

Making a better USG% metric

USG is currently:

Definition:

- The percentage of team plays used by a player when they are on the floor

Formula:

- $(\text{FGA} + \text{Possession Ending FTA} + \text{TO}) / \text{POSS}$

I would like to make an offensive_load metric that is more of an overall offensive_load rating.

I really don't think turnovers should be included in the USG metric, at least for what I use USG for. So maybe, not a BETTER metric, just a different one, more specific to my needs.

The offensive_load metric would be:

- $(\text{FGA} / \text{TEAM FGA}) + (\text{FTA} / \text{TEAM FTA})$

The offensive_load per minute metric would be:

- $((\text{FGA} / \text{TEAM FGA}) + (\text{FTA} / \text{TEAM FTA})) / \text{MIN}$

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import sqlite3
import seaborn as sns
from matplotlib.pyplot import figure
from bs4 import BeautifulSoup
import time
import requests
import shutil
import datetime
from scipy.stats import norm
import warnings
warnings.filterwarnings('ignore')
import requests
import json
import xgboost
from xgboost import XGBClassifier
from random import randint
import random
import os
os.chdir('C:\\\\Users\\\\Travis\\\\OneDrive\\\\Data Science\\\\Personal_Projects\\\\Sports\\\\NBA')
from cmath import nan
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from sklearn.pipeline import Pipeline, make_pipeline, FeatureUnion
from sklearn.tree import plot_tree
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.compose import make_column_selector as selector, ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler, OrdinalEncoder
import pickle
from sklearn.metrics import fbeta_score
import winsound
from sklearn.linear_model import LinearRegression
from sklearn import tree, preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier, ExtraTreesClassifier
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_curve, auc, f1_score, make_scorer, recall_score
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from matplotlib.offsetbox import OffsetImage, AnnotationBbox
import streamlit as st

today = datetime.datetime.today().strftime('%Y-%m-%d')
```

```
In [ ]: # Load updated player data
trad_player_data = pd.read_csv('data/player/aggregates/Trad&Adv_box_scores_GameView')
# fix date so it is sortable
trad_player_data['trad_game_date'] = pd.to_datetime(trad_player_data['trad_game_date'])
# convert to just date
trad_player_data['trad_game_date'] = trad_player_data['trad_game_date'].dt.date
# to date
#trad_player_data['trad_game_date'] = trad_player_data['trad_game_date'].dt.strftime('%Y-%m-%d')
# sort by date
trad_player_data = trad_player_data.sort_values(by = ['trad_game_date'], ascending=True)
# drop nan for trad_player
trad_player_data = trad_player_data.dropna(subset = ['trad_player'])
trad_player_data.head()
```

Out[]:

		Unnamed: 0	trad_unnamed: 0	trad_player	trad_team	trad_matchup	trad_game_date	trad_w/l	trad
263003	263483	172	Nickeil Alexander-Walker	UTA	UTA @ MIN	2023-01-16	W		
262971	263451	140	Bogdan Bogdanovic	ATL	ATL vs. MIA	2023-01-16	W		
262981	263461	150	Kristaps Porzingis	WAS	WAS vs. GSW	2023-01-16	L		
262980	263460	149	Jordan Poole	GSW	GSW @ WAS	2023-01-16	W		
262979	263459	148	Bam Adebayo	MIA	MIA @ ATL	2023-01-16	L		

5 rows × 53 columns

```
In [ ]: # add team totals for each game for points, rebounds, assists, turnovers, steals, f
trad_player_data = trad_player_data.merge(trad_player_data.groupby(['trad_game_date'])

trad_player_data.head()
```

Out[]:

	Unnamed: 0	trad_unnamed: 0	trad_player	trad_team	trad_match up	trad_game date	trad_w/l	trad_min
0	263483	172	Nickeil Alexander-Walker	UTA	UTA @ MIN	2023-01-16	W	10.0
1	263451	140	Bogdan Bogdanovic	ATL	ATL vs. MIA	2023-01-16	W	31.0
2	263461	150	Kristaps Porzingis	WAS	WAS vs. GSW	2023-01-16	L	37.0
3	263460	149	Jordan Poole	GSW	GSW @ WAS	2023-01-16	W	37.0
4	263459	148	Bam Adebayo	MIA	MIA @ ATL	2023-01-16	L	36.0

5 rows × 60 columns

In []:

```
# add player_points_percentage, player_rebounds_percentage, player_assists_percentage
trad_player_data['player_points_percentage'] = trad_player_data['trad_pts'] / trad_player_data['trad_min']
trad_player_data['player_rebounds_percentage'] = trad_player_data['trad_reb'] / trad_player_data['trad_min']
trad_player_data['player_assists_percentage'] = trad_player_data['trad_ast'] / trad_player_data['trad_min']
trad_player_data['player_turnovers_percentage'] = trad_player_data['trad_tov'] / trad_player_data['trad_min']
trad_player_data['player_steals_percentage'] = trad_player_data['trad_stl'] / trad_player_data['trad_min']
trad_player_data['player_free_throws_attempted_percentage'] = trad_player_data['trad_fta'] / trad_player_data['trad_min']
trad_player_data['player_field_goal_attepted_percentage'] = trad_player_data['trad_fg'] / trad_player_data['trad_min']
```

In []:

```
# add per_min for each of the above
trad_player_data['player_points_percentage_per_min'] = trad_player_data['player_points_percentage'] * trad_player_data['trad_min']
trad_player_data['player_rebounds_percentage_per_min'] = trad_player_data['player_rebounds_percentage'] * trad_player_data['trad_min']
trad_player_data['player_assists_percentage_per_min'] = trad_player_data['player_assists_percentage'] * trad_player_data['trad_min']
trad_player_data['player_turnovers_percentage_per_min'] = trad_player_data['player_turnovers_percentage'] * trad_player_data['trad_min']
trad_player_data['player_steals_percentage_per_min'] = trad_player_data['player_steals_percentage'] * trad_player_data['trad_min']
```

In []:

```
# fix nans with 0
trad_player_data = trad_player_data.fillna(0)
trad_player_data
```

Out[]:

	Unnamed: 0	trad_unnamed: 0	trad_player	trad_team	trad_match up	trad_game date	trad_w/l	trad
0	263483	172	Nickeil Alexander-Walker	UTA	UTA @ MIN	2023-01-16	W	
1	263451	140	Bogdan Bogdanovic	ATL	ATL vs. MIA	2023-01-16	W	
2	263461	150	Kristaps Porzingis	WAS	WAS vs. GSW	2023-01-16	L	
3	263460	149	Jordan Poole	GSW	GSW @ WAS	2023-01-16	W	
4	263459	148	Bam Adebayo	MIA	MIA @ ATL	2023-01-16	L	
...
265291	221	215	Jared Sullinger	BOS	BOS @ MIA	2012-10-30	L	
265292	222	216	Udonis Haslem	MIA	MIA vs. BOS	2012-10-30	W	
265293	223	217	Mike Miller	MIA	MIA vs. BOS	2012-10-30	W	
265294	224	218	LeBron James	MIA	MIA vs. BOS	2012-10-30	W	
265295	206	200	Dwyane Wade	MIA	MIA vs. BOS	2012-10-30	W	

265296 rows × 72 columns

In []:

```
# get this season averages
tpd22 = trad_player_data[trad_player_data['trad_game date'] >= datetime.date(2022,
# get averages
tpd22_avg = tpd22.groupby(['trad_player'])[['trad_pts', 'trad_reb', 'trad_ast', 'tra
tpd22_avg
```

Out[]:

	trad_player	trad_pts	trad_reb	trad_ast	trad_tov	trad_stl	trad_fta	trad_fga	player_pc
0	A.J. Lawson	1.375000	0.500000	0.125000	0.000000	0.000000	0.125000	1.250000	
1	AJ Green	5.047619	1.142857	0.428571	0.190476	0.190476	0.190476	3.761905	
2	AJ Griffin	9.900000	2.200000	1.000000	0.725000	0.825000	0.400000	8.250000	
3	Aaron Gordon	16.736842	6.842105	2.447368	1.605263	0.842105	4.789474	10.736842	
4	Aaron Holiday	4.500000	1.425000	1.475000	0.600000	0.625000	0.650000	4.025000	
...
490	Zach Collins	9.218750	5.781250	2.468750	1.937500	0.375000	2.156250	6.750000	
491	Zach LaVine	24.153846	4.589744	4.358974	2.435897	1.025641	4.948718	18.230769	
492	Zeke Nnaji	4.727273	1.818182	0.242424	0.454545	0.181818	0.909091	3.151515	
493	Ziaire Williams	6.588235	2.941176	1.000000	1.235294	0.470588	0.588235	6.000000	
494	Zion Williamson	26.214286	6.892857	4.714286	3.214286	1.142857	8.500000	16.178571	

495 rows × 20 columns

In []:

```
# rank by player_points_percentage_per_min
tpd22_avg = tpd22_avg.sort_values(by = ['player_field_goal_attepted_percentage'], axis=0).head(50)
```

Out[]:	trad_player	trad_pts	trad_reb	trad_ast	trad_tov	trad_stl	trad_fta	trad_fga	player_poi
312	Luka Doncic	33.33	9.03	8.82	3.74	1.54	11.21	22.64	
231	Joel Embiid	33.97	9.71	4.10	3.65	1.06	11.81	21.00	
307	LeBron James	29.71	8.41	6.97	3.18	1.00	6.09	22.62	
219	Jayson Tatum	31.09	8.26	4.28	2.60	1.02	8.56	21.60	
124	Donovan Mitchell	28.72	3.90	4.72	2.87	1.36	5.92	20.26	
301	Kyrie Irving	25.55	4.81	4.84	2.26	0.77	3.87	19.97	
86	Damian Lillard	28.84	3.90	7.06	3.19	0.87	8.06	19.71	
154	Giannis Antetokounmpo	31.00	11.94	5.34	4.00	0.80	12.94	20.83	
216	Jaylen Brown	27.15	7.08	3.20	3.05	1.08	5.48	20.42	
116	Devin Booker	27.10	4.62	5.55	2.41	0.90	6.21	20.45	
285	Kevin Durant	29.87	6.76	5.34	3.50	0.84	7.42	18.92	
181	Ja Morant	27.34	5.80	7.89	3.49	0.97	8.34	20.77	
302	LaMelo Ball	24.05	5.30	8.30	2.95	1.10	3.20	20.60	
454	Trae Young	27.38	2.82	9.77	3.97	0.92	8.77	20.33	
430	Stephen Curry	29.30	6.43	6.30	3.13	0.93	5.13	20.03	
425	Shai Gilgeous-Alexander	31.08	4.87	5.51	3.15	1.72	10.44	20.26	
23	Anthony Edwards	23.77	5.95	4.41	3.16	1.68	5.36	18.75	
389	Paul George	23.73	5.97	5.13	3.47	1.53	5.03	17.97	
54	Cade Cunningham	19.92	6.17	6.00	3.25	0.83	3.58	18.67	
103	DeMar DeRozan	26.20	5.05	4.95	1.90	0.95	8.02	18.40	
491	Zach LaVine	24.15	4.59	4.36	2.44	1.03	4.95	18.23	
383	Pascal Siakam	25.56	8.21	6.47	2.59	0.82	8.47	18.53	
53	CJ McCollum	21.61	4.71	5.89	2.50	1.03	3.00	18.50	
299	Kyle Kuzma	21.60	7.47	3.91	3.02	0.51	3.70	18.00	
21	Anfernee Simons	21.90	2.90	4.05	2.15	0.76	3.10	17.51	
443	Terry Rozier	20.97	4.18	5.35	2.50	0.82	3.06	19.24	
196	Jalen Green	21.62	4.36	3.57	2.83	0.79	5.90	17.88	
99	De'Aaron Fox	23.79	4.36	6.00	2.62	0.87	5.51	17.56	
471	Tyrese Maxey	20.83	3.04	3.88	1.75	0.88	4.04	16.79	

	trad_player	trad_pts	trad_reb	trad_ast	trad_tov	trad_stl	trad_fta	trad_fga	player_poi
259	Julius Randle	24.20	10.59	3.82	2.70	0.75	7.02	17.89	
92	Darius Garland	21.39	2.81	7.81	3.14	1.17	4.94	16.42	
296	Klay Thompson	20.85	3.91	2.53	1.68	0.59	1.85	18.00	
106	Dejounte Murray	20.38	5.46	6.13	2.41	1.72	2.67	18.03	
468	Tyler Herro	20.72	5.84	4.38	2.44	0.91	2.84	16.72	
271	Keldon Johnson	20.89	4.78	2.73	2.27	0.68	5.14	17.46	
22	Anthony Davis	27.40	12.08	2.64	1.88	1.32	8.04	17.16	
273	Kelly Oubre Jr.	20.17	5.11	1.20	1.26	1.57	4.00	17.83	
194	Jalen Brunson	22.38	3.69	6.31	2.14	0.93	5.26	17.17	
239	Jordan Clarkson	21.09	4.07	4.38	2.87	0.53	3.80	16.87	
41	Bradley Beal	22.88	3.46	5.17	2.75	1.04	4.79	16.42	
382	Paolo Banchero	21.08	6.65	3.84	2.92	1.05	7.76	15.76	
29	Bam Adebayo	21.15	10.18	3.05	2.77	1.10	5.03	16.03	
244	Jordan Poole	21.00	2.61	4.43	3.50	0.77	4.84	16.30	
494	Zion Williamson	26.21	6.89	4.71	3.21	1.14	8.50	16.18	
396	RJ Barrett	20.00	5.46	2.90	2.33	0.46	5.69	16.26	
205	James Harden	22.25	6.46	11.21	3.50	1.32	6.50	15.29	
222	Jerami Grant	21.73	4.32	2.51	1.88	0.88	5.56	15.00	
306	Lauri Markkanen	24.51	8.46	1.90	1.90	0.61	5.68	16.00	
297	Kristaps Porzingis	22.21	8.77	2.36	2.18	0.97	6.95	15.46	
143	Franz Wagner	20.28	3.88	3.33	2.14	0.93	4.49	15.12	

```
In [ ]: # keep just new columns
current_usg_metrics = tpd22_avg[['trad_player', 'player_points_percentage', 'player_assists_percentage', 'player_turnovers_percent', 'player_free_throws_attempted_percentage', 'player_points_per_min', 'player_rebounds_per_min', 'player_turnovers_per_min', 'player_steals_per_min']]
```

```
In [ ]: current_usg_metrics
```

Out[]:

	trad_player	player_points_percentage	player_rebounds_percentage	player_assists_percentage
312	Luka Doncic	0.30	0.23	0.40
231	Joel Embiid	0.30	0.25	0.17
307	LeBron James	0.25	0.19	0.27
219	Jayson Tatum	0.26	0.19	0.16
124	Donovan Mitchell	0.25	0.09	0.20
...
45	Braxton Key	0.01	0.01	0.00
319	Malcolm Hill	0.00	0.01	0.00
14	Alondes Williams	0.00	0.02	0.00
341	Michael Foster Jr.	0.00	0.00	0.00
51	Buddy Boeheim	0.00	0.00	0.00

495 rows × 13 columns

In []:

```
# add percentiles for each metric
current_usg_metrics['player_points_percentage_percentile'] = current_usg_metrics['p
current_usg_metrics['player_rebounds_percentage_percentile'] = current_usg_metrics[
current_usg_metrics['player_assists_percentage_percentile'] = current_usg_metrics['
current_usg_metrics['player_turnovers_percentage_percentile'] = current_usg_metrics[
current_usg_metrics['player_steals_percentage_percentile'] = current_usg_metrics['p
current_usg_metrics['player_free_throws_attempted_percentage_percentile'] = current_
current_usg_metrics['player_field_goal_attepted_percentage_percentile'] = current_u
current_usg_metrics['player_points_percentage_per_min_percentile'] = current_usg_me
current_usg_metrics['player_rebounds_percentage_per_min_percentile'] = current_usg_
current_usg_metrics['player_assists_percentage_per_min_percentile'] = current_usg_m
current_usg_metrics['player_turnovers_percentage_per_min_percentile'] = current_usg_
current_usg_metrics['player_steals_percentage_per_min_percentile'] = current_usg_me
```

In []:

```
current_usg_metrics = current_usg_metrics.sort_values(by = ['player_points_percenta
current_usg_metrics.head(10)
```

```
Out[ ]:   trad_player  player_points_percentage  player_rebounds_percentage  player_assists_percentage
          312        Luka Doncic             0.30                      0.23                      0.18
          231        Joel Embiid             0.30                      0.25                      0.17
          154      Giannis Antetokounmpo            0.27                      0.24                      0.16
          425    Shai Gilgeous-Alexander            0.27                      0.11                      0.15
          219        Jayson Tatum             0.26                      0.19                      0.14
          285        Kevin Durant             0.26                      0.17                      0.14
          307       LeBron James             0.25                      0.19                      0.14
          124      Donovan Mitchell             0.25                      0.09                      0.14
          86        Damian Lillard             0.25                      0.09                      0.14
          430       Stephen Curry             0.25                      0.15                      0.14
```

10 rows × 25 columns

```
In [ ]: # save as current usg metrics
current_usg_metrics.to_csv('data/player/aggregates/current_usg_metrics_'+ today +'.csv')
```