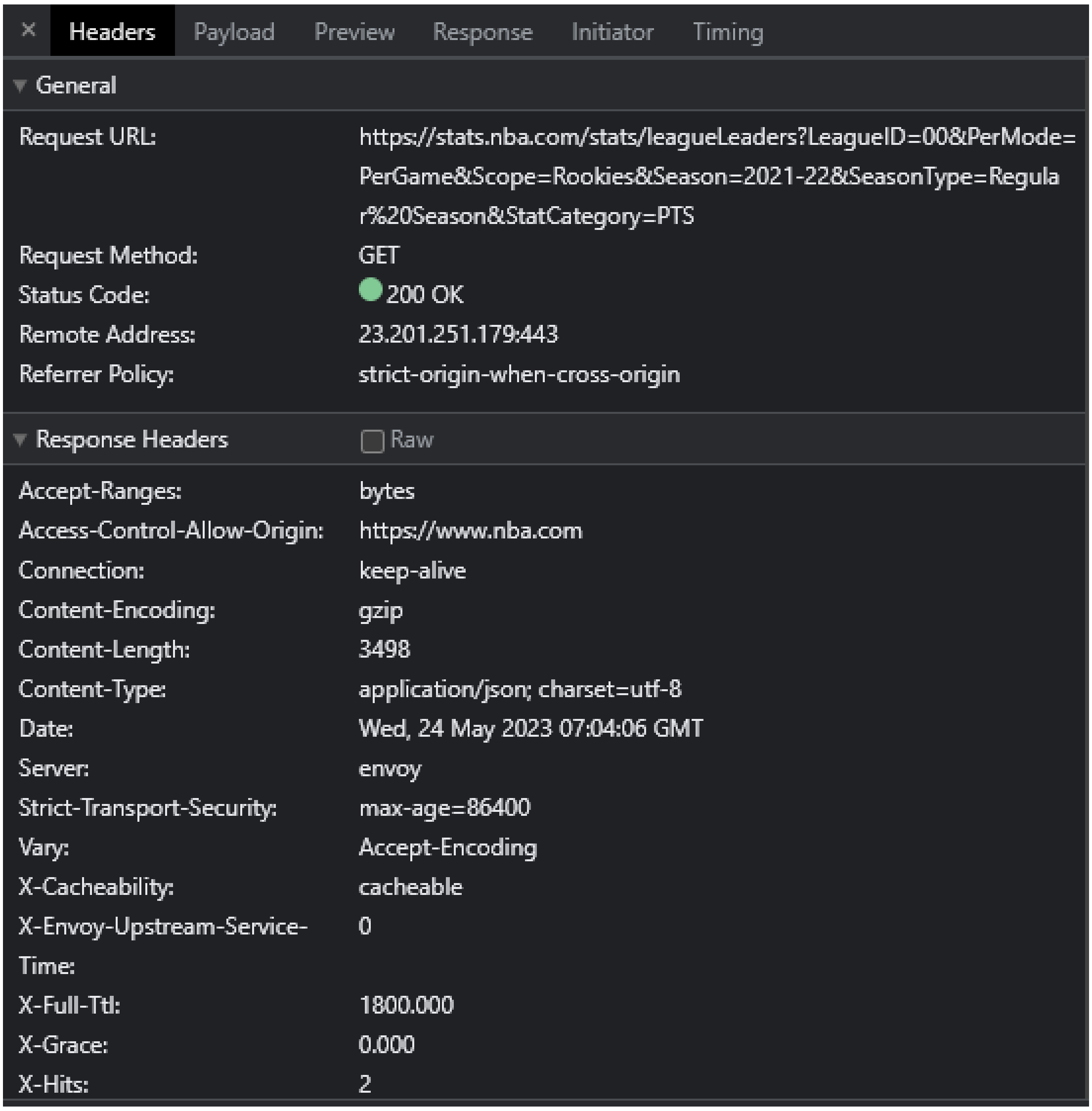
FIZ425E Project Report

Ömer Tüntül
091070109

My main motive for this project is to see the change over the 20 years in the regular season and playoffs of the NBA games. On top of that, I also add a small side project to see whether sklearn can guess if a player is a rookie or not.

First, I need to create a dataset for all of this. I don't want to make it with web scrapping. Under this URL: https://www.nba.com/stats/leaders?SeasonType=Regular+Season&Season=2021-22&Scope=Rookies there is an API to store the information I need which can be seen down below.

https://stats.nba.com/stats/leagueLeaders?LeagueID=00&PerMode=Totals&Scope=Rookies&Season=2020-21&SeasonType=Playoffs&StatCategory=PTS



```
Request URL:                   https://stats.nba.com/stats/leagueLeaders?LeagueID=00&PerMode=
                               PerGame&Scope=Rookies&Season=2021-22&SeasonType=Regula
                               r%20Season&StatCategory=PTS
Request Method:                GET
Status Code:                   ● 200 OK
Remote Address:                23.201.251.179:443
Referrer Policy:               strict-origin-when-cross-origin

Accept-Ranges:                 bytes
Access-Control-Allow-Origin:   https://www.nba.com
Connection:                    keep-alive
Content-Encoding:              gzip
Content-Length:                3498
Content-Type:                  application/json; charset=utf-8
Date:                          Wed, 24 May 2023 07:04:06 GMT
Server:                        envoy
Strict-Transport-Security:     max-age=86400
Vary:                          Accept-Encoding
X-Cacheability:                cacheable
X-Envoy-Upstream-Service-      0
Time:
X-Full-Ttl:                    1800.000
X-Grace:                       0.000
X-Hits:                        2
```

This also gives us the Request method as well. But the problem is it provides only one year, one type of season type, and one scope of player type. So, by changing the request URL in three For Loops the dataset can be created.

After the dataset is created, it needs to be cleaned and updated slightly. Some procedures I have done,

- Updating the old team names
- Dropping unnecessary columns: Rank, Team_id, Player_id

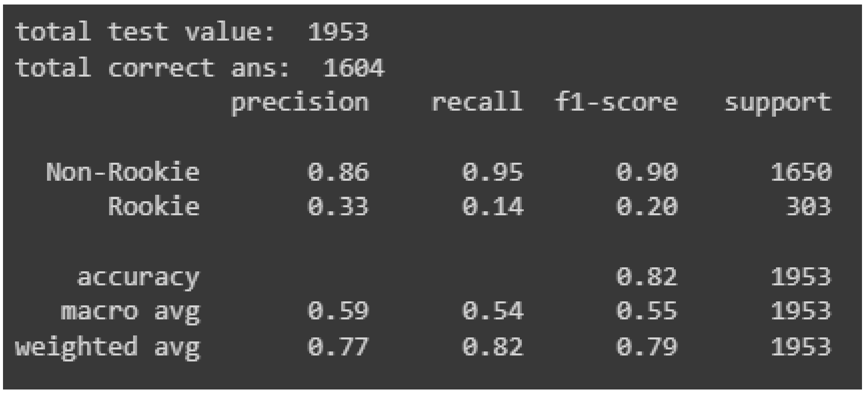
And most importantly:

- Dropping the rows where a player's rookie year but labeled as Non-Rookie.

Guessing a player is a rookie:

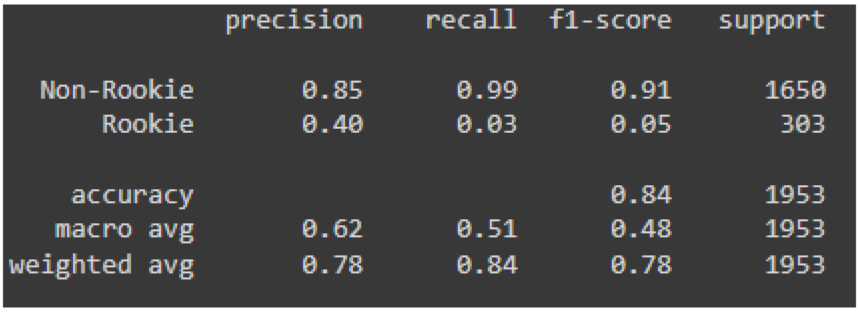
The first step is to manipulate the dataset for the use case, to do that it needs to be grouped by Rookie, Player, and Year so we can see a player's yearly change.

Then, the next step is preparing the test and train data and splitting it for the model. I choose knn model for the job. After semi-guessing the hyperparameters



```
total test value:  1953
total correct ans:  1604
              precision    recall  f1-score   support

  Non-Rookie       0.86      0.95      0.90      1650
      Rookie       0.33      0.14      0.20       303

    accuracy                           0.82      1953
   macro avg       0.59      0.54      0.55      1953
weighted avg       0.77      0.82      0.79      1953
```

It gives this result. It can be seen that model is not able to guess rookie players. I believe that it is because of the unbalance of the data set, not because of the model.



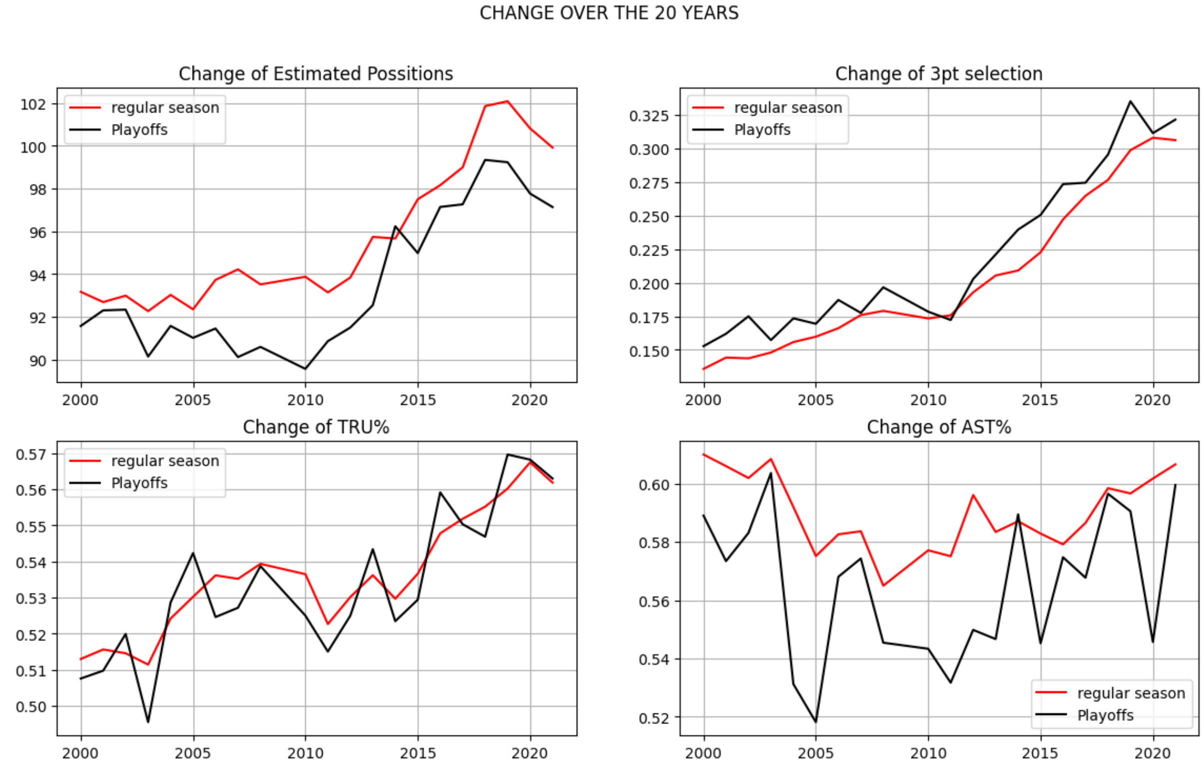
```
              precision    recall  f1-score   support

  Non-Rookie       0.85      0.99      0.91      1650
      Rookie       0.40      0.03      0.05       303

    accuracy                           0.84      1953
   macro avg       0.62      0.51      0.48      1953
weighted avg       0.78      0.84      0.78      1953
```

This result is after the optimization. It gets better slightly. Again it is because of the imbalance of the dataset.

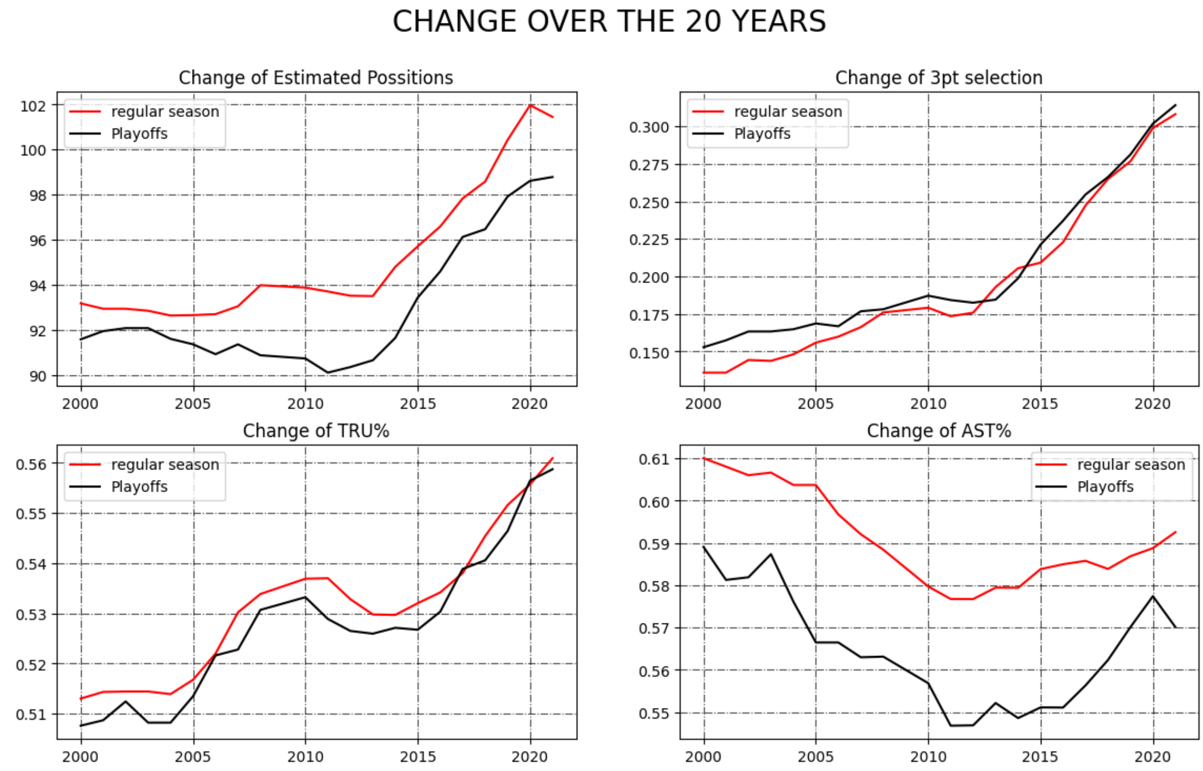
Visualization of the data:

To see the difference between playoffs and regular season games, first I separate those pieces of information from the raw table and then created the data frames called yearly for the regular seasons, playly for the playoffs. After the data sets are set, I defined an estimated position number based on the variables I have on hand. Also, added the percentages as well since they can not be summed.

Some plots are added below:



After the smoothing:



Based on these graphs, once can deduce that

- Since 2010 the pace of the game has increased
- In Playoff there are less possession and also they play less team ball since the AST% drops in playoffs
- Teams have started to take more 3 point shots
- Teams improves their shot selection.

At Last, Guessing if a player plays more than 5 year using pytorch?

It is completely trial and error works, the algorithm is taken directly from the sources listed at the sources. All I have done is to prepare the dataset for this task.
The result is not surprising but I don't understand to set the hyperparameters; no matter what the epoch number is it can directly set its final value. So even if the result makes sense I can't tell if it is overfitted or not.

Sources:
Project:
https://github.com/tuntul17/NBA-Analysis

Smoothing techniques:
https://medium.com/@srv96/smoothing-techniques-for-time-series-data-91cccfd008a2#:~:text=Smoothing%20techniques%20are%20kinds%20of,economy%20compared%20to%20unsmoothed%20data.

Required process:
https://web.archive.org/web/20130723023923/http://www.nba.com/thunder/news/stats101.html
Sklearn optimization:
https://medium.datadriveninvestor.com/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f
Pytorch classification:
https://medium.com/analytics-vidhya/pytorch-for-deep-learning-binary-classification-logistic-regression-382abd97fb43