Percentage of successful pass
DribbleCom	Number of successful dribble
Tackles	Number of successful tackles
Possession	Percentage of ball control
Result	Outcome of the match

4 Description of the Models

We have used 6 models in this problem to see which one gives the better accuracy. Score for each model is tested individually.Then it is split for training and testing. 80% data is used for training and the rest 20% is used for testing purpose. As we used K-Fold Cross Validation with k=5 we divided our data set into 5 slices and ran classifier on different slices exactly 5 times and we kept lists to keep the results of the classifier runs. At the very end we used mean function from numpy library to calculate the mean value from the lists .

4.1 K-Nearest Neighbours Classifier

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.
Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.
The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.
In our code we used this classifier with k=5 .

4.2 Decision Tree Classifier

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.
A decision tree has 2 kinds of nodes:

1. Each leaf node has a class label, determined by majority vote of training examples reaching that leaf.
2. Each internal node is a question on features. It branches out according to the answers.

4.3 Random forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees habit of over fitting to their training set.

- Random forest classifier will handle the missing values.

- When we have more trees in the forest, random forest classifier wont over fit the model.
- In the random forest classifier, the higher the number of trees in the forest gives the high accuracy results.

4.4 AdaBoost Classifier

Ada-boost, like Random Forest Classifier is another ensemble classifier. Ensemble classifier are made up of multiple classifier algorithms and whose output is combined result of output of those classifier algorithms.
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.

1. Retrains the algorithm iteratively by choosing the training set based on accuracy of previous training.
2. The weight-age of each trained classifier at any iteration depends on the accuracy achieved.

4.5 SVM Classifier

Support vector machines (SVMs) are a set of supervised learning methods used for classification and regression The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels. The disadvantages of support vector machines include:
- If the number of features is much greater than the number of samples, avoid overfitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

4.6 SGD (Stochastic Gradient Descent) Classifier

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. SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and

natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features.
The advantages of Stochastic Gradient Descent are:

1. Efficiency.

2. Ease of implementation (lots of opportunities for code tuning).

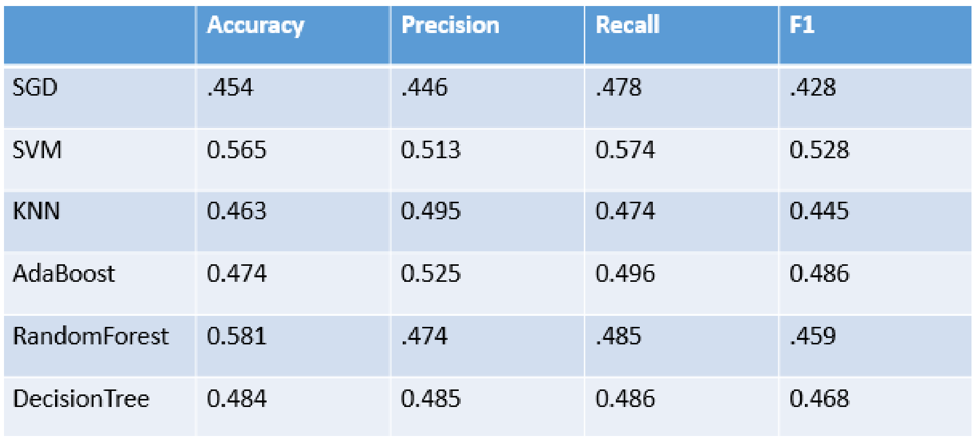
The disadvantages of Stochastic Gradient Descent include:

1. SGD requires a number of hyper parameters such as the regularization parameter and the number of iterations.

2. SGD is sensitive to feature scaling.

5 Comparison of the performance scores

For testing the performance we have chosen 4 performance matrices which are accuracy, precision, recall and F1score. The comparison of the performance scores for each model is shown below:



	Accuracy	Precision	Recall	F1
SGD	.454	.446	.478	.428
SVM	0.565	0.513	0.574	0.528
KNN	0.463	0.495	0.474	0.445
AdaBoost	0.474	0.525	0.496	0.486
RandomForest	0.581	.474	.485	.459
DecisionTree	0.484	0.485	0.486	0.468

6 Discussion

As we can see that the SVM classifier shows most accuracy regarding this data set as SVM can capture much more complex relationships between our data points.On the other classifiers we get low accuracy as we think our feature engineering was not top notch and we have limited amount of data.