

JACOBS UNIVERSITY

SEMESTER PROJECT

# Football Prediction Challenge

*Rahul Bhat*

supervised by  
Dr. Prof. Adalbert F.X. Wilhelm

July 31, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Objective</b>	<b>3</b>
2.1	File Description . . . . .	4
2.2	Data Fields . . . . .	4
<b>3</b>	<b>Importing Data</b>	<b>5</b>
<b>4</b>	<b>Separate into smaller data frames</b>	<b>5</b>
<b>5</b>	<b>Predictive Analytical Techniques</b>	<b>5</b>
5.1	<b>Basic Tree Model</b> . . . . .	5
5.1.1	Regression Trees vs Classification Trees . . . . .	6
5.1.2	Packages . . . . .	7
5.1.3	Cross-Validation To Improve The Model . . . . .	7
5.2	<b>Random Forest</b> . . . . .	7
5.2.1	Properties of Random Forests . . . . .	7
5.2.2	Packages . . . . .	8
5.2.3	How Random Forest Works . . . . .	8
5.3	<b>Neural Networks</b> . . . . .	11
5.3.1	Properties of Neural Networks . . . . .	11
5.3.2	Packages . . . . .	12
5.3.3	How Neural Networks Works . . . . .	12
5.4	<b>Xgboost</b> . . . . .	13
5.4.1	Features . . . . .	13
5.4.2	Parameters . . . . .	13
5.4.3	Packages . . . . .	14
5.4.4	How Xgboost Works . . . . .	14
<b>6</b>	<b>Conclusion</b>	<b>15</b>

## **Abstract**

*Demand for "Big Data" expertise started growing over the last few years and so as the demands for the new tools and the techniques that transform this huge amount of data into useful Knowledge. The data volume is growing at a rapid rate in every industry or organization and this is due to explosive growth of cloud computing, social media, online videos and more. Data Science is gaining traction in every sector and using "Big Data" and "Analytics" allows to find correlations, discover unexpected patterns and predict future outcomes which can be used to forecast future probabilities. Different techniques can be used, including data mining, statistical modeling and machine learning which helps the analysts to forecast the outcome from the given set of data. In this project, the football matches of the last 8 years of the Italian League are provided and using predictive analytical techniques, the final results of football matches are predicted. The provided data is split non-randomly into two data sets, train data and test data. The train data (training dataset) is used for supervised learning, in which the input data is given with the correct or expected output for every data row and the test data (testing dataset) is on which the model is applied.*

**Keywords:** Big Data, Data Science, Forecast, predictive analytics

# 1 Introduction

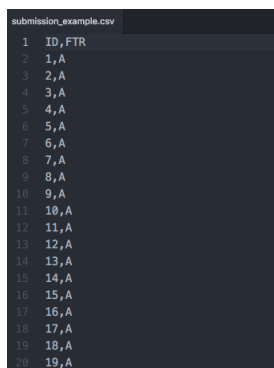
"Data mining is the most important research domain in the 21st century". Data mining is sorting through data and to extract regularities to identify patterns and establish relationships. In data mining, different parameters are used, that are association, sequence or path analysis, classification, clustering, and forecasting. Forecasting in data mining is to collect the data, discover patterns in data and statistical model is formulated which provides the ability to predict the future. This area of data mining is also known as predictive analytics. There are different approaches and techniques that can broadly be grouped into regression techniques and machine learning techniques. [6].

In this paper, we used four different predictive analytical techniques to predict the final result of the football matches that are Basic Tree Model Technique, Random Forest, Neural Networks and Xgboost. Cross-Validation method is also used in Basic Tree Model and Xgboost to improve the model. The working of the four techniques, the packages which are used and the properties are also explained in detail.

The most complicated task in "Data Analytics" is choosing the right language. In "Data Science" two tools are most popular and important that are "R" and "Python". "R" is the most popular tool among data miners and R is often praised for its great features for data visualization, as it was developed by statisticians in mind; plenty of programmers like Python for its simple syntax. In this project, "R" is used to predict the final results of the football matches using predictive analytics.

## 2 Objective

The objective of this project is to predict the final results of the football matches of the Italian League using the odds of various bookmakers, but the final result of the football matches should be in a particular format which is shown below.



ID	FTR
1	A
2	A
3	A
4	A
5	A
6	A
7	A
8	A
9	A
10	A
11	A
12	A
13	A
14	A
15	A
16	A
17	A
18	A
19	A

Fig. 1 submission\_example.csv

Fig. 1 The above image is a final result sample in which the full time result (FTR) of first match which has a ID (1) is away (A).

## 2.1 File Description

train.csv - the training set

test.csv - the testing set

submission\_example.csv - a sample submission file in the correct format

### Format

The format of the file is "CSV" stands for "Common Separated Values" which is often used to exchange and convert data between various spreadsheet programs. There are cells inside such data file, which is separated by a special character, which usually is a comma and other characters can also be used as well. The first row of this data file contains the column names instead of actual data. Here is a sample of the expected format [7].

```
Col1,Col2,Col3
100,a1,b1
200,a2,b2
300,a3,b3
```

### CSV format sample

## 2.2 Data Fields

- ID = An anonymous ID unique to a given match
- Date = Date of the match
- FTR = Full Time Result (H=Home Win, D=Draw, A=Away Win). This is the target variable
- HomeTeam = Home Team
- AwayTeam = Away Team
- B365H = Bet365 home win odds
- B365D = Bet365 draw odds
- B365A = Bet365 away win odds
- BWH = Bet&Win home win odds
- BWD = Bet&Win draw odds
- BWA = Bet&Win away win odds
- IWH = Interwetten home win odds
- IWD = Interwetten draw odds
- IWA = Interwetten away win odds
- LBH = Ladbrokes home win odds

- LBD = Ladbrokes draw odds
- LBA = Ladbrokes away win odds
- VCH = VC Bet home win odds
- VCD = VC Bet draw odds
- VCA = VC Bet away win odds
- WHH = William Hill home win odds
- WHD = William Hill draw odds
- WHA = William Hill away win odds

### 3 Importing Data

Before performing any function, the first step is to import data in R, which is fairly simple. After downloading the data, there is a function called "read.csv" through which the data can be read.

```
train = read.csv("/Users/Bhat/Downloads/train.csv",header = TRUE,sep=",")
test = read.csv("/Users/Bhat/Downloads/test.csv",header = TRUE,sep=",")
```

In the above code, it shows that the data is imported into R where testing data is stored into test and training data is stored into train.

### 4 Separate into smaller data frames

The next step is to separate the data into smaller data frames through which the list of teams and data frames are created for each home team.

#### Data Frames

Data frames are used for storing data tables. Data frame has similar dimensional properties like the matrix but the difference is, it contains categorical data, as well as numeric data. Data frames allow to store the data in rectangular grids and the rows in these rectangular grids correspond to measurements or values of an instance, while each column is a vector containing data for a specific variable.

### 5 Predictive Analytical Techniques

#### 5.1 Basic Tree Model

The first technique which is used to predict the final result is very simple and easy. In this approach, all the teams are ignored and only the odds which are given are considered and then the basic tree model is used to get the final result. The final result of the football matches using tree-based model is named as "submission 1".

## Tree-Based Models

Recursive partitioning is a primary tool in data mining which helps to explore the structure of a dataset. Tree-based methods are simple and very useful for interpretation. Tree-based learning algorithms are one of the best and mostly used supervised learning methods, while developing it helps to visualize decision rules for predicting a categorical (classification tree) or continuous (regression tree) outcome which is also called as CART (classification and regression tree) [9].

### 5.1.1 Regression Trees vs Classification Trees

The terminal nodes (leaves) lies at the bottom of the decision tree. So it means that decision trees are typically drawn upside down such that the leaves are at the bottom and the roots are at the top.

Regression Trees and Classification Trees are almost similar to each other. The primary differences and similarity between them are discussed below.

- The primary objective is to choose the right tree and it is the dependent variable that determines the type of decision tree needed. When the dependent variable is continuous then Regression Trees are used and when the dependent variable is categorical then Classification Trees are used.
- Regression Tree - In training data set the terminal nodes obtain some value and that value is the mean response of observation falling in that region. So if there is an unseen data falls in that region, we make its prediction with mean value.
- Classification Tree - In the training dataset the value obtained by the terminal node is the mode of observations falling in that region. Thus, if an unseen data observation falls in that region, we make its prediction with mode value.
- Classification Tree and Regression Tree divides the independent variables (predictor space) into distinct and non-overlapping regions and these regions are nothing but the high dimensional boxes.
- Classification Tree and Regression Tree follow a top-down greedy approach from a root node which is also called as a recursive binary splitting. In this approach, the data is partitioned into subsets that contain instances with similar values (homogenous). It is a greedy approach because the algorithm only looks for best variable available and only cares about the current split but not about future splits which will lead to a better tree.
- The splitting process continues in both the cases until it reaches the value which is defined by the user.
- The splitting process results in fully grown trees. The fully grown trees sometimes overfit data which leads to poor accuracy on unseen data.
- The technique called pruning is used to tackle overfitting and this technique is performed in order to remove anomalies in the training dataset due to noise or outliers. The pruned trees are smaller and less complex.

### 5.1.2 Packages

`library(Tree)`

The package which is required for tree-based models is "Tree".

### 5.1.3 Cross-Validation To Improve The Model

Cross-validation technique is used to improve our previous model. The purpose of cross-validation is to qualify the model. Cross-validation gives us the better estimate of the performance of our trained model when used on different data and this process can be repeated using different parameters until we are satisfied with the performance. Then it helps us to train the model with the best parameters on the whole data [4].

We used cross-validation in order to determine the optimal level of tree complexity and pruning technique is used to improve the test error rate. After improving our model, we tested that improved model on our testing dataset and got the final result (submission 2).

For cross-validation no extra packages are required.

## 5.2 Random Forest

Random forest is a very popular method for predictive analytics. Random forest is an ensemble of decision trees. It is a type of “ensemble learning” technique for classification, regression and for some other tasks as well which extends on decision trees. Ensemble learning is a procedure which gives a prediction value, in this approach a team of predictive models is constructed to solve the given prediction task [5].

Random Forest grows many classification trees using a subset of the available input variables and their values. Each tree in the forest gives a classification by putting an input vector down step by step to each of the trees in the forest and at every step, a new object is formed from the input vector. It is like tree votes for that class and the forest chooses the one with the maximum number of votes [5].

### 5.2.1 Properties of Random Forests

The properties of Random Forest are listed below.

- Random forest can run multiple trees in parallel, therefore it is easy to train large number of data.
- Random forest works very well on a wide variety of data.
- Random forest is also very effective in eliminating noise in the model input data.
- Random forest can also handle large number of input variables without variable deletion.
- The forests which are generated can be saved in future for some other data.



### 5.2.2 Packages

- library(caret)
- library(ggplot2)
- library(randomForest)

These are the list of packages which are used for "Random Forest".

- caret - Caret is used for for training and plotting classification and regression models.
- ggplot2 - This package is used for plotting and ggplot2 package is based on the grammar of graphics. The main advantage of this packages is, it tries to take the good parts of base and lattice graphics and none of the bad parts.

### 5.2.3 How Random Forest Works

There are twenty-three variables in our training set. The "ID" and "FTR" are our outcome variables and ultimately these two variables we want to predict in the test data. We will make predictions for the ID and FTR, in the end, we will add them together. The rest of the variables are so-called predictors or predictor variables, which we want to use for predicting the outcome. First, we will use "pairs" command (**pairs(train)**) which will plot all the variables of a dataset against each other.

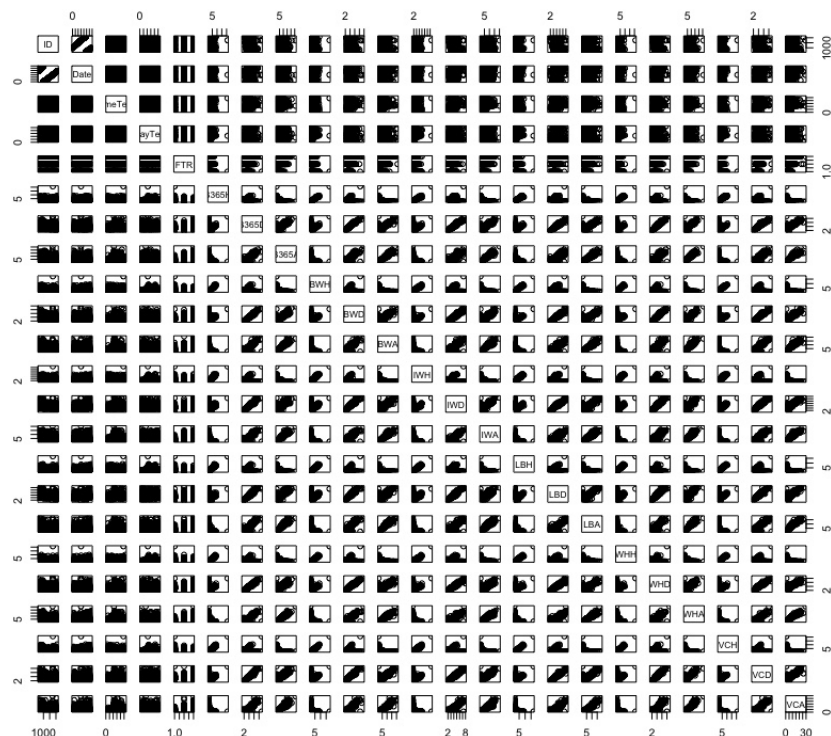


Fig. 2 Scatterplot Matrix

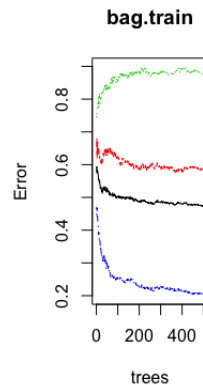
Fig. 2 The above image is the scatterplot matrix of all the variables in the training dataset. Scatterplot matrix helps us to determine a linear correlation between multiple variables. At first glance, we can see that there are some loose connections to the "FTR" variables, but there is no variable in the graph which shows a nice linear connection.

We have to take care that any manipulation of the training data has to be done in the exact same manner to the testing data. The reason is that the machine learning algorithm expects to find the same data structure used for learning when it makes predictions.

So our goal here is to have a function that performs the same operations on the training dataset and the testing dataset. The function which we will use takes in a data frame and returns a data frame.

## Decision Trees

The code we will write, we have to put that code into this function for building Random Forest model. After that, we will build decision trees using Random Forest.



**Fig. 3 Decision Tree**

Fig. 3 The above figure is the decision tree or a forest which is built using the Random Forest algorithm based learning. In the above figure, there are 500 trees with some error rate. The formula which we used to built the decision tree is :

```
bag.train = randomForest(FTR ~ID-Date-HomeTeam-AwayTeam,data = train,importance =
TRUE,na.action = na.omit)
```

In this formula, we have used four variables to get the full time result (FTR) and the four variables are ID, Date, HomeTeam, and AwayTeam which are taken from our training dataset. The four different color lines in the above figure are these four variables with their error rate. There are some missing values in the data which we have to be omitted, otherwise, warnings will present (na.action = na.omit).

## Confusion Matrix

Now we will use confusion matrix function from caret package which can be used for creating confusion matrix based on actual response variable and predicted value.

Below you can see the confusion matrix.

```
Call:
randomForest(formula = FTR ~ . - ID - Date - HomeTeam - AwayTeam,      da
              ta = train, importance = TRUE, na.action = na.omit)
              Type of random forest: classification
              Number of trees: 500
              No. of variables tried at each split: 4

              OOB estimate of  error rate: 48.92%
Confusion matrix:
      A  D  H class.error
A 151 51 190  0.6147959
D  92 41 237  0.8891892
H  93 62 565  0.2152778
```

**Fig. 4 Confusion Matrix**

Fig. 4 In the above image you can see the formula in which we will get the FTR (full time result) using these four variables : ID, Date, HomeTeam, and AwayTeam. It is also shown that which type of random forest it is, how many numbers of trees are there and the number of variables tried at each split. The estimated error rate is also given which is 48.92%.

## Histogram



**Fig. 5 Histogram**

Fig. 5 Above you can see the histogram of HomeTeam and AwayTeam in which the full time result (FTR) is shown in three different color dots : red dot for away, the green dot for the draw and the blue dot for home. This result is taken from the training dataset.

The results have been saved and after learning, our machine can make predictions from these results. We will use generic predict function and we will apply them on our testing dataset. R will then take the new data frame, process the variables according to the Random Forest formula and give out a result for each row of the new data frame. We will save the final results in a new column(submission3).

## 5.3 Neural Networks

- Neural Networks is the machine learning technique.
- Neural networks are non-linear statistical data modeling tools.
- Neural networks are the artificial systems which are sophisticated, perhaps intelligent.
- Neural networks perform the computations same as the human brain routinely performs, and thereby possibly enhance understanding of the human brain.
- Neural networks are very good in recognizing patterns and good at fitting non-linear functions.
- Neural networks learn from examples, so there is no need to describe the problems that is why there is no need for a programmer.

Neural networks need training sessions in which it adapts itself based on examples. After completion of the training sessions, the neural computer is able to relate the problem data to the solutions and then it offers a viable solution to a brand new problem. Neural networks can also generalize and handle incomplete data. Their ability to learn by example makes them very flexible and powerful. There is no need to devise an algorithm in order to perform a specific task [8].

Neural networks are tools that have application in many areas and that is why neural networks are used in the aerospace, automotive, banking, defense, electronics, entertainment, financial, insurance, manufacturing, oil and gas, robotics, telecommunications, and transportation industries [2].

### 5.3.1 Properties of Neural Networks

Below are some of the properties of neural networks.

- Neural networks are asynchronous, parallel and computation is collective.
- Memory is distributed and internalized.
- Neural networks are fault tolerant and redundant.
- Neural networks have dynamic connectivity.

### 5.3.2 Packages

- `library(nnet)`
- `library(neuralnet)`

The packages which are required for neural networks are "nnet" and "neuralnet". Package "nnet" is used to generate a class indicator function from a given factor. The "neuralnet" package is used to visualize the generated model and shows the found weights.

### 5.3.3 How Neural Networks Works

- Neural networks get trained using training data and then we test our trained model on testing data.
- First, we have to load all the package and the datasets (testing dataset and training dataset).
- The training dataset contains information on the football matches played between two teams and the final results of the matches are given which are based on the odds of various bookmakers.
- Our goal is to devise a model which predicts the final result of the football matches given in testing data.
- Now we'll build a neural network with hidden nodes (a neural network is comprised of an input, hidden and output nodes). The number of nodes which are chosen here, is without a clear method, however, there are some rules of thumb.
- The output which we will get wouldn't be linear, so we have to use a threshold value. The neuralnet package uses resilient backpropagation with weight backtracking as its standard algorithm.
- Once the neural network is trained, we are ready to test it on our testing data. The compute function is applied for computing the outputs which will give us the final result and that is ID and FTR (submission4).

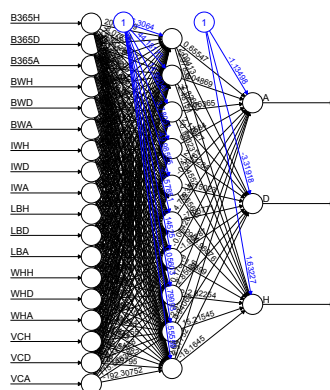


Fig. 6 Neural Network Graph

Fig. 6 The graph shown above is the neural network graph which is taken from the training dataset. In this graph, the final result of the football matches is shown as "A" as away, "D" as draw and "H" as home, using the odds of various bookmakers, which are shown on the left-hand side. There are eighteen odds given, on the basis of these odds we have to find out the final result between two teams.

## 5.4 Xgboost

Xgboost is short for eXtreme Gradient Boosting package. Xgboost is a library which is designed and optimized for boosting tree algorithms. Extreme Gradient Boosting (xgboost) is same as gradient boosting framework but Xgboost is more efficient, flexible and portable. It is the package which is used to solve data-science problems which include both linear model solver and tree learning algorithms. [1].

### 5.4.1 Features

- Xgboost has several features but the most important feature of this package is, it can automatically do parallel computation on a single machine which could be more than 10 times faster than existing gradient boosting packages. So that's why xgboost is able to utilize the more computational power and get a more accurate prediction.
- Xgboost supports various objective functions, including regression, classification, and ranking.
- Xgboost can take several types of input data, for example, dense matrix, sparse matrix, data file (local data files) and xgb.DMatrix, which is nothing but xgboost's own class and it speeds up Xgboost as well.
- Xgboost can also do cross validation and can be used to find important variables.
- Xgboost has better performance on large number of different datasets.

### 5.4.2 Parameters

There are various parameters used in Xgboost model : general parameters, booster parameters, and task parameters.

- General parameters are used to guide the overall functioning, it refers to which booster is used for boosting and the commonly used are the tree or linear model.
- Booster parameters depends on the booster which has been chosen.
- Task parameters are used to guide the optimization performed and task parameters that decide on the learning scenario, for example, different tasks may use different parameters.

These parameters are used to improve the model and for that parameter tuning is must.

### 5.4.3 Packages

- `library(caret)`
- `library(corrplot)`
- `library(Rtsne)`
- `library(xgboost)`
- `library(knitr)`
- `library(ggplot2)`

Above are the list of packages which we have used for "xgboost".

- `caret` - Caret is used for for training and plotting classification and regression models.
- `corrplot` - This package is used to display graphs of a corelation matrix and it is good at choosing colors, labels, layouts, etc.
- `Rtsne` - Rtsne package is used for constructing a low dimensional embedding of high-dimensional data, distances and similarities.
- `knitr` - This package has more flexible design and new features which are used in caching and finer control of graphics.
- `ggplot2` - This package is used for plotting. This package is also used for sophisticated multidimensional conditioning system and a consistent interface to map data to aesthetic attributes.

### 5.4.4 How Xgboost Works

- Xgboost only works with numeric vectors. Therefore, we need to convert all other forms of data into numeric vectors. After converting the data we will use the parameters.

#### Code Sample

```
train.xg = as.matrix(train.xg)
mode(train.xg) = 'numeric'
```

In the above code, we have changed the dataset into numeric.

- Each column in a dataset represents a feature measured by an integer and we know that the first column (ID) doesn't contain any useful information. To let the algorithm focus on real stuff, we will delete it and make that "NULL".
- In the next step, we have to extract the labels from the dataset. We have two files and that is testing dataset and training dataset. We know that the training file contains the class we are looking for. Usually, the labels are in the first column or in the last column and we already know what is in the first column.
- Xgboost doesn't support anything else than numbers. So we will convert classes to integers. To do so we have to first extract the target column and replace all NA's with 0.

- Before learning, we will use the cross validation model to evaluate our error rate.
- The main idea behind Xgboost is, it divides the training data into parts and then Xgboost will retain the first part and use it as the test data. After this, it will reintegrate the first part to the training dataset and retain the second part and it goes on.
- After using the parameters, model training, and cross-validation technique, the error rate will be very low on the test dataset.
- Finally, we are ready to train the real model.
- This will give the final result and that is ID and FTR. The final result is named as "submission5"

## 6 Conclusion

Predictive analytics is the technique to determine patterns and predict future outcomes and trends from the existing datasets. In predictive analytics, number of advanced techniques are used, which helps us to make future forecasts. We used four different techniques to predict the final result of the football matches. Tree-based model and a random forest are simple yet relatively accurate. Neural networks have a kind of universality. The neural network can be used anywhere, no matter what function we want to compute, we know that there is a neural network which can do the job. Xgboost is highly efficient, flexible and portable. The most important feature of Xgboost is it can automatically do parallel computation on a single machine. If there is a large number of data then Xgboost will give the better performance. Data visualization is the fastest and most useful way to learn and understand more about the data, which can yield better results. So more plots and graphs can be constructed to understand the data in depth. Some other techniques can also be used to predict the final result and other parameters like goals, weather, injuries can be added to improve the accuracy.

## References

- [1] Tianqi Chen and Tong He. xgboost: extreme gradient boosting. *R package version 0.4-2*, 2015.
- [2] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. PWS publishing company Boston, 1996.
- [3] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. 2003.
- [4] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.
- [5] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.



- [6] Neelamadhab Padhy and Rasmita Panigrahi. Data mining: A prediction technique for the workers in the pr department of orissa (block and panchayat). *arXiv preprint arXiv:1211.5724*, 2012.
- [7] Yakov Shafranovich. Common format and mime type for comma-separated values (csv) files. 2005.
- [8] Yashpal Singh and Alok Singh Chauhan. Neural networks in data mining. *Journal of Theoretical and Applied Information Technology*, 5(6):36–42, 2009.
- [9] Leland Wilkinson. Classification and regression trees. *Systat*, 11:35–56, 2004.