Preparation

In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from plot_measures import plot_measures
from algorithms_comparison import plot_algorithms_comparison
```

Introduction

The purpose of this project is to implement a logistic model from scratch. We begin with preparing the data for analysis, then we implement 3 optimization algorithms for training the model, and finally we assess the result using statistical measures.

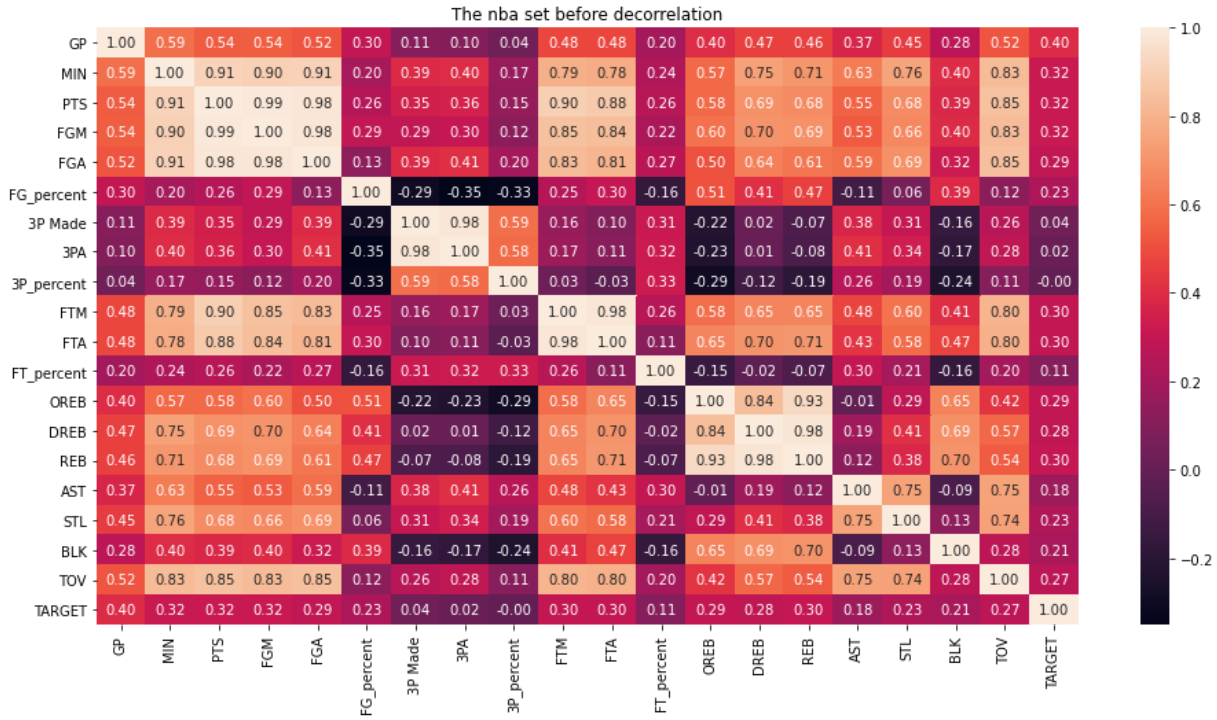
Datasets

Since the model needs to be trained and tested, there is a need to prepare some data. We collected 5 datasets:

- **candies** - dataset concerning production of candies. Sample questions addressed by the variables: *Does it contain chocolate?, Is it fruit flavored?, Is there caramel in the candy?, Does it contain peanuts, peanut butter or almonds?, Does it contain nougat?*. The target variable describes percentage of people who liked this candie type.
- **breast cancer** - dataset describing health state of a patients suffering from breast cancer. The target variable describes if the cancer is maligant.
- **bankrupcy** - this is a dataset from a recruiting task from a consulting firm. Variables include information of each candidating company, for example income, marketing statistics, ratings, sector number of employees, etc. The target variable represents information about bankrupcy. This dataset is significantly larger than other.
- **nba** - dataset on all the NBA finals results and NBA Regular Seasons MVP.
- **wine** - dataset related to red and white variants of the Portuguese "Vinho Verde" wine.

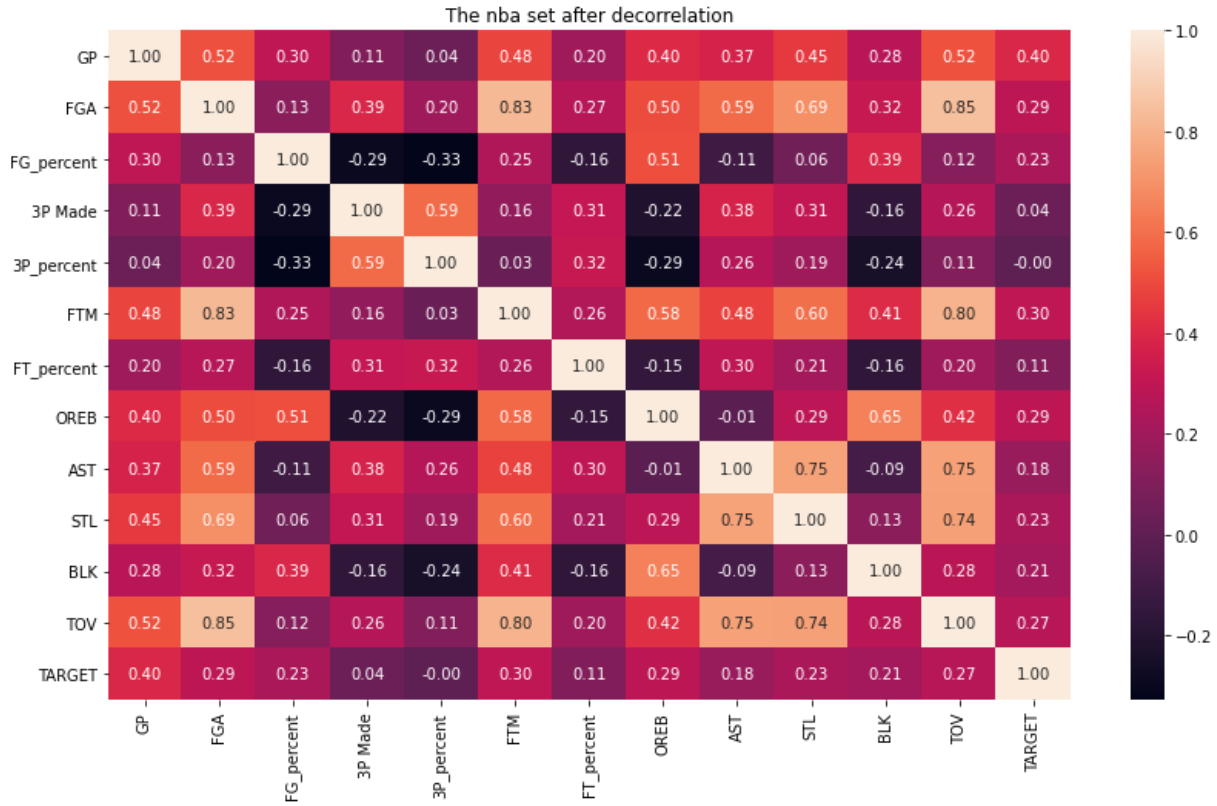
The first step was downloading the data and conversion of types. We saved the initially preprocessed data under directory `data/before preprocessing`. Next step was to replace missing data with mean value of each variable. Then we analyzed correlation matric for each dataset. Let's consider the **nba** dataset

In [2]:
```
nba = pd.read_csv('data/before preprocessing/nba.csv')
plt.figure(figsize=(16, 8))
plt.title('The nba set before decorrelation')
_ = sns.heatmap(nba.corr(), annot=True, fmt='.2f')
```



We dropped the most correlated variables. Below we present the correlation plot after decorrelation.

In [3]:
```
nba = pd.read_csv('data/after preprocessing/nba.csv')
plt.figure(figsize=(14, 8))
sns.heatmap(nba.corr(), annot=True, fmt='.2f')
_ = plt.title('The nba set after decorrelation')
```



Measures

We introduce 3 measures of fit: *accuracy*, *precission* and *recall*. The implementation can be found in `log_regression.py` file.

$$accuracy(y, \hat{y}) = \frac{1}{n} \sum_i I_{\{y_i = \hat{y}_i\}}$$

$$precision(y, \hat{y}) = \frac{\sum_i I_{\{y_i = \hat{y}_i = 1\}}}{\sum_i I_{\{\hat{y}_i = 1\}}}$$

$$recall(y, \hat{y}) = \frac{\sum_i I_{\{y_i = \hat{y}_i = 1\}}}{\sum_i I_{\{y_i = 1\}}}$$

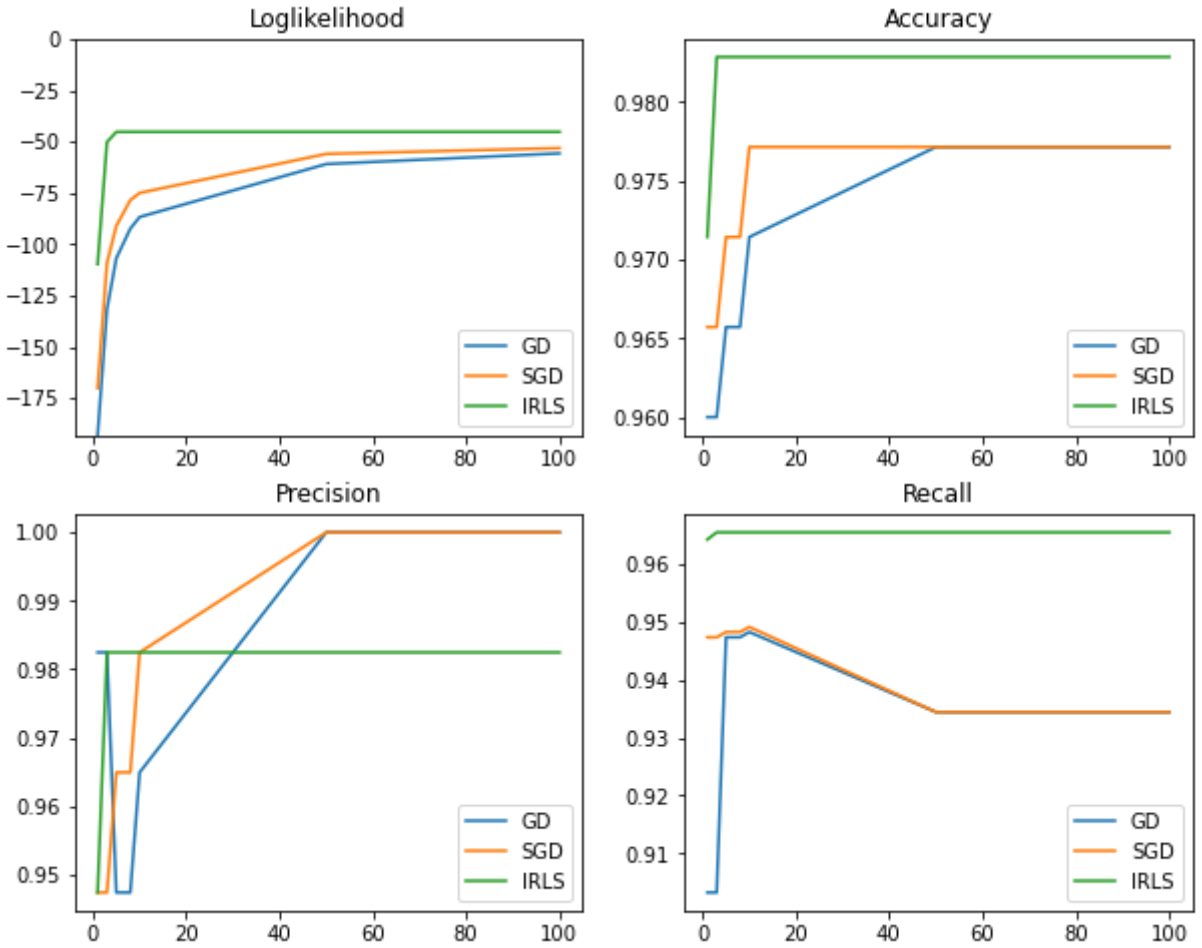
Model

The main component of our model is the `LogisticModel` class present in `log_regression.py` file. It includes 3 training methods - `GD`, `SGD` and `IRLS`, method `fit` which given a dataset predicts the results and measures like `log_likelihood` and `R2_measure`.

Loglikelihood and goodness of fit

We have examined how loglikelihood and other measures are affected by number of iterations. Increasing number of iterations makes a difference mainly for smaller number of iterations.

In [11]:
```
plot_measures()
```



Comparison against other algorithms

We compared the logistic model created by us (concerning all the implemented optiomization algorithms) againts other popular models, like LDA, QDA and KNN. We used the **breast cancer** for the computation. Below we present we result of the comparison:

In [15]:
```
plot_algorithms_comparison()
```

```
Algorithm converged after 10 iterations
Maximum metric:  0.6628571428571428  at K =  11
```