# Machine Learning Engineer Nanodegree

## Capstone Project

Matheus Vilachã Ferreira Pires

May 25th, 2018

## Using Machine Learning to Predict NBA Games Winners

## I. Definition

### Project Overview

The National Basketball Association (NBA) is the major men's professional basketball league in North America. It was founded in 1946 and has, today, 30 teams. Its players are the world best paid athletes by average annual salary per player. A lot of data exists based on NBA, which is useful for simulations, to analyze teams and players performances, to assist coaches and their staff, and even to predict results.

My intention with this project is to use both unsupervised and supervised learning techniques to predict NBA results. The dataset I will use contains the following information about the two teams (home and visitor): stats from the last game, the average of the stats from the last two games, and the average of the stats from the last five games.

The dataset and all the code that were necessary to execute this project can be found in the project repository at https://github.com/vilacham/capstone_report. The dataset used in this project are the files *capstone_database.xlsx* and *capstone_database_2016_2017.xlsx* in that repository.

### Problem Statement

Since it is a complex job to analyze and predict results based only on the large amount of data about NBA that exists, the main objective of this project is to train a machine to achieve a reasonable prediction rate on stating whether the home team or the road team will win a NBA game using exclusively Machine Learning. Besides that, I also want to determine which of the three possible datasets (the one with last game stats, the one with the average of the last two games or the one with the average of the last five games) can predict more correctly a NBA game result.

I will use unsupervised learning to try to group the most relevant features on my dataset and reduce the curse of dimensionality over it, and supervised learning algorithms to predict whether the home team or the visitor team will win the game.

My objective is to create a Majority Vote Classifier to predict the winner of a game. The prediction of this classifier will be the same as the majority of a group of another five classifiers: a logistic regression classifier, a decision tree classifier, a nearest neighbors classifier, a naïve Bayes classifier and a multi-layer perceptron classifier.

### Metrics

To evaluate the models, I will use accuracy and confusion matrix to determine the best trained model.

Accuracy is calculated dividing the sum of correct predictions by the total number of predictions (it is the percentage of correct predictions). With this metric, I plan to compare the performance of the model in the three different datasets and determine which is the best.

The confusion matrix will be used to identify true positives, true negatives, false positives and false negatives. In layman`s terms, I intend to determine if a model has difficulty to identify when the home team will win, when the home team will lose, when the visitor team will win or when the visitor team will lose.

## II. Analysis

### Data Exploration

The dataset for this project was obtained using the Sports Data Query Language in the killersports.com website and contains games from 2002 to 2016. The code used to extract it can be found in the repository of the project.

Initially, the dataset contained 36154 samples. Each game was considered twice, so I had to exclude half of the samples. Since one of my goals is to compare the performance of the model in three different datasets, I excluded every sample that had non-numeric values in any of the attributes. After all these and other necessary preprocessing steps, the dataset ended up with 16926 samples. In other to make the model more reliable, I opted to remove outliers (I considered outliers data points which were 1.5 lower than the 1st quartile or 1.5 higher than the 3rd quartile in any of the numerical features). The final dataset has 13712 samples.

The 2016-2017 regular season dataset, which will be used in this project as test dataset, initially contained 2460 samples. After the necessary preprocessing steps, it ended up with 1151 samples.

It contains the following stats (continuous variables) from the last game, the last two games (average) and the last five games (average), for both the home and the visitor team:

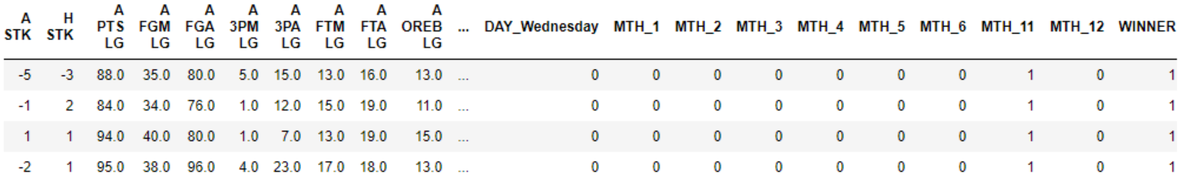
- streak wins;
- points made;
- field goals made;

- field goals attempted;
- three points shots made;
- three points shots attempted;
- free throws made;
- free throws attempted;
- offensive rebounds;
- defensive rebounds;
- rebounds;
- assists;
- turnovers;
- steals; and
- blocks.

The dataset also contains the following features (discrete variables):

- number of overtimes (if any);
- week day;
- month; and
- playoffs (dummy variable).

The first lines of the dataset look like the following:



| A STK | H STK | A PTS LG | A FGM LG | A FGA LG | A 3PM LG | A 3PA LG | A FTM LG | A FTA LG | A OREB LG | ... | DAY_Wednesday | MTH_1 | MTH_2 | MTH_3 | MTH_4 | MTH_5 | MTH_6 | MTH_11 | MTH_12 | WINNER |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -5 | -3 | 88.0 | 35.0 | 80.0 | 5.0 | 15.0 | 13.0 | 16.0 | 13.0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| -1 | 2 | 84.0 | 34.0 | 76.0 | 1.0 | 12.0 | 15.0 | 19.0 | 11.0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 94.0 | 40.0 | 80.0 | 1.0 | 7.0 | 13.0 | 19.0 | 15.0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| -2 | 1 | 95.0 | 38.0 | 96.0 | 4.0 | 23.0 | 17.0 | 18.0 | 13.0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

The first observations I will make are:

I. the away team makes an average of 96.89 points per game (with a standard deviation of 12.20 points);

II. the home team makes an average of 100.11 points per game (with a standard deviation of 12.23 points);

III. the home team won 60.25% of the matches in the dataset.

It is important to note, at this point, that the target column contains more than 60% of one label (home team victory) and less than 40% of the other (away team victory). I will consider this when splitting the dataset into training and testing sets, so it can create a balance and avoid distortion in the results.

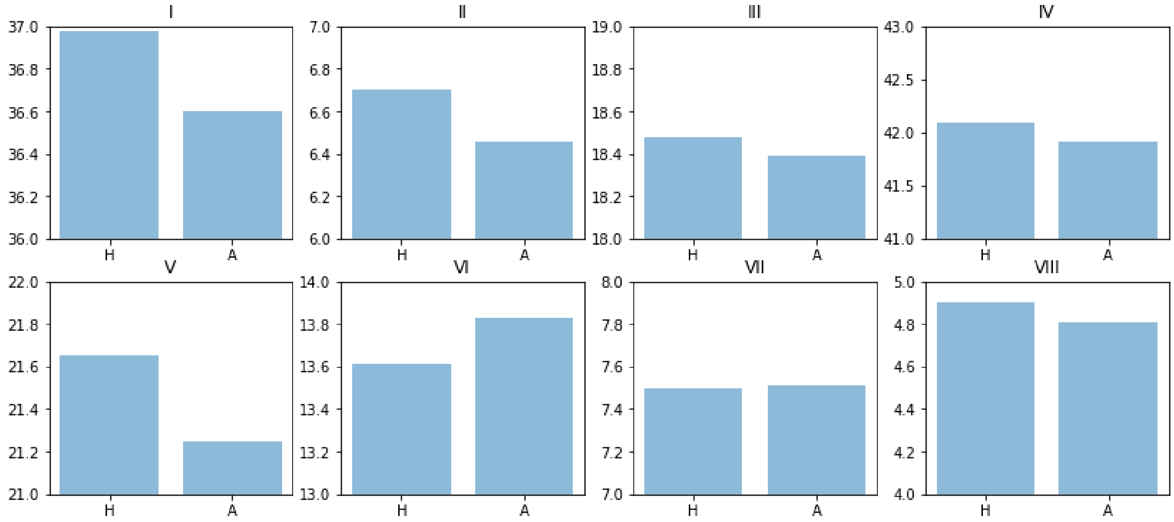
## Exploratory Visualization

One of my objectives is to determine if the performance of a team in the last match (or matches) can be used to predict if it will win a match; another one is to determine the best dataset to predict the winner of a NBA game (the one with last game stats, the one with last two games stats, or the one with the last five games stats). That said, I will divide this section in three parts to analyze the most relevant data from the last game, from the last two games and from the last five games.

From the points discussed in the sub-sections below, it is possible to note that the winner team has better stats than the loser team. This suggests that the team that played better in the last game, in the last two games or even on the last five games may be the favorite to win.

### a) Data from the last game

When the home team won:



I. the home team converted an average of 36.98 field goals (with a standard deviation of 5.15) in its last game, while the away team converted an average of 36.60 field goals (with a standard deviation of 5.18) in its last game;

II. the home team converted an average of 6.70 three points shots (with a standard deviation of 3.27) in its last game, while the away team converted an average of 6.46 three points shots (with a standard deviation of 3.22) in its last game;

III. the home team converted an average of 18.48 free throws (with a standard deviation of 6.29) in its last game, while the away team converted an average of 18.39 free throws (with a standard deviation of 6.35) in its last game;

IV. the home team had an average of 42.09 rebounds (with a standard deviation of 6.42) in its last game, while the away team had an average of 41.92 (with a standard deviation of 6.49) in its last game;

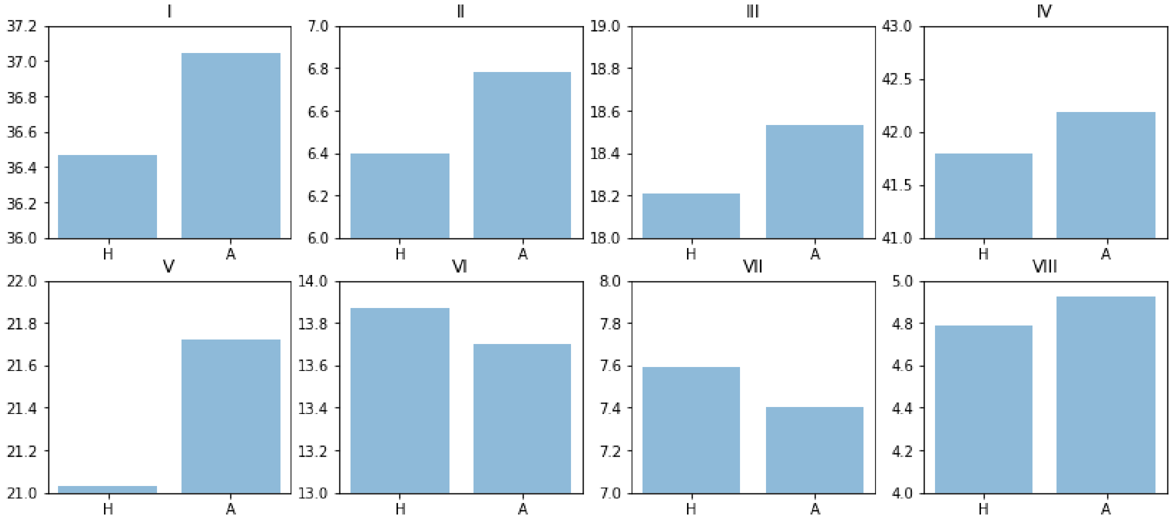
V. the home team counted an average of 21.65 assists (with a standard deviation of 5.13) in its last game, while the away team counted an average of 21.25 assists (with a standard deviation of 5.10) in its last game;

VI. the home team had an average of 13.61 turnovers (with a standard deviation of 3.80) in its last game, while the away team had an average of 13.83 turnovers (with a standard deviation of 3.89) in its last game;

VII. the home team had an average of 7.50 steals (with a standard deviation of 2.93) in its last game, while the away team had an average of 7.51 steals (with a standard deviation of 2.89) in its last game;

VIII. the home team had an average of 4.90 blocks (with a standard deviation of 2.58) in its last game, while the away team had an average of 4.81 (with a standard deviation of 2.57) in its last game.

When the away team won:



I. the away team converted an average of 37.04 field goals (with a standard deviation of 5.17) in its last game, while the home team converted an average of 36.47 field goals (with a standard deviation of 5.04) in its last game;

II. the away team converted an average of 6.78 three points shots (with a standard deviation of 3.29) in its last game, while the home team converted an average of 6.40 three points shots (with a standard deviation of 3.22) in its last game;

III. the away team converted an average of 18.53 free throws (with a standard deviation of 6.17) in its last game, while the home team converted an average of 18.21 free throws (with a standard deviation of 6.25) in its last game;

IV. the away team had an average of 42.18 rebounds (with a standard deviation of 6.53) in its last game, while the home team had an average of 41.79 (with a standard deviation of 6.51) in its last game;

V. the away team counted an average of 21.72 assists (with a standard deviation of 5.13) in its last game, while the home team counted an average of 21.03 assists (with a standard deviation of 4.85) in its last game;

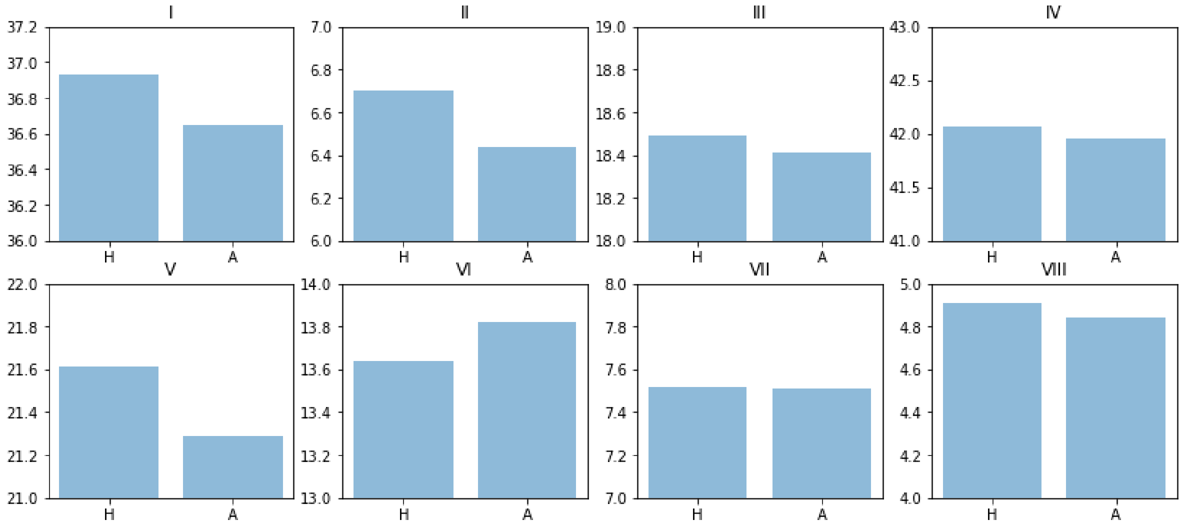
VI. the away team had an average of 13.70 turnovers (with a standard deviation of 3.84) in its last game, while the home team had an average of 13.87 turnovers (with a standard deviation of 3.81) in its last game;

VII. the away team had an average of 7.59 steals (with a standard deviation of 2.89) in its last game, while the home team had an average of 7.40 steals (with a standard deviation of 2.84) in its last game;

VIII. the away team had an average of 4.92 blocks (with a standard deviation of 2.58) in its last game, while the home team had an average of 4.79 (with a standard deviation of 2.58) in its last game.

### b) Data from the last two games

When the home team won:



I. the home team converted an average of 36.93 field goals (with a standard deviation of 3.92) in its last two games, while the away team converted an average of 36.65 field goals (with a standard deviation of 3.89) in its last two games;

II. the home team converted an average of 6.70 three points shots (with a standard deviation of 2.69) in its last two games, while the away team converted an average of 6.44 three points shots (with a standard deviation of 2.63) in its last two games;

III. the home team converted an average of 18.49 free throws (with a standard deviation of 4.68) in its last two games, while the away team converted an average of 18.41 free throws (with a standard deviation of 4.72) in its last two games;

IV. the home team had an average of 42.07 rebounds (with a standard deviation of 4.82) in its last two games, while the away team had an average of 41.96 (with a standard deviation of 4.78) in its last two games;

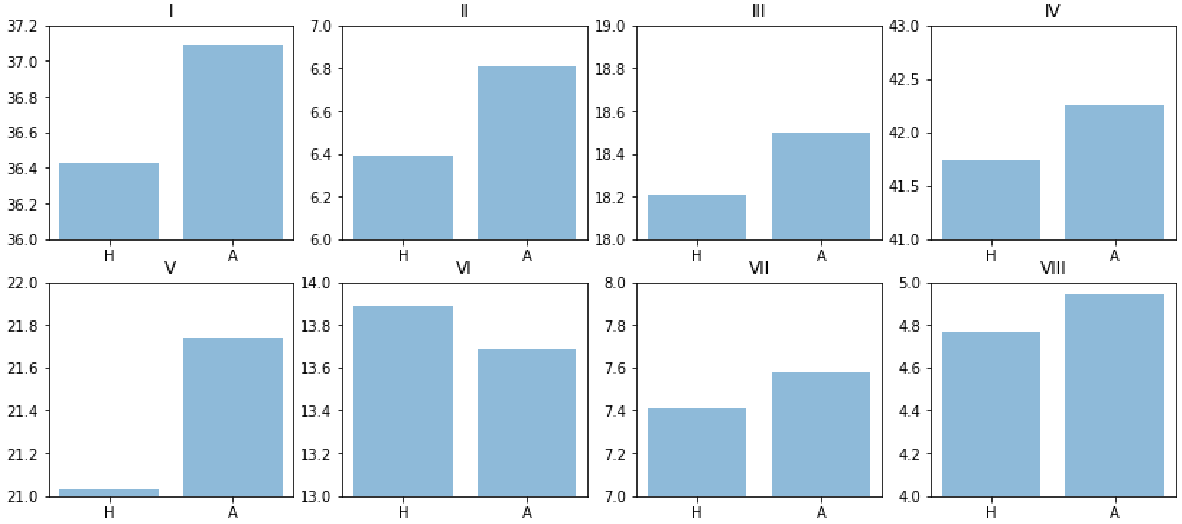
V. the home team counted an average of 21.61 assists (with a standard deviation of 3.89) in its last two games, while the away team counted an average of 21.29 assists (with a standard deviation of 3.83) in its last two games;

VI. the home team had an average of 13.64 turnovers (with a standard deviation of 2.84) in its last two games, while the away team had an average of 13.82 turnovers (with a standard deviation of 2.86) in its last two games;

VII. the home team had an average of 7.52 steals (with a standard deviation of 2.17) in its last two games, while the away team had an average of 7.51 steals (with a standard deviation of 2.14) in its last two games;

VIII. the home team had an average of 4.91 blocks (with a standard deviation of 1.92) in its last two games, while the away team had an average of 4.81 (with a standard deviation of 1.93) in its last two games.

When the away team won:



I. the away team converted an average of 37.09 field goals (with a standard deviation of 3.95) in its last two games, while the home team converted an average of 36.43 field goals (with a standard deviation of 3.84) in its last two games;

II. the away team converted an average of 6.81 three points shots (with a standard deviation of 2.71) in its last two games, while the home team converted an average of 6.39 three points shots (with a standard deviation of 2.59) in its last two games;

III. the away team converted an average of 18.50 free throws (with a standard deviation of 4.66) in its last two games, while the home team converted an average of 18.21 free throws (with a standard deviation of 4.70) in its last two games;

IV. the away team had an average of 42.25 rebounds (with a standard deviation of 4.82) in its last two games, while the home team had an average of 41.74 (with a standard deviation of 4.77) in its last two games;

V. the away team counted an average of 21.74 assists (with a standard deviation of 3.95) in its last two games, while the home team counted an average of 21.03 assists (with a standard deviation of 3.71) in its last two games;

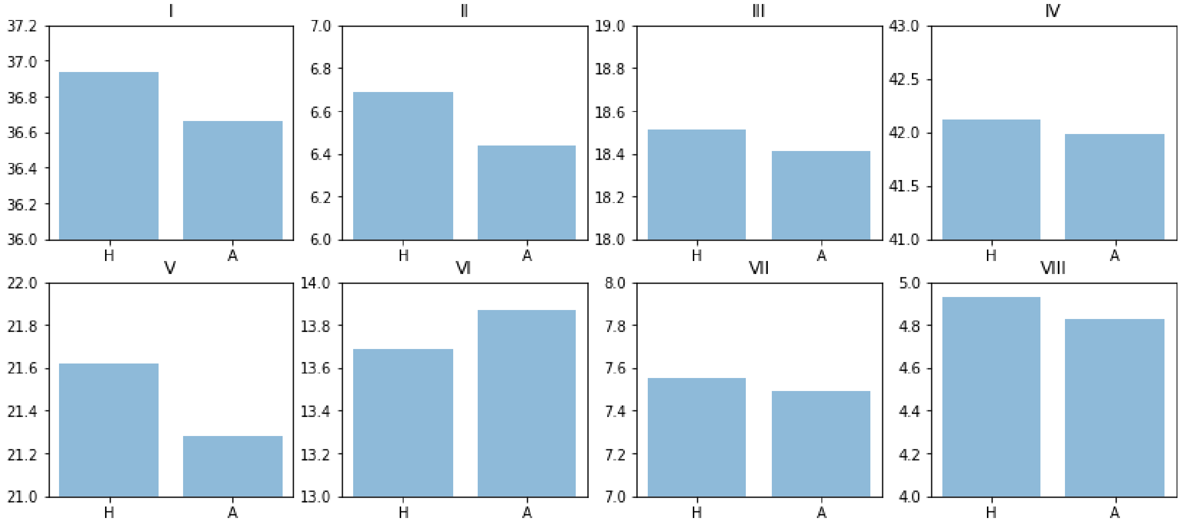
VI. the away team had an average of 13.69 turnovers (with a standard deviation of 2.83) in its last two games, while the home team had an average of 13.89 turnovers (with a standard deviation of 2.81) in its last two games;

VII. the away team had an average of 7.58 steals (with a standard deviation of 2.15) in its last two games, while the home team had an average of 7.41 steals (with a standard deviation of 2.12) in its last two games;

VIII. the away team had an average of 4.94 blocks (with a standard deviation of 1.95) in its last two games, while the home team had an average of 4.77 (with a standard deviation of 1.92) in its last two games.

### c) Data from the last five games

When the home team won:



I. the home team converted an average of 36.94 field goals (with a standard deviation of 2.90) in its last two games, while the away team converted an average of 36.66 field goals (with a standard deviation of 2.87) in its last two games;

II. the home team converted an average of 6.69 three points shots (with a standard deviation of 2.24) in its last two games, while the away team converted an average of 6.44 three points shots (with a standard deviation of 2.20) in its last two games;

III. the home team converted an average of 18.51 free throws (with a standard deviation of 3.38) in its last two games, while the away team converted an average of 18.41 free throws (with a standard deviation of 3.42) in its last two games;

IV. the home team had an average of 42.12 rebounds (with a standard deviation of 3.38) in its last two games, while the away team had an average of 41.99 (with a standard deviation of 3.36) in its last two games;

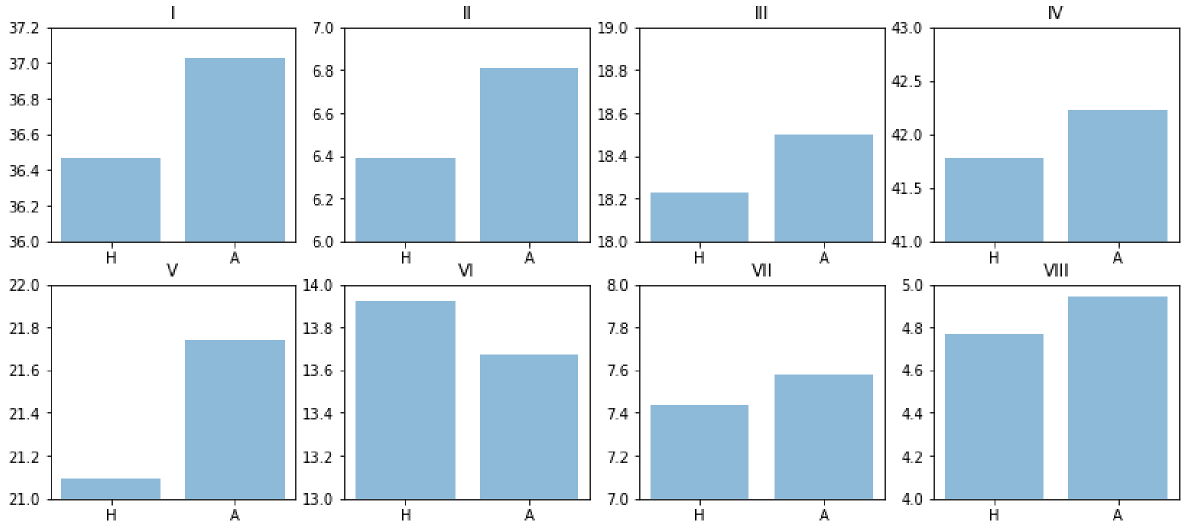
V. the home team counted an average of 21.62 assists (with a standard deviation of 2.89) in its last two games, while the away team counted an average of 21.28 assists (with a standard deviation of 2.80) in its last two games;

VI. the home team had an average of 13.69 turnovers (with a standard deviation of 2.03) in its last two games, while the away team had an average of 13.87 turnovers (with a standard deviation of 2.05) in its last two games;

VII. the home team had an average of 7.55 steals (with a standard deviation of 1.55) in its last two games, while the away team had an average of 7.49 steals (with a standard deviation of 1.56) in its last two games;

VIII. the home team had an average of 4.93 blocks (with a standard deviation of 1.38) in its last two games, while the away team had an average of 4.83 (with a standard deviation of 1.39) in its last two games.

When the away team won:



I. the away team converted an average of 37.03 field goals (with a standard deviation of 2.93) in its last two games, while the home team converted an average of 36.47 field goals (with a standard deviation of 2.83) in its last two games;

II. the away team converted an average of 6.81 three points shots (with a standard deviation of 2.25) in its last two games, while the home team converted an average of 6.39 three points shots (with a standard deviation of 2.16) in its last two games;

III. the away team converted an average of 18.50 free throws (with a standard deviation of 3.35) in its last two games, while the home team converted an average of 18.23 free throws (with a standard deviation of 3.34) in its last two games;

IV. the away team had an average of 42.22 rebounds (with a standard deviation of 3.38) in its last two games, while the home team had an average of 41.78 (with a standard deviation of 3.35) in its last two games;

V. the away team counted an average of 21.74 assists (with a standard deviation of 2.94) in its last two games, while the home team counted an average of 21.09 assists (with a standard deviation of 2.76) in its last two games;

VI. the away team had an average of 13.67 turnovers (with a standard deviation of 2.00) in its last two games, while the home team had an average of 13.92 turnovers (with a standard deviation of 2.01) in its last two games;

VII. the away team had an average of 7.58 steals (with a standard deviation of 1.53) in its last two games, while the home team had an average of 7.44 steals (with a standard deviation of 1.53) in its last two games;

VIII. the away team had an average of 4.94 blocks (with a standard deviation of 1.40) in its last two games, while the home team had an average of 4.77 (with a standard deviation of 1.40) in its last two games.

## Algorithms and Techniques

I will first use unsupervised learning to find hidden information in the dataset and reduce its dimensionality. To do so, I wrote code to get the *n* principal components (using Principal Component Analysis technique) and another one to reduce both the training and the testing sets to this *n* components. This first step has the main objective of reducing the computational cost of running supervised learning algorithms and making it faster.

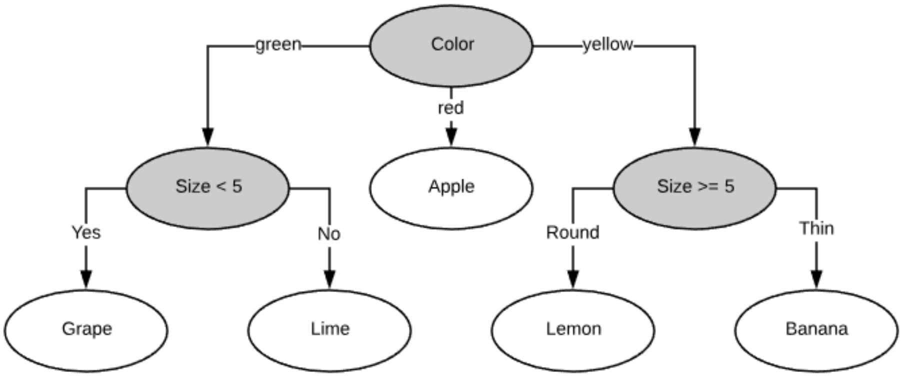
Principal Components Analysis, or PCA, is an unsupervised linear transformation technique used for dimensionality reduction (it identifies patterns in the data based on the correlation between features to reduce the number of features in the dataset). In brief, PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions that the original one. In this project, my goal is to use the *n* components that explains at least 60% of the variance in the dataset.

The next step is to use supervised learning to predict the results. I wrote code to create a majority vote classifier that predicts the result of a game based in the predictions of another five optimized classifiers:

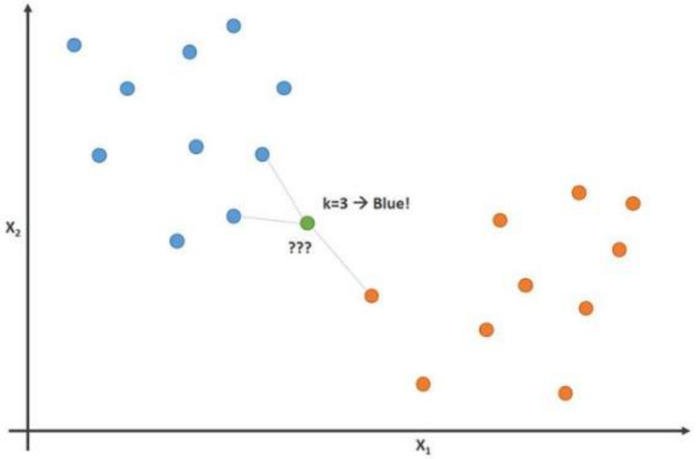
- Logistic regression classifier: a linear model that uses a logistic function to predict categorical classes, estimating the probability of each output value based on one or more features. Logistic regression algorithm can be regularized to avoid overfitting, but on the other hand it tends to underperform if there are a lot of non-linear decision boundaries. A standard logistic function is a “S” shape curve with the following equation:

$$f(x) = \frac{1}{1 + e^{-x}}$$

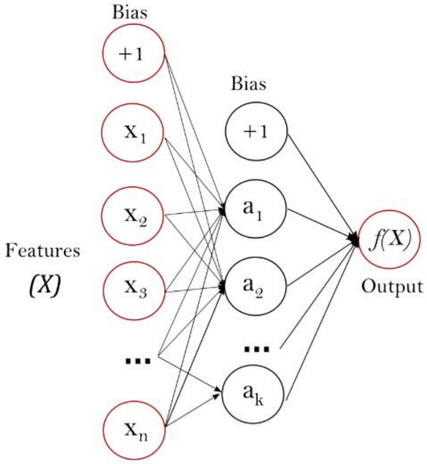
- Decision tree classifier: a non-parametric supervised learning method, used for both classification and regression, that aims to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Although they are simple to understand and interpret, a small change in the data can lead to a large change in the structure if the decision tree. The following image is a simple example of how a decision tree works:



- K nearest neighbors classifier: a type of instance-based learning that simply stores instances of the training data and uses it to classify with a simple majority vote of the nearest neighbors of each point. A query point is assigned the data class which has the most representatives within the *k* nearest neighbors of the point. The next figure is a simple example of how this classifier works:



- Multi-layer Perceptron classifier: a deep neural network that can learn a non-linear function approximator, for both classification and regression, using backpropagation to adjust weights and an optimizing algorithm to make its results more reliable. The next figure is a one hidden layer MLP:

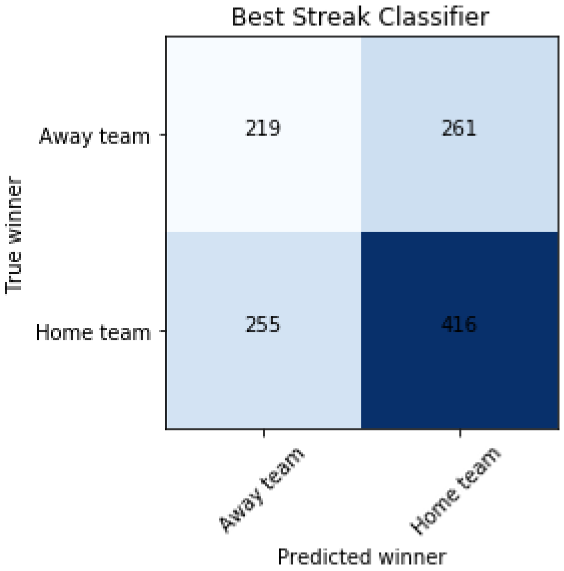


The majority vote classifier will predict a home team's victory if most of the classifiers above also predicts a home team's victory; otherwise, it will predict a visitor team's victory.

## Benchmark

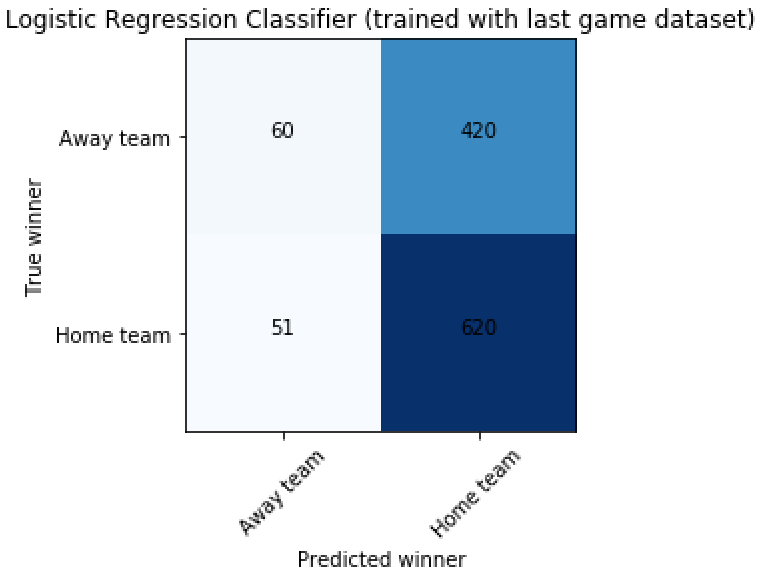
I will use the majority vote classifier to predict the results from the 2016-2017 regular season which is not in the dataset (as mentioned before, the data from this season will be used as testing set), and after that compare the results of this classifier with the results from that season. Besides that, I will use as benchmark models a logistic regression model (untuned and untouched) and a naïve assumption that the winner team will be the one with the highest number of streak wins and, in the case of tie, the home team will be predicted as the winner (this classification will be made by a best streak classifier).

The use of the best streak classifier results in an accuracy of 55.2% and in the following confusion matrix:



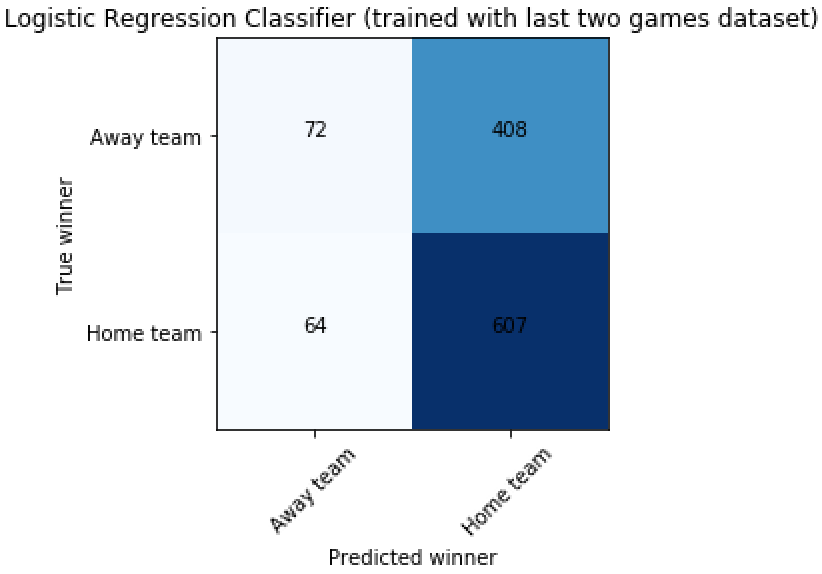
The confusion matrix above suggests that this classifier has difficulty to correctly predict the away team as the winner of a match (it predicted correctly 19.0% of the matches which the road team won) and is relatively reliable to correctly predict the home team as the winner (it predicted correctly 36.1% of the matches which the home team won).

If the classification is based on the performance of both the home and away team in their last game, the use of an untuned and untouched logistic regression classifier results in an accuracy of 59.1% and in the following confusion matrix:



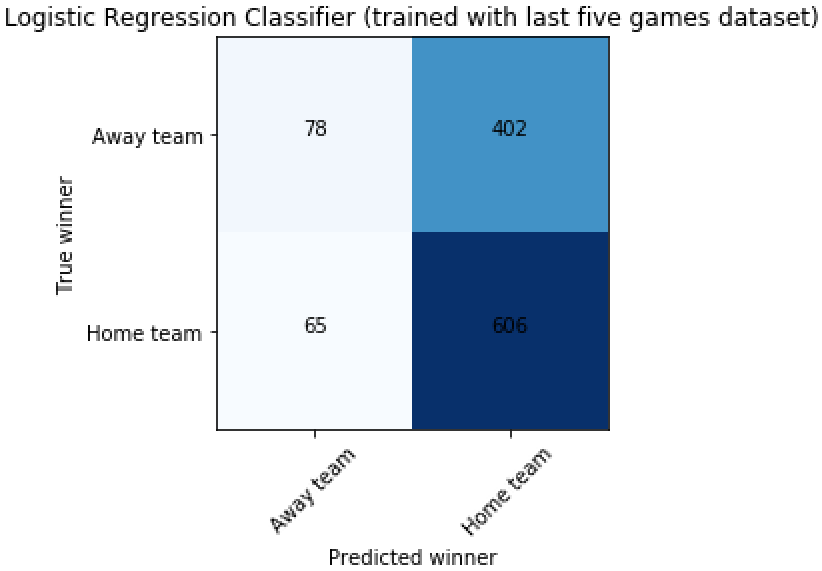
The confusion matrix shows that this classifier, with this dataset, has more difficulty to predict the away team as the winner (it correctly predicts the away team as the winner in only 5.2% of the matches) and is more reliable to predict the home team as the winner (it correctly predicts the home team as the winner in 53.9% of the matches).

If the classification is based on the performance of both the home and away team in their last two games, the use of an untuned and untouched logistic regression classifier results in an accuracy of 58.9% and in the following confusion matrix:



The confusion matrix above shows that this is the less efficient classifier, correctly predicting the home team as the winner in 52.7% of the matches. This is, though, a more efficient classifier than the last analyzed to predict the away team as the winner: it correctly predicts the away team as the winner in 6.1% of the matches.

If the classification is based on the performance of both the home and away team in their last five games, the use of an untuned and untouched logistic regression classifier results in an accuracy of 59.4% and in the following confusion matrix:



From the confusion matrices above, it is possible to conclude that the more games we use to create the dataset, the more efficient the logistic regression classifier will be to predict an away team victory and the less efficient it will be to predict a home team victory.

# III. Methodology

## Data Preprocessing

To make the dataset easier to work with, I will first rename its columns. Since the dataset has information of the last games of the teams, there are some missing values (e.g. the rows corresponding to the first game of the season does not contain any information of its last game). I will not consider these rows in my project. The next step will be deal with categorical variables (e.g. those containing the day and the month of the game). I will do so by converting these categorical variables into dummy/indicator variables. After that, I will create the label column, which will also contain dummy variables (1 if the home team won the game, 0 otherwise). When this is all done, I will drop all the unnecessary features.

Another important step is that I will exclude outliers that pollutes all the continuous variables. The objective here is to avoid distortion when fitting the classifiers that participates in the majority vote classifier, and therefore achieving a better accuracy with the model when testing it in the 2016-2017 regular season dataset.

As mentioned earlier in this report, the label column is not balanced (there are more home team victories than away team victories). To deal with this, when splitting the dataset, I will use stratified sampling so that both the training and the testing sets are similar with respect to the percentage of home team and away team victories in each of them. At this point it is important to note that this is only true for the training data and that this technique was not used in the 2016-2017 data.

Since I will utilize two different nearest neighbors' algorithms, I will also normalize the continuous variables in the training set. It is necessary because these algorithms will not work properly if, for example, 3 points shots converted is not in the same scale than rebounds.

The last step will be creating the three different datasets: the one with only the last game stats of the teams, the one with the last two games stats of the teams, and the one with the last five games stats of the teams. This is necessary because one of my goals with this project is to determine which of them is better to predict NBA games results.

## Implementation

All the steps in this section will be followed for each of the three datasets created earlier.

The first step is to reduce the curse of dimensionality. It will be done with Principal Component Analysis technique. The objective here is to reduce it to the *n* principal components that explains at least 60% of the variance in the dataset. For all the datasets, the value of *n* is 6. The PCA object created for each dataset will be used to transform (that is, to reduce) the correspondent 2016-2017 regular season dataset.

The next step will be creating a list of classifiers that contains all classifiers used in the majority vote one. At this point, all of them will have its parameters with their default values. For more information on these default values, I recommend reading scikit-learn 0.19.1 documentation at scikit-learn.org/stable/documentation.

After this, I will try to optimize the parameter's values of each classifier in the list created in the last paragraph. Although this step will be discussed in more details in the next section, it is important to list the parameters that I will try to optimize and which values I will use in order to achieve a better final result:

- Logistic regression classifier
    - C: .0001, .001, .01, .1, 1, 10
    - solver: saga, sag, newton-cg, lbfgs
    - class_weight: None, balanced
- Decision tree classifier
    - criterion: gini, entropy
    - min_samples_split: .1, .05, .025, .01, 2
    - min_samples_leaf: .05, .025, .0125, .005, 1
    - min_weight_fraction_leaf: 0, .125, .25, .5
    - class_weight: balanced, None
- K nearest neighbors
    - n_neighbors: 1, 3, 5, 7, 9, 11, 13
    - weights: uniform, distance
    - metric: euclidean, manhattan, chebyshev
- Multi-layer Perceptron
    - solver: lbfgs, sgd, adam

- alpha: .0001, .001, .01, .1
- learning_rate: constant, invscaling, adaptative

With the classifiers optimized, I will use them as parameters for the majority vote classifier. As mentioned earlier, it will predict a home team's victory if three or more of the classifiers predict home team's victory and will predict away team's victory if three or more of the classifiers predict away team's victory.

At this point, it is important to point the main complications I faced when performing the steps above:

- Defining the minimum variance explained by the *n* principal components: I tested values between 50% and 80% to define 60% as the most indicated for this problem.
- Choosing which parameters to optimize for each classifier: the classifiers (except Gaussian Naïve Bayes) has a lot of parameters, and I read a lot to understand the importance of each of them and choose the most relevant for the problem.
- Defining the values of some parameters: C in logistic regression, n_neighbors in K nearest neighbors and alpha in Multi-layer Perceptron were the most difficult ones.

### Refinement

For each of the reduced dataset, I will optimize each of the classifiers contained in the list. To do so, I will use stratified k-folds cross-validation and grid search techniques.

The stratified k-folds technique will be used to create train/test indices and split the data in train/test sets. The k-folds technique splits the data in *n* folds and test a model *n* times, each of them with *(n – 1)* folds as train set and the remaining fold as test set. The stratified means that the technique rearrange the data as to ensure each fold is a representative of the whole (at this point, it is important to remember that 60% of the victories in the data set are of the home team, which indicates the dataset is not well-balanced).

The grid search technique will be used to test each possible combination of the parameter's values and to indicate which is optimal. This technique consists on an exhaustive search over specified parameters values.

My idea is to test each combination of the parameter's value 10 times and, with this, determine which is the optimal parameter's values for the classifiers that will be used in the majority vote classifier.

## IV. Results

### Model Evaluation and Validation

For the dataset that considers the stats of the last game of each team involved in the match, the majority vote classifier has an accuracy of 59.4% when the classifier's parameters are the default values. After the optimization of these parameters, the following changes in the parameters' values occurred:

- Logistic Regression Classifier: *solver* changed from *liblinear* to *saga* (which was expected since the new parameter's value is recommended for large dataset because it is faster and more reliable, using a stochastic average gradient and supporting L1 penalization), and *C* changed from 1.0 to 0.0001 (this parameter indicates the inverse of regularization strength, therefore a smaller value leads to a stronger regularization);
- Decision Tree Classifier: *min_samples_split* changed from 2 to 0.1 (this parameter indicates the minimum number of samples required to split an internal node, and changed from 2 to 960), *min_samples_leaf* changed from 1 to 0.05 (this parameter indicates the minimum number of samples required to be at a leaf node, and changed from 1 to 480), and *min_weight_fraction_leaf* changed from 0.0 to 0.125 (this parameter indicates the minimum weighted fraction of the sum of weights required to be at a leaf node); and
- K nearest neighbors: *n_neighbors* changed from 5 to 13 (this parameter indicates the number of neighbors to consider when predicting) and *metric* changed from *minkowski* to *manhattan* (which means that the optimized classifier calculates the distance of two points as the sum of the absolut differences of their cartesian coordinates).

The accuracy of the majority vote classifier after the optimization is 59.5%. The optimization of the classifiers using this dataset led to an increase of 0.2% in the score of the majority vote classifier. The confusion matrix of this classifier can be found at the end of this section.

For the dataset that considers the stats of the last two games of each team involved in the match, the majority vote classifier has an accuracy of 58.7% before optimizing the classifier's parameters. After optimizing them, the following changes in the parameters' values occurred:

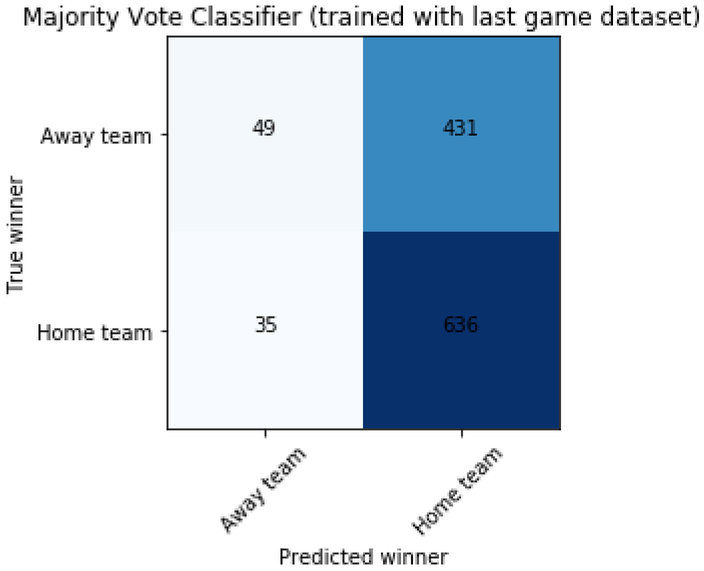
- Logistic Regression Classifier: *solver* changed from *liblinear* to *saga* (which was expected since the new parameter's value is recommended for large datasets), and *C* changed from 1.0 to 0.0001;
- Decision Tree Classifier: *min_samples_split* changed from 2 to 0.1, *min_samples_leaf* changed from 1 to 0.05, and *min_weight_fraction_leaf* changed from 0.0 to 0.125;
- K nearest neighbors: *n_neighbors* changed from 5 to 13, and *metric* changed from *minkowski* to *manhattan*; and
- Multi-layer Perceptron Classifier: *alpha* changed from 0.0001 to 0.01 (which indicates smaller weights for the parameters and a less curve decision boundary)

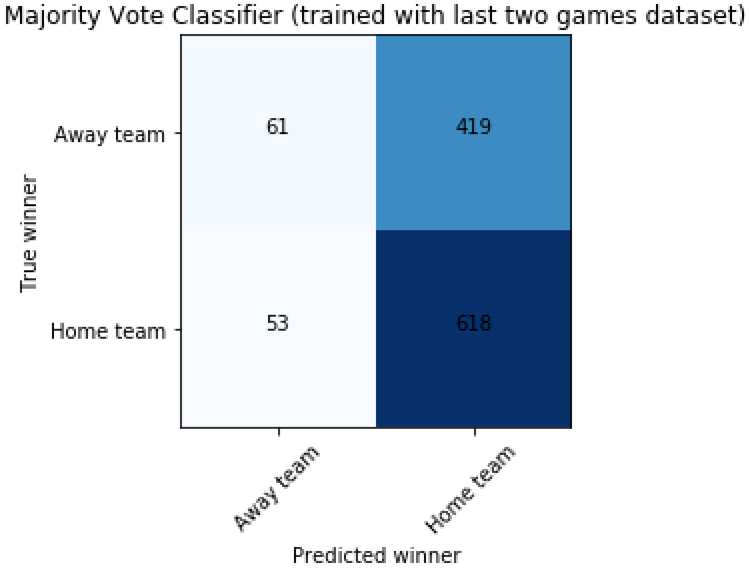
After the optimization, the accuracy of the majority vote classifier is 59.0%. For this dataset, it means an increase of 0.5% in the accuracy. The confusion matrix of this classifier is at the end of this section.

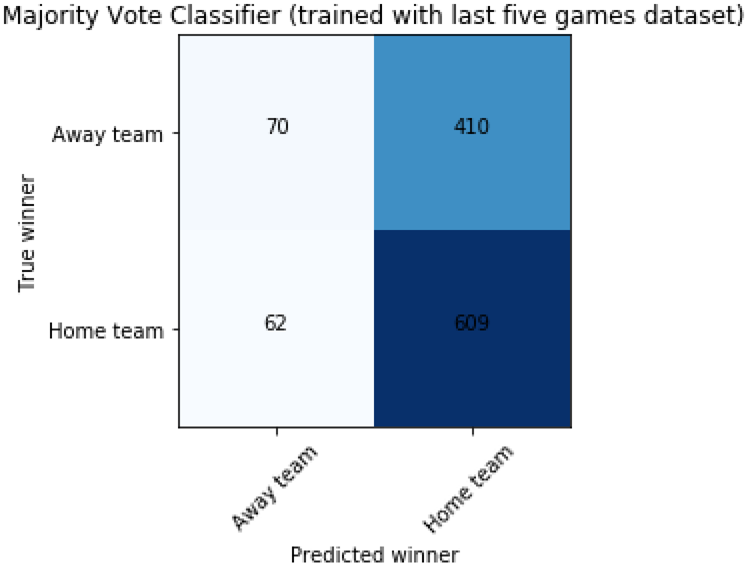
For the dataset that considers the stats of the last five games of each team involved in the match, the majority vote classifier has an accuracy of 58.8%. The optimization of the classifier's parameters led to the following changes in the parameters' values:

- Logistic Regression Classifier: *solver* changed from *liblinear* to *saga*, and *C* changed from 1.0 to 0.001;
- Decision Tree Classifier: *min_samples_split* changed from 2 to 0.1, *min_samples_leaf* changed from 1 to 0.05, and *min_weight_fraction_leaf* changed from 0.0 to 0.125;
- K nearest neighbors: *n_neighbors* changed from 5 to 13, and *metric* changed from *minkowski* to *chebyshev* (which indicates that the classifier now considers that the distance between two points is the greatest of their differences along any coordinate dimension); and
- Multi-layer Perceptron Classifier: *solver* changed from *adam* to *sgd* (which refers to stochastic gradient descent) and *learning_rate* changed from *constant* to *adaptive* (which means that after two consecutive epochs without any decrease in training loss, the learning rate is divided by 5).

After optimizing, the accuracy of the majority vote classifier is 59.0%, which represents an increase of 0.2%. The confusion matrix of this classifier, as well as the other confusion matrix discussed in this section, are the following:







From the confusion matrices above, it is possible to observe that the majority vote classifier has the same behavior of the untouched and untuned linear regression classifier: the more games we use to create the dataset, the more efficient the classifier will be to predict an away team victory and the less efficient it will be to predict a home team victory.

Besides these two observations made from the confusion matrices, it is important to note what happens with the accuracy of the classifier: the highest one is from the predictions made on the last game dataset, and it decreases as the model considers the last two and five games.

That said, I would not consider that the model is capable of generalize well to unseen data, since it tends to predict the home team as the winner. Improvements must be made to avoid this problem and turn the model more reliable (the last subsection of this document discuss improvements).

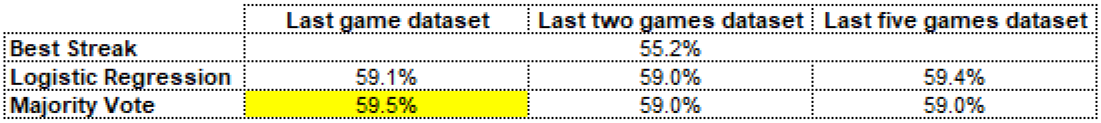
### Justification

For the dataset that considers stats of the last game of each team involved in the match, the final model works properly, increasing in 7.8% the correct predictions if compared with the best streak classifier, and in 0.7% if compared with the untouched and untuned logistic regression classifier.

For the dataset that considers stats of the last two games, the accuracy of the majority vote classifier has a curious characteristic: although it is 6.9% higher than the accuracy of the best streak classifier, it is the same as the untouched and untuned logistic regression classifier.

A more curious behave occurred for the dataset that considers stats of the last five games: the majority vote classifier has an accuracy 6.9% higher than the best streak classifier, but 0.7% lower than the untouched and untuned logistic regression classifier.

The following table has all the accuracies of the classifiers analyzed:



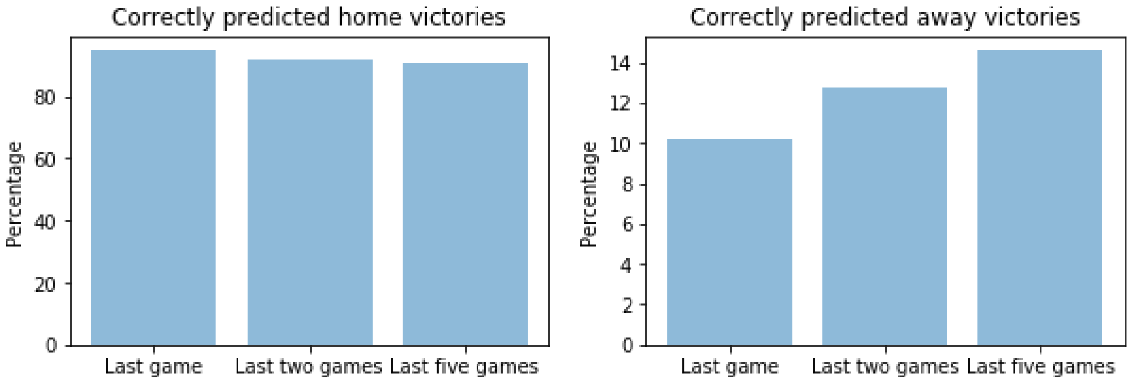
| | Last game dataset | Last two games dataset | Last five games dataset |
|---|---|---|---|
| Best Streak | 55.2% | | |
| Logistic Regression | 59.1% | 59.0% | 59.4% |
| Majority Vote | 59.5% | 59.0% | 59.0% |

From the table above, it is possible to state that the best classifier to predict the winner of a NBA match is the majority vote classifier trained with the last game dataset.

## V. Conclusion

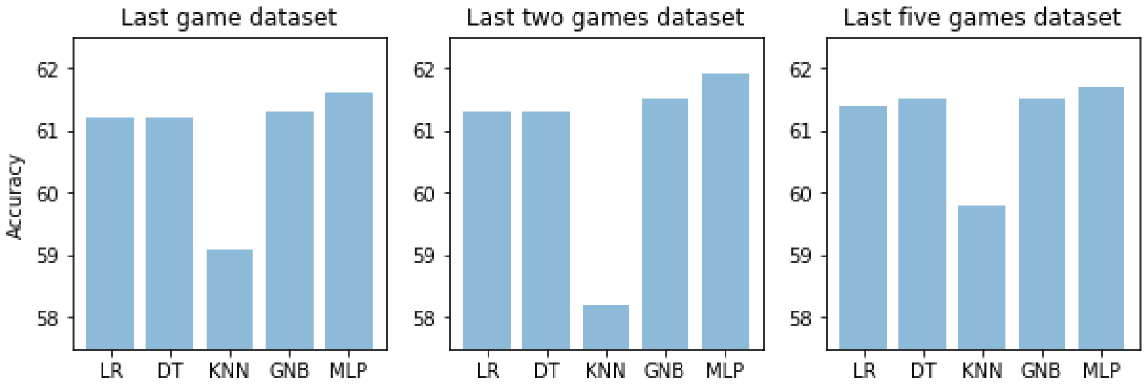
### Free-Form Visualization

After analyzing all the confusion matrices, I believe it is important to highlight that all the classifiers have difficult to predict an away team victory. The figure below shows the percentage of correctly home team victories and the percentage of correctly away team victories for the three datasets analyzed:



That said, I believe that possible users of this model should trust a lot more when it predicts a home team victory.

Another thing I noticed and consider important to say is that the Multi-layer Perceptron Classifier had the best accuracies of the classifiers used by the Majority Vote Classifier in all the datasets (approximately 61.8%), as the next figures shows:



Because of this, I plan to work in a model with neural networks in this same problem in the future.

## Reflection

The whole process when creating the final model can be summarized as follows:

- Getting the data in a specialized website using Sports Query Data Language
- Removing repeated samples and outliers
- Exploring the dataset
- Preprocessing the dataset
- Selecting classifiers used in the ensemble model and choosing benchmark
- Reducing dataset dimensionality via unsupervised learning
- Optimizing classifiers
- Defining the final model

Although the Majority Vote Classifier haven't achieved a satisfactory performance, I think this project was very enriching for me because I had to read a lot to conclude it. The results were not the ones I expected: since I believe that the phase of a team indicates how well it will probably play a match, I expected that it would work better with the last five games dataset, which did not occur, and also expected that the model created by me would work better.

## Improvement

As mentioned in the Free-Form Visualization, I plan to work in a model with neural networks in this problem in the future. Since this course was for me an introduction to Machine Learning and AI universe, I did not feel comfortable to go deeper in the MLP classifier, modifying the number of parameters or even trying to optimize its hyperparameters. Creating a more robust and well-tuned neural network is one of the possible improvements I believe would be great for the stated problem and its solution.

Because of the limitations of the computer I used to complete this Capstone Report, I was not able to run certain algorithms that has a higher computational cost to be executed, such as Support Vector Machines (SVM), and it took me too long to execute Multi-Layer Perceptron. SVM would be interesting in this problem because of its capability of performing non-linear classification using kernel tricks. That said, I think that another possible improvement for this Capstone Report would be to include a SVM classifier in the list of classifiers used by the Majority Vote Classifier and to run the code in a better machine.