# Data Mining

## CSCI-6674-01

## DAT (Data Analysis Team)


**UNIVERSITY OF NEW HAVEN**

**TAGLIATELA COLLEGE OF ENGINEERING, WEST HAVEN, CT.**

**SUBMITTED TO:**

**Prof. Reza Sadeghi**

# DAT (Data Analysis Team)

**Head of Team:**

Anuj Parikh apari6@unh.newhaven.edu

**Team Members:**

Anuj Parikh apari6@unh.newhaven.edu

Rahul Akkineni rakki1@unh.newhaven.edu

## TABLE OF CONTENTS

# 1. Research Question:

Predicting the 5-year career longevity of the NBA Rookies.

That is, we try to predict whether if a rookie stays for 5-years after coming to the NBA based on his performance during his time.

## 1.1 Objective:

By predicting longevity of NBA Rookies, our objective is to predict which player is going to last long in NBA and how much score he can make in a single game. At the same time, Coach can choose healthy players for greater winning strategy and energetic game.

# 2. Data Modeling:

## 2.1List of Modeling Techniques:

- Classification Methods
    - Logistic Regression
    - Naïve Bayes
        - Gaussian
        - Bernoulli
    - SGD (Stochastic Gradient Descent)
    - K Nearest Neighbors
    - Decision Tree
    - Random Forest
    - Extra Tree Classifier
    - Support Vector Machine
    - Bagging
    - Gradient Boosting
    - AdaBoost
    - XGBoost
    - LGBM
    - Multi-Layer Perceptron Neural Networks

## 2.2Hardware:

- **Processor:** Intel Core i7 – 10510U @ 1.8GHz  2.30GHz
- **RAM:** 8.00 GB
- **System Type:** 64-bit Operating System, x64-based processor

### 2.3 Parameter and hyperparameter:

- **Following hyperparameters are used in above mentioned classification methods:**
    - **Logistic Regression:** We used bfgs (Broyden–Fletcher–Goldfarb–Shanno) algorithm for nonlinear optimization. While setting maximum iteration up to 1000.
    - **Naïve Bayes:** We set hyperparameter of variance smoothing to 1e-9 for achieving stability.
    - **SGD classifier:** In SGD (Stochastic Gradient Descent), we used hinge loss based l2-norm penalty. We performed SGD with boundary of 1000 maximum iterations.
    - **KNN classification:** We set deciding neighbours to 5 As odd number is easier to deal with when it comes to same votes for new classification object.
    - **Decision Tree, Extra-Tree Classifier and Random Forest:** We set decision tree's max depth unbounded, so we can leverage all the branches split data in decision tree as well as random forest.
    - **SVM:** We used radial basis kernel in Support Vector machine.
    - **Bagging and Boosting Classifier:** We set 10 and 100 for bagging and boosting algorithms' estimators, respectively.
    - **AdaBoost:** we used SAME.R algorithm for optimizing adaboost with 50 estimators.
    - **LGBM classifier:** We leverage lgbm gradient boosting classifier on 200000 bin-sample-size and 50 estimators with having upto 31 leaves for base learners.
    - **Multilayer Perceptron Neural Networks:** We used relu activation function with adam optimizer with batch-size set to 200. We set learning rate to 0.001.

## 3. Performance Metric

- We get best accuracy of 72.56 % using Logistic Regression on all available features from dataset. Graph 1 shows other classification method comparisons followed by Table 1 which describe values compared in graph.

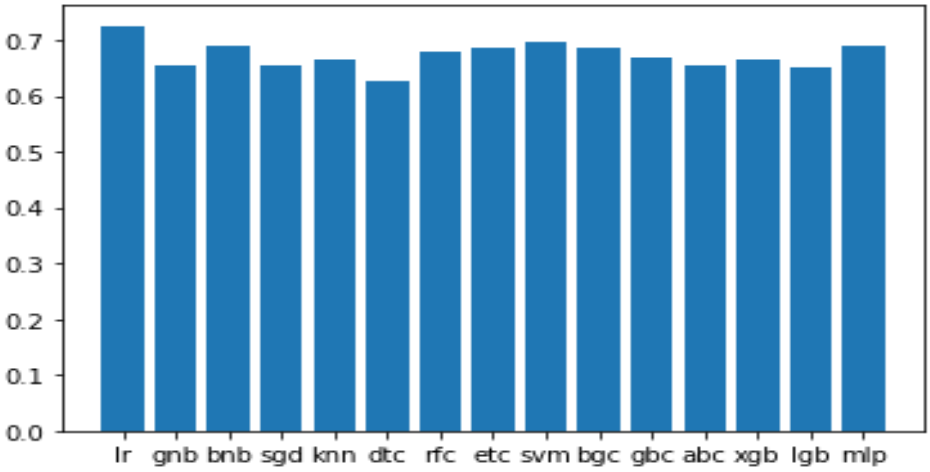


**Figure 1 After Standardization with All features**

| Model | Train Accuracy | Test Accuracy |
|---|---|---|
| Logistic Regression | 0.708 | 0.7256 |
| Gaussian Naïve Bayes | 0.626 | 0.6555 |
| Bernoulli Naïve Bayes | 0.656 | 0.6890 |
| Stochastic Gradient Descent | 0.636 | 0.6555 |
| K Nearest Neighbors | 0.646 | 0.6646 |
| Decision Tree | 0.583 | 0.6250 |
| Random Forest | 0.688 | 0.6799 |
| Extra Classifier | 0.692 | 0.6860 |
| Support Vector Machine | 0.702 | 0.6951 |
| Bagging | 0.652 | 0.6860 |
| Gradient Boosting | 0.686 | 0.6677 |
| Ada Boost | 0.676 | 0.6555 |
| XGBoost | 0.681 | 0.6646 |
| LGBM | 0.670 | 0.6524 |
| Multi -Layer Perceptron | 0.682 | 0.6890 |

## Table 2 Accuracy Table

- We try several hyperparameter setting with variations of features passes to learning model. 2 variations are depicted in Graph 2 and 3 Followed by table to describe value compared in graphs.

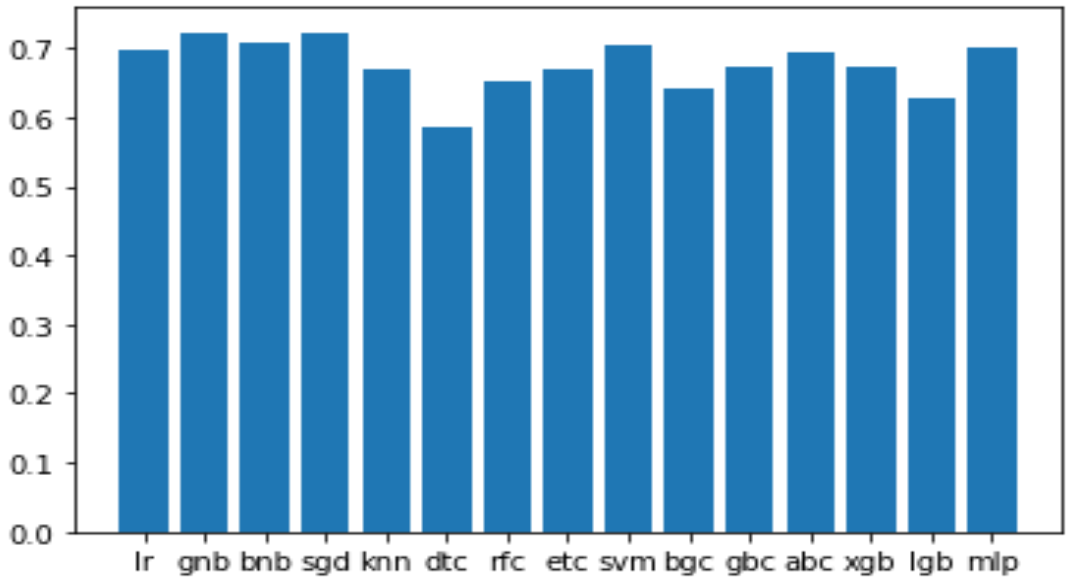


**Figure 2 After Standardization with 5 features**

| Model | Train Accuracy | Test Accuracy |
|---|---|---|
| Logistic Regression | 0.700 | 0.6982 |
| Gaussian Naïve Bayes | 0.668 | 0.7226 |
| Bernoulli Naïve Bayes | 0.683 | 0.7073 |
| Stochastic Gradient Descent | 0.622 | 0.7226 |
| K Nearest Neighbors | 0.651 | 0.6707 |
| Decision Tree | 0.582 | 0.5854 |
| Random Forest | 0.642 | 0.6524 |
| Extra Classifier | 0.630 | 0.6677 |
| Support Vector Machine | 0.701 | 0.7043 |
| Bagging | 0.622 | 0.6402 |
| Gradient Boosting | 0.667 | 0.6738 |
| Ada Boost | 0.673 | 0.6921 |
| XGBoost | 0.674 | 0.6738 |
| LGBM | 0.648 | 0.6280 |
| Multi -Layer Perceptron | 0.701 | 0.7012 |

**Table 2 Accuracy Table**

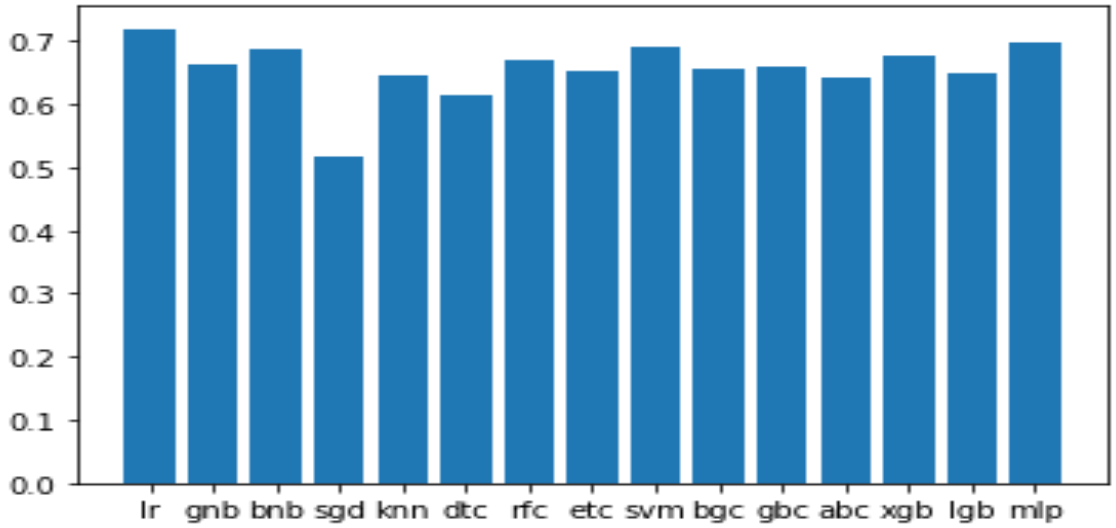


**Figure 3 After Standardization with 10 features**

| Model | Train Accuracy | Test Accuracy |
|---|---|---|
| Logistic Regression | 0.703 | 0.7195 |
| Gaussian Naïve Bayes | 0.632 | 0.6616 |
| Bernoulli Naïve Bayes | 0.661 | 0.6860 |
| Stochastic Gradient Descent | 0.671 | 0.5183 |
| K Nearest Neighbors | 0.639 | 0.6463 |
| Decision Tree | 0.605 | 0.6128 |
| Random Forest | 0.676 | 0.6707 |
| Extra Classifier | 0.674 | 0.6524 |
| Support Vector Machine | 0.696 | 0.6921 |
| Bagging | 0.663 | 0.6555 |
| Gradient Boosting | 0.694 | 0.6585 |
| Ada Boost | 0.674 | 0.6433 |
| XGBoost | 0.688 | 0.6768 |
| LGBM | 0.666 | 0.6494 |
| Multi -Layer Perceptron | 0.688 | 0.6982 |

## Table 3 Accuracy Table

## 4. Conclusion:

After analysing assortment of learning model experimented on various hyperparameters, we get 70+% accuracy on learning methods, which strongly support the argument of relating player historical performance to longevity of player. However, 72.56% is not good enough to use it in practical manner, But It is pretty good to deduce that pattern exist in asked research question. We strongly believe that Data Model with larger data can more accurately pinpoint relations between performance and longevity.

**GitHub link:** https://github.com/Anuj-parikh/DAT