```
import numpy as np
import pandas as pd

import os

from sklearn import preprocessing
import random
```

```
%%time
train = pd.read_csv("player_stats_2023.csv")

train.head()
```

```
CPU times: user 36.9 ms, sys: 9.4 ms, total: 46.3 ms
Wall time: 47.5 ms
```

	player_id	player_name	player_display_name	position	position_group	headshot_url	recent_team	season	we
0	00-0023459	A.Rodgers	Aaron Rodgers	QB	QB	https://static.www.nfl.com/image/private/f_aut...	NYJ	2023	
1	00-0024243	M.Lewis	Marcedes Lewis	TE	TE	https://static.www.nfl.com/image/private/f_aut...	CHI	2023	
2	00-0024243	M.Lewis	Marcedes Lewis	TE	TE	https://static.www.nfl.com/image/private/f_aut...	CHI	2023	
3	00-0024243	M.Lewis	Marcedes Lewis	TE	TE	https://static.www.nfl.com/image/private/f_aut...	CHI	2023	
4	00-0024243	M.Lewis	Marcedes Lewis	TE	TE	https://static.www.nfl.com/image/private/f_aut...	CHI	2023	

5 rows × 53 columns

```
train.memory_usage(deep = True)  * 1e-6
```

```
Index                          0.000128
player_id                      0.378751
player_name                    0.370080
player_display_name            0.395533
position                       0.333518
position_group                 0.333551
headshot_url                   0.781440
recent_team                    0.337739
season                         0.045224
week                           0.045224
season_type                    0.339428
opponent_team                  0.337708
completions                    0.045224
attempts                       0.045224
passing_yards                  0.045224
passing_tds                    0.045224
interceptions                  0.045224
sacks                          0.045224
sack_yards                     0.045224
sack_fumbles                   0.045224
sack_fumbles_lost              0.045224
passing_air_yards              0.045224
passing_yards_after_catch      0.045224
passing_first_downs            0.045224
passing_epa                    0.045224
passing_2pt_conversions        0.045224
pacr                           0.045224
dakota                         0.045224
carries                        0.045224
rushing_yards                  0.045224
rushing_tds                    0.045224
rushing_fumbles                0.045224
rushing_fumbles_lost           0.045224
rushing_first_downs            0.045224
rushing_epa                    0.045224
rushing_2pt_conversions        0.045224
receptions                     0.045224
targets                        0.045224
receiving_yards                0.045224
receiving_tds                  0.045224
receiving_fumbles              0.045224
receiving_fumbles_lost         0.045224
receiving_air_yards            0.045224
```

```
receiving_yards_after_catch    0.045224
receiving_first_downs          0.045224
receiving_epa                  0.045224
receiving_2pt_conversions      0.045224
racr                           0.045224
target_share                   0.045224
air_yards_share                0.045224
wopr                           0.045224
special_teams_tds              0.045224
fantasy_points                 0.045224
fantasy_points_ppr             0.045224
dtype: float64
```

```
print(train.dtypes)
```

```
player_id                       object
player_name                     object
player_display_name             object
position                        object
position_group                  object
headshot_url                    object
recent_team                     object
season                           int64
week                             int64
season_type                     object
opponent_team                   object
completions                      int64
attempts                         int64
passing_yards                    int64
passing_tds                      int64
interceptions                    int64
sacks                            int64
sack_yards                       int64
sack_fumbles                     int64
sack_fumbles_lost                int64
passing_air_yards                int64
passing_yards_after_catch        int64
passing_first_downs              int64
passing_epa                    float64
passing_2pt_conversions          int64
pacr                           float64
dakota                         float64
carries                          int64
rushing_yards                    int64
rushing_tds                      int64
rushing_fumbles                  int64
rushing_fumbles_lost             int64
rushing_first_downs              int64
rushing_epa                    float64
rushing_2pt_conversions          int64
receptions                       int64
targets                          int64
receiving_yards                  int64
receiving_tds                    int64
receiving_fumbles                int64
receiving_fumbles_lost           int64
receiving_air_yards              int64
receiving_yards_after_catch      int64
receiving_first_downs            int64
receiving_epa                  float64
receiving_2pt_conversions        int64
racr                           float64
target_share                   float64
air_yards_share                float64
wopr                           float64
special_teams_tds                int64
fantasy_points                 float64
fantasy_points_ppr             float64
dtype: object
```

```
print("size before:", train["passing_epa"].memory_usage(deep = True) * 1e-6)
train['passing_epa'] = round(train['passing_epa'])
print(train['passing_epa'])
```

```
size before: 0.045351999999999996
0      -2.0
1       NaN
2       NaN
3       NaN
4       NaN
       ...
5648    NaN
```

```
5649    NaN
5650    NaN
5651    NaN
5652    NaN
Name: passing_epa, Length: 5653, dtype: float64
```

https://github.com/nflverse/nflverse-data/releases/tag/player_stats

2023 player stats csv

```
train = pd.read_csv("player_stats_2023.csv")
print("size before:", train["receiving_epa"].memory_usage(deep=True) * 1e-6)
train["receiving_epa"] = train["receiving_epa"].astype("category")
print("size after :", train["receiving_epa"].memory_usage(deep=True) * 1e-6)
```

```
size before: 0.045351999999999996
size after : 0.18032199999999998
```

```
print(train.memory_usage(deep=True)*1e-6)
print("total:", train.memory_usage(deep=True).sum()*1e-6)
train['passing_epa'] = train['passing_epa'].astype('int')
print("size after:", train["passing_epa"].memory_usage(deep = True) * 1e-6)
```

```
train.to_pickle("train.pkl")
# size is shown in bytes again and needs to be converted to megabytes
print("player_stats_2023.csv:", os.stat('player_stats_2023.csv').st_size * 1e-6)
print("train.pkl:", os.stat('train.pkl').st_size * 1e-6)

```

```
player_stats_2023.csv: 1.696564
train.pkl: 2.306803
```

```
del train
```

Findings: Our attempts to shrink the data ended up increasing its size.

```
%%time
train = pd.read_pickle("train.pkl")

```

```
CPU times: user 4.67 ms, sys: 958 µs, total: 5.63 ms
Wall time: 5.57 ms
```

We tried to adjust the EPA stats from floats to categories, which actually ended up making our data larger than it did before. We also converted to file to a pickle file, which also made the file larger. Next time, we may reevaluating other float objects to see if we can decrease the size of those.