NBA salary prediction: Data engineer/Software engineer

Giovanni Zanin

University of Trieste

Data engineer/Software engineer

GIOVANNI.ZANIN@studenti.units.it

ABSTRACT

The "NBA salary prediction" software has the aim of helping the president or the managers of an NBA basketball team to decide the most appropriate salary to offer to a player such as to ensure the fastest negotiation and closing of an agreement, according to the league average values and to the stats of the player in the last season. To find a prediction function for the best salary value, machine learning techniques will be used.

Keywords

prediction; salary; NBA; dataset; learning.

1. INTRODUCTION

The negotiation of the annual salary of a professional athlete is an important aspect in modern sports. It happens often that the two parts involved in the negotiation cannot find a deal to sign a contract, even after hours of talks. In the past few years, in many negotiations, data analysts have been consulted. Their job consists in finding the most appropriate contract for an athlete relying on his stats and on the contribute he could give to the team (one of the first and most relevant cases is the one of the football player Kevin De Bruyne, who consulted exclusively data analysts for the sign of his contract with Manchester City [1]). This solution, which does not involve any external agent, is in general advantageous because both parts are usually more inclined to accept and consider fair an offer "scientifically" determined. Negotiation times will be shortened and the relation will be based on higher level of trust and satisfaction.

In basketball, players' stats have a fundamental role in the evaluations, and so is in the NBA (National Basketball Association) the most prestigious basketball league in the world. In this project data from the contracts stipulated in the NBA in the last 10 years will be used to predict the best salary a player should receive and to create an interactive application for this purpose. The assumption is that the information about the contribution a player could give to a team, in terms of successes and eventually trophies, could be entirely (or in large part) determined by his data from the previous year.

2. SOFTWARE DEVELOPMENT METHOD

A software developer and a software/data engineer composed the team. The aim was to create a general-purpose product consisting in an interactive application that used modeling techniques of machine learning.

The software process adopted for the project was partially a "plan driven" approach and partially "agile". In fact, until the first prototype was completed, a predefined plan was followed. It was composed by many tasks each one assigned to one of the two developers, with a given deadline. For example, the data preprocessing task was assigned to the data engineer with a deadline of three days, then, after other three days, the first prediction function had to be ready and so on. Of course, no task was assigned to a member of the team until he had completed his previous one. This phase was characterized by a clear distinction of:

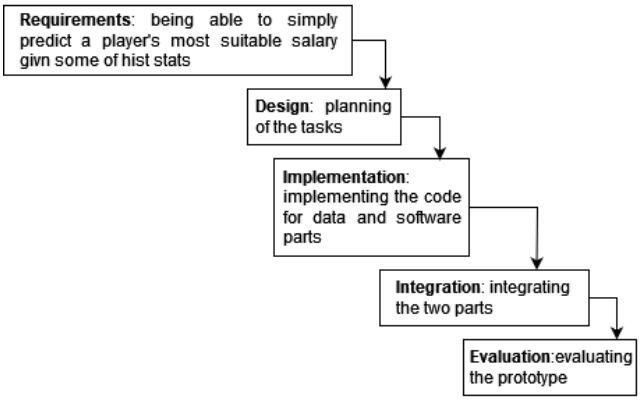

Figure 1. Waterfall software process model adopted for the first prototype

- Specification: definition of the objective of the project, its audience (possibly NBA managers), its requirements and its initial structure and milestones,
- Development: tasks' design, implementation and integration of the software components, including a first simple UI/UX design,
- Validation: verification of units and integrated components.

This plan led quickly to the creation of the first software prototype, which was the result of a waterfall process (see figure 1). The main objectives of this prototype were validating simple requirements but most of all having a starting point from where it was possible to think and design possible improvements (in fact this protype was created also to be a software initial version).

In the succeeding evolution of the product, the software process adopted was "agile", as no specific plan was followed. The development method was incremental as specification, implementation and validation interleaved while searching for possible improvements both in the backend (mainly trying to improve the results of the salary predictor) and in the frontend (adding some functionalities to the initial prototype). This procedure took some aspects of the "Scrum" extreme programming model as the improvements/functional addictions to the system were planned as "to do" tasks at first and then developed in "sprints". Of course, the first improvements made were the ones that were considered most important for the software requirements.

The incremental development from the creation of the first prototype could be represented with the Boehm's spiral model. After planning and specifying the objectives of an improvement, the risk evaluation consisted mainly in evaluating the feasibility of the increment (influenced both from the developers' ability and the time available) and the consequences it would have caused to the rest of the software and to the software specifics. Finally, the new component was implemented and tested (see figure 2).

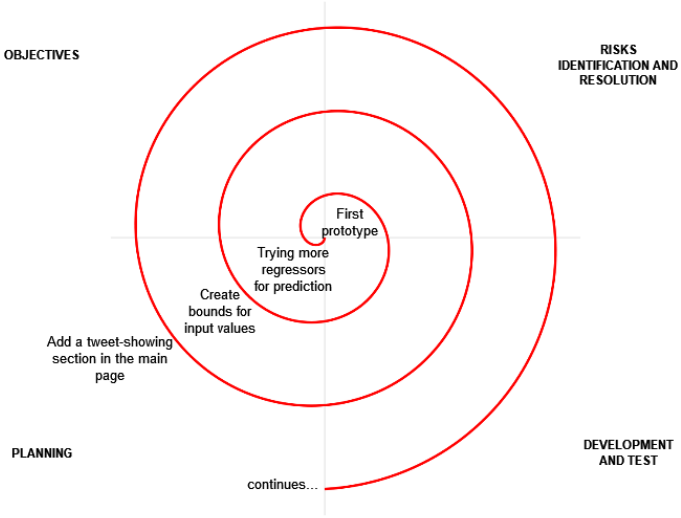

Figure 2. Portion of the spiral that describes the software developing after the creation of the first prototype

In conclusion the simplicity of the project and the small size of the team led inevitably to a mainly agile/incremental process, but for sure even aspects of waterfall (specially for the creation of the first prototype) and reuse-oriented processes were adopted (portions of code of old projects of the team members were reused).

2.1 Software requirements

The main software requirements the team worked on were:

- receive input data and return a prediction (availability),
- find a prediction as precise as possible relying on the league's standards (reliability),
- be usable,
- the code should be easy to inspect and to update (maintainability),
- the user's interface should be visually pleasing.

During the incremental developing there were added many other requirements as implementing secondary features in the main page (for example tweets and news about the NBA).

2.2 Technical architecture

The tools used in the creation of the software were:

- Github for version control of the overall project,
- Google Colab for versioning of personal parts of the project (for example tests on data mining),
- Github Projects for tasks scheduling,
- Google Colab and Spyder for coding,
- Python, HTML and CSS as languages,
- Streamlit and Streamlit Cloud as platforms.

2.3 Testing

For the backend part of data mining the development was test-driven as automated tests were implemented to, after every modification in the dataset or in the learning technique, validate the model and evaluate the effectiveness of the technique. Automated tests were used also to validate the prediction functions build with the best algorithm previously found. These tests consisted in giving some values (including extreme values) as input and assessing the output. On the frontend side no real plan was followed for testing as graphical aspects were only visually evaluated.

3. DATA PROCESSING

3.1 Dataset

The chosen Kaggle dataset [2] includes information about NBA contracts signed from 2010 to 2020 by players active in the 2020/2021 season. The contracts are only the ones signed by players who were already playing in the NBA (this means there is not any information about contracts signed by players who hadn't played in the NBA the year before). Every record contains:

- Player's general information (name and age when signing the contract)
- Contract's information (starting year, ending year, average salary per year)
- Player's stats registered the year before the sign (games won, games lost, total games played, total minutes played, points made, 2-points baskets made, 2-points baskets attempted, 2-points baskets percentage, 3-points baskets made, 3-points baskets attempted, 3-points baskets percentage, free throws made, free throws attempted, free throws percentage, offensive rebounds, defensive rebounds, total rebounds, assists, turn overs, steals, blocks, personal fouls, plus/minus)

The dataset contains 199 records in which the average salary per year (the feature that this project wants to predict) variates from a minimum of 823.000$ to a maximum of 33.600.000$ (the mean value is 11.100.000$). Except for players' names and plus/minus stat, all the twenty-eight features are quantities greater or equal to zero (see table 1).

3.2 Preprocessing

To prepare the dataset for the learning phase some preprocessing actions were applied:

- removing players' names' feature,
- substituting "CONTRACT_START" and "CONTRACT_END" with a column simply specifying the duration of the contract; it was chosen to ignore the information about the beginning year of the contract even then that could be important as, even in a brief period as 2010-2020, salaries' averages have slightly changed,
- removing duplicates (only one row),
- normalizing the observations with min-max scaler.

	mean	std	min	max
CONTRACT_START	2015.23	2.07	2011	2019
CONTRACT_END	2017.51	1.71	2013	2020
AVG_SALARY	11125155.67	7884208.66	823244	33599500
AGE	25.94	2.85	20	36
GP	64.42	19.29	1	82
W	34.34	14.41	0	64
L	30.08	12.93	0	62
MIN	1754.86	776.44	2	3125
PTS	817.16	498.44	0	2376
FGM	301.75	178.06	0	743
FGA	641.66	373.32	1	1643
FG%	46.73	8.11	0.00	100.00
3PM	63.87	58.43	0	272
3PA	174.70	148.65	0	657
3P%	29.66	13.24	0.00	50.00
FTM	149.79	128.28	0	720
FTA	196.48	162.08	0	837
FT%	74.20	14.82	0.00	100.00
OREB	79.74	72.68	0	397
DREB	250.21	163.52	1	829
REB	329.95	225.51	1	1226
AST	172.41	163.96	0	839
TOV	103.84	70.78	0	374
STL	58.46	37.20	0	169
BLK	39.83	43.37	0	269
PF	138.30	63.82	1	291
+/-	63.40	228.14	-628	839

Table 1. Statistics of dataset's numerical features

3.3 Learning phase and dataset adaptations

This phase consisted in fitting and evaluating many regressors (12) and trying to improve effectiveness slightly modifying the dataset. For some regressors hyperparameter tuning was actuated. Every algorithm was evaluated using 10-CV and the mean percentage error (MPE) on the test set, with the Dummy regressor, which scored MPE=155%, kept as a baseline. Here is the list of the actions made:

- The twelve regressors considered were fitted and evaluated on the initial dataset.
- The same regressors were fitted on the dataset adding the "CONTRACT_START" column. The results did not improve.
- The features that were arithmetically dependent to others (for example REB=DREB+OREB) were removed from the dataset. The results improved and the changing was maintained.
- The most important features were selected in the dataset using feature ablation method. The results didn't improve.
- Standardization was used as normalizing method, instead of min-max scaler. The results did not improve.
- Every on-field stat column (excluding the ones with percentages) was substituted with the same stat in relation to the minutes played by the athlete. The results did not improve.
- A 5-bins discretization was actuated on data before processing them. The results didn't improve.

At this point, a prediction function was defined by fitting, on the obtained dataset, the regressor with the best MPE. It was the K-Nearest Neighbors regressor which scored MPE=45% on test set for $n_{neighbors}=4$ (obtained with hyperparameter tuning). Table 2 shows the performances of some of the regressors adopted in the best dataset conditions.

	Linear	Random Forest ($n_{tree}=500$)	4-Nearest Neighbors	Dummy
Initial dataset	55%	50%	50%	155%
After removing dependent features	55%	50%	45%	155%
After selecting 15 best features	53%	51%	50%	155%

Table 2. Most relevant results of data mining, in terms of MPE

4. PROPOSAL LIMITATIONS

4.1 Applicability limitations

- The regression is designed on data regarding NBA, so it will not be effective if players' stats from the previous season given by input are collected in another league (the contribution a player could give to an NBA team may be very different from the one he could give in other leagues).
- The regression cannot be applied to predict salaries of players at their first experience in basketball, as no stats would be available.
- Some bounds have been created for the input to be acceptable (for example FTM can't be greater than FTA), but still there are some unlikely but theoretically possible inputs (i.e. with one or more features far from the mean values) which would lead to unpredictable outputs. This is the case of giving as input stats like every value equal to 0, except for MIN=1, 3PM=500 and 3PM=500 (very unrealistic). The algorithm could interpret these dummy stats as the ones of a very valuable player (so many 3-pointers in only one minute!) and give a realistic salary as output.

4.2 Effectiveness limitations

- The learning function considers stats only for the previous season. Many times, a season could be poorly representative of a player's abilities (he could be injured, unlucky…).
- The interest of a team to a player (and the salary the team is willing to give him) is often not exclusively dependent on athlete's on-field qualities. It can involve, for example, player's leadership with the teammates or even player's popularity and visibility (in this way columns like "number of sponsors" or "followers on social media" could be added to the dataset).
- Even in a sport dominated by stats like basketball, numbers are not everything!

The high value of MPE obtained does not necessarily have to be seen as a limitation: the project tries to predict the most suitable salary for the player, not the real salary stipulated. It's possible that in the test set (which contains real contracts signed) there are some outliers (i.e. some negotiations have been particularly successful for one of the parties).

5. REFERENCES

[1] Kevin De Bruyne uses data analysts to broker £83m Man City contract without agent, Mirror

https://www.mirror.co.uk/sport/football/news/kevin-de-bruyne-uses-data-23870686

[2] Current NBA Players Contracts History, Jarosław Jaworski, Kaggle.com

https://www.kaggle.com/datasets/jarosawjaworski/current-nba-players-contracts-history

[3] Github repository of the project

https://github.com/JohnnyBravo10/NBA-salary-prediction-

[4] Home page of the application

https://fedrosauro-nba-salary-prediction--mainpage-k0mruc.streamlit.app/