```
import pandas as pd
import numpy as np
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.ensemble import RandomForestRegressor
from sklearn.model selection import train test split, GridSearchCV
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
draft data = pd.read csv("NBADraft.csv")
draft data.shape
(415, 29)
draft data.head()
             School G_college GS MP_college
                                                      FGA FG% college
   Pk
                                                FG
2P
   \
           Kentucky
                                37
                                          34.8
                                                5.5
                                                     11.8
                                                                 0.461
0
   1
                            37
4.5
   2
        Ohio State
                           101
                               94
                                          32.8
                                                5.3
                                                     10.6
                                                                 0.502
1
4.8
2
      Georgia Tech
                               35
                                          27.5
                                                5.0
   3
                            36
                                                      8.1
                                                                 0.611
5.0
3
   5
           Kentucky
                            38
                               37
                                          23.5
                                                5.4
                                                      9.7
                                                                 0.558
5.4
4 7
                            65 65
                                          32.6
                                                5.3
                                                      9.8
        Georgetown
                                                                 0.543
5.2
            TRB college AST college STL BLK TOV
   2PA
                                                       PF
                                                           PTS college
S0S \
8.8
                     4.3
                                  6.5
                                      1.8
                                           0.5
                                                4.0
                                                     1.9
                                                                  16.6
6.82
1 9.1
                                  4.1
                                      1.6 0.7 3.5
                                                     2.7
                     6.8
                                                                  15.0
       . . .
7.86
2 8.1
                     8.4
                                  1.0
                                      0.9 2.1 2.5 2.6
                                                                  12.4
        . . .
9.02
3 9.6
                                      1.0 1.8 2.1 3.2
                     9.8
                                  1.0
                                                                  15.1
6.82
4 9.3
                     8.2
                                  3.2 1.5 1.5 2.9 2.5
                                                                  14.5
9.26
               name
                    lottery
0
          john wall
                        True
1
        evan turner
                        True
2
     derrick favors
                       True
  demarcus cousins
                        True
```

```
4
                   True
      greg monroe
[5 rows x 29 columns]
draft data.columns
'FT',
dtype='object')
len(draft data)
415
draft_data.drop(["name","lottery"],axis=1,inplace=True)
draft data.isna().sum()
Pk
             0
School
             0
             0
G college
             0
MP college
             0
FG
             0
             0
FGA
FG% college
             0
             0
2P
             0
2PA
2P%
             0
3P
             0
3PA
             0
3P% college
            22
FT
             0
FTA
             0
             0
FT% college
             0
0RB
             0
DRB
             0
TRB college
             0
AST college
             0
STL
BLK
             0
TOV
             0
PF
             0
PTS_college
             0
SOS
             0
dtype: int64
draft data.dropna(inplace=True)
```

draft_data.describe()

ECA \	Pk	G_college	GS	MP_college	FG
FGA \ count 393.000000		393.000000	393.000000	393.000000	393.000000
393.000000 mean 28.974555		81.875318	64.643766	28.463104	4.595674
9.706361 std 17.194694		39.879311	33.876431	4.422506	1.190355
2.610318 min 1.000000		3.000000	0.000000	13.200000	1.300000
2.400000 25% 14.000000		38.000000	35.000000	25.900000	3.700000
7.900000 50% 28.000000		75.000000	65.000000	29.100000	4.600000
9.700000 75% 44.000000		117.000000	89.000000	31.800000	5.300000
11.400000 max 60.000000 19.300000		152.000000	147.000000	36.500000	9.000000
	FG%_college	2P	2PA	2P%	ORB
\ count	393.000000	393.00000	393.000000	393.000000	393.000000
mean	0.478646	3.50229	6.721374	0.520654	1.493893
std	0.059706	1.21442	2.164156	0.058541	0.871290
min	0.356000	0.60000	1.200000	0.374000	0.100000
25%	0.438000	2.60000	5.100000	0.483000	0.80000
50%	0.465000	3.30000	6.500000	0.513000	1.300000
75%	0.514000	4.30000	8.200000	0.553000	2.100000
max	0.769000	8.20000	14.300000	0.800000	4.300000
count mean std min 25% 50% 75% max	DRB 393.000000 3.961323 1.382057 1.500000 2.900000 3.700000 4.800000 8.600000	TRB_college 393.000000 5.450891 2.072563 1.900000 3.800000 5.100000 6.800000 11.800000	AST_college 393.000000 2.246050 1.510110 0.200000 1.200000 1.800000 2.900000	393.00000 1.02341 0.44648 0.20000 0.70000 0.90000 1.30000	0 393.000000 0 0.774300 5 0.702057 0 0.000000 0 0.300000 0 0.500000 0 1.100000

```
TOV
                                                                      PF
                                                                                PTS college
                                                                                                                                    SOS
                  393.000000
                                                 393.000000
                                                                                  393.000000
                                                                                                                  393.000000
count
                       1.940967
                                                      2.173028
                                                                                     13.086768
                                                                                                                      7.220967
mean
std
                       0.659140
                                                      0.454103
                                                                                        3.495679
                                                                                                                       3.190615
                                                                                        3.400000
min
                       0.600000
                                                      1.000000
                                                                                                                    -5.410000
25%
                       1.500000
                                                      1.900000
                                                                                     10.400000
                                                                                                                      6.650000
                                                                                     13.100000
50%
                       1.900000
                                                      2.200000
                                                                                                                       7.840000
                                                      2.500000
                                                                                     15.500000
75%
                       2.300000
                                                                                                                       9.160000
                                                      4.100000
                       5.200000
                                                                                     27.400000
                                                                                                                    12.750000
max
[8 rows x 26 columns]
corr matrix = draft data.corr()
plt.figure(figsize=(15, 10))
sns.heatmap(corr matrix,
                               annot=True,
                               linewidths=0.5,
                               fmt= ".2f",
                               cmap="YlGnBu");
                                                                                                                                                                 1.0
              Pk -100 0.45 0.41 0.16 0.28 0.20 0.17 0.24 0.19 0.21 0.05 0.05 0.04 0.26 0.28 0.02 0.14 0.18 0.05 0.19 0.11 0.19 0.13 0.29 0.17
       G_college - 0.45 100 0.87 0.18 0.23 0.22 0.01 0.22 0.23 0.01 0.01 0.04 0.11 0.25 0.29 0.10 0.09 0.19 0.16 0.08 0.09 0.12 0.30 0.16 0.24 0.21
              GS -0.41 0.87 100 0.17 0.05 0.06 0.05 0.00 0.01 -0.06 0.08 0.06 0.05 -0.03 -0.08 0.16 -0.04 -0.03 -0.03 -0.03 -0.08 -0.07 -0.05 0.04 -0.26
                                                                                                                                                                 0.8
      MP_college -0.16 0.18 0.17 100 0.66 0.76 0.33 0.35 0.46 0.26 0.47 0.50 0.10 0.53 0.47 0.31 0.13 0.24 0.10 0.56 0.47 0.17 0.57 0.04 0.72 0.10 FG -0.28 0.23 0.05 0.66 1.00 0.90 0.09 0.80 0.81 0.10 0.28 0.30 0.02 0.66 0.60 0.17 0.20 0.42 0.37 0.24 0.20 0.06 0.52 0.14 0.95 0.19
             FGA -0.20-0.22 0.06 0.76 0.90 1.00 -0.34 0.54 0.68 -0.29 0.53 0.58 0.08 0.65 0.59 0.33 -0.09 0.23 0.11 0.41 0.33 -0.19 0.60
    FG% college -0.17 -0.01 -0.05 -0.33 0.09 -0.34 1.00 0.47 0.19 0.89 -0.61 -0.67 -0.24 -0.03 0.08 -0.43 0.63 0.38 0.52 -0.42 -0.34 0.58 -0.22 0.21 -0.08 -0.09
                                                                                                                                                                 0.6
              2P -0.24 0.22 0.00 0.35 0.80 0.54 0.47 1.00 0.94 0.35 0.35 0.32 0.25 0.59 0.69 0.21 0.59 0.59 0.64 0.01 0.00 0.39 0.39
             2PA -0.19 -0.23 0.01 0.46 0.81 0.68 0.19 0.94 1.00 0.03 -0.24 -0.20 -0.20 0.66 0.74 -0.13 0.46 0.52 0.54 0.12 0.09 0.26
            2P% -0.21 -0.01 -0.06 -0.26 0.10 -0.29 0.89 0.35 0.03 1.00 -0.39 -0.42 -0.19 -0.08 -0.00 -0.30 0.47 0.30 0.39 -0.34 -0.23 0.43 -0.23 0.11 -0.04 -0.09
                                                                                                                                                                 0.4
              3P -0.05 0.01 0.08 0.47 0.28 0.53 0.61 0.35 0.24 0.39 100 0.98 0.38 0.09 0.08 0.60 0.63 0.29 0.46 0.38 0.31 0.53 0.19 0.30 0.44 0.02
             3PA -0.05 0.04 0.06 0.50 0.30 0.58 0.67 0.32 0.20 0.42 0.98 1.00 0.33 0.13 0.03 0.58 0.64 0.28 0.45 0.42 0.35 0.55 0.24 0.28 0.46 0.00
    3P%_college -0.04 0.11 0.05 0.10 0.02 0.08 0.24 0.25 0.20 0.19 0.38 0.33 100 0.05 0.14 0.30 0.32 0.14 0.23 0.11 0.10 0.19 0.04 0.15 0.05 0.05 0.05 FT -0.26 0.25 0.03 0.53 0.66 0.65 0.03 0.59 0.66 0.08 0.09 0.13 0.05 100 0.96 0.30 0.13 0.28 0.24 0.32 0.26 0.00 0.58 0.23 0.79 0.16 FTA -0.28 0.29 0.08 0.47 0.66 0.59 0.08 0.69 0.74 0.00 0.08 0.03 0.14 0.96 100 0.03 0.29 0.37 0.37 0.25 0.25 0.2 0.13 0.59 0.32 0.74 0.15
                                                                                                                                                                 0.2
     FT%_college -0.02 0.10 0.16 0.31 0.17 0.33 0.43 0.21 0.13 0.30 0.60 0.58 0.30 0.30 0.03 1.00 0.49 0.22 0.35 0.28 0.16 0.43 0.06 0.26 0.34 0.03
                                                                                                                                                                 0.0
            ORB -0.14-0.09 -0.04-0.13 0.20 -0.09 0.63 0.59 0.46 0.47 -0.63-0.64-0.32 0.13 0.29 -0.49 1.00 0.67 0.87 -0.52-0.28 0.63 -0.12 0.39 0.05 -0.10
     DRB -0.18-0.19 0.03 0.24 0.42 0.23 0.38 0.59 0.52 0.30 0.29 0.28 0.14 0.28 0.37 0.22 0.67 100 0.95 0.21 0.12 0.52 0.19 0.38 TRB_college -0.18-0.16 0.03 0.10 0.37 0.11 0.52 0.64 0.54 0.39 0.46 0.45 0.23 0.24 0.37 0.35 0.87 0.95 100 0.36 0.20 0.61 0.08 0.42
                                                                                                                                                                 -0.2
     AST_college <0.05-0.08 0.05 0.56 0.24 0.41 0.42 0.01 0.12 0.34 0.38 0.42 0.11 0.32 0.25 0.28 0.52 0.21 0.36 1.00 0.59 0.44 0.64 0.13 0.35 0.03 STL <0.19 0.09 0.03 0.47 0.20 0.33 0.34 0.00 0.09 0.23 0.31 0.35 0.10 0.26 0.22 0.16 0.28 0.12 0.20 0.59 1.00 0.27 0.45 0.07 0.28 0.00
             BLK -0.11-0.12-0.08-0.17 0.06-0.19 0.58 0.39 0.26 0.43 0.53-0.55-0.19 0.00 0.13 0.43 0.63 0.52 0.61 0.44-0.27 1.00 0.11 0.33 0.07-0.14
             TOV -0.19 -0.30 -0.07 0.57 0.52 0.60 -0.22 0.39 0.51 -0.23 0.19 0.24 -0.04 0.58 0.59 0.06 -0.12 0.19 0.08 0.64 0.45 -0.11 1.00 0.30 0.58 -0.10
              PF -0.13-0.16-0.05 0.04 0.14 0.06 0.21 0.33 0.32 0.11 -0.30 -0.28-0.15 0.23 0.32 -0.26 0.39 0.38 0.42 -0.13 0.07 0.33 0.30
     PTS_college -0.29-0.24 0.04 0.72 0.95 0.93 0.08 0.66 0.71 0.04 0.44 0.46 0.05 0.79 0.74 0.34 0.05 0.31 0.23 0.35 0.28 0.07 0.58 0.10 1.00 0.18
             SOS -0.17 0.21 0.26 0.10 0.19 0.14 0.09 0.18 0.17 0.09 0.02 0.00 0.05 0.16 0.15 0.03 0.10 0.21 0.18 0.03 0.00 0.14 0.10 0.08 0.18 1.00
```

TRB_ AST (

PTS

draft_data.drop(["School"],axis=1,inplace=True)

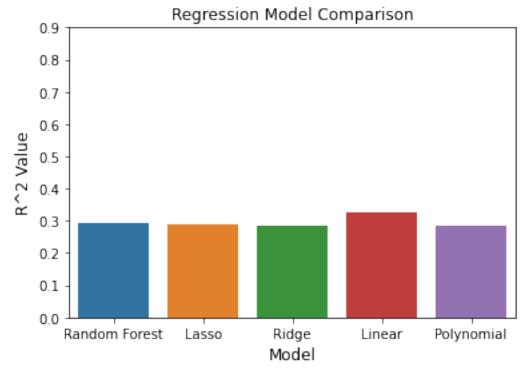
#model with all variables

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

```
from sklearn.model selection import cross_val_score
np.random.seed(2899)
X = draft data.drop("Pk",axis=1)
y = draft data["Pk"]
X_train, X_test,y_train,y_test = train_test_split(X,y,test size=0.2)
rf=RandomForestRegressor().fit(X train,y train)
rfa = rf.score(X_test,y_test)
rfa
0.2949091184120498
#ridge regession
from sklearn.linear model import Ridge
np.random.seed(2899)
ridge=Ridge().fit(X_train,y_train)
ra = ridge.score(X_test,y_test)
ra
0.2831271077839256
#lasso regression
from sklearn.linear_model import Lasso
lasso=Lasso().fit(X train,y train)
la = lasso.score(X test,y test)
la
0.2883184186274941
y preds = rf.predict(X test)
from sklearn.metrics import mean squared error
#rf RMSE
rmse = mean squared error(y test, y preds, squared=False)
rmse
13.767979545288906
y preds = ridge.predict(X test)
rmse = mean squared error(y test, y preds, squared=False)
rmse
13.88253388229793
y preds = lasso.predict(X test)
rmse = mean_squared_error(y_test, y_preds, squared=False)
rmse
13.832176631359042
```

```
#plt.style.use('fivethirtyeight')
models=["Random Forest","Lasso","Ridge","Linear","Polynomial"]
r2 = [rfa,la,ra,.327,.286]
fig=sns.barplot(models, r2)
fig.set_xlabel("Model", fontsize = 12)
fig.set_ylabel("R^2 Value", fontsize = 12)
fig.set_title("Regression Model Comparison")
fig.set_yticks(np.arange(0, 1, .1))
plt.show()
```

/Users/jackpiccione/opt/anaconda3/lib/python3.7/site-packages/ seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation. FutureWarning



```
'BLK', 'TOV', 'PF', 'PTS_college', 'SOS', 'name', 'lottery'],
      dtype='object')
draft dataLOT.drop(["Pk","name"],inplace=True,axis=1)
draft_dataLOT.drop("School",inplace=True,axis=1)
draft dataLOT.dropna(inplace=True)
draft dataLOT.columns
Index(['G_college', 'GS', 'MP_college', 'FG', 'FGA', 'FG%_college',
'2P',
       '2PA', '2P%', '3P', '3PA', '3P%_college', 'FT', 'FTA', 'FT
% college'
       'ORB', 'DRB', 'TRB college', 'AST college', 'STL', 'BLK',
       'PF',
'TOV',
       'PTS college', 'SOS', 'lottery'],
      dtype='object')
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import Ridge
np.random.seed(2899)
X = draft dataLOT.drop("lottery",axis=1)
y = draft dataLOT["lottery"]
X_train, X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
rfc=RandomForestClassifier().fit(X_train,y_train)
rfc.score(X test, y test)
0.7341772151898734
from sklearn.ensemble import GradientBoostingClassifier
np.random.seed(2899)
X = draft dataLOT.drop("lottery",axis=1)
y = draft dataLOT["lottery"]
X train, X test, y train, y test = train test split(X, y, test size=0.2)
gbc=GradientBoostingClassifier().fit(X train,y train)
gbc.score(X test, y test)
0.759493670886076
prospects = pd.read csv("prospects.csv")
prospects.head()
                                                     3P ...
    G GS
             MP
                FG
                       FGA
                              FG%
                                    2P 2PA
                                               2P%
                                                              DRB TRB
AST \
```

```
26.9 5.3
0 32
       31
                       8.8 0.607 4.0 5.5 0.737
                                                    1.3
                                                              8.1 9.9
                                                          . . .
1.9
           33.0
1 39
       39
                 6.3
                      13.2 0.478
                                   5.2 9.8 0.525
                                                    1.1
                                                              6.1
                                                                   7.8
3.2
                 4.9
2 29
           25.3
                       8.1 0.597
                                             0.600
       29
                                   4.9 8.1
                                                    0.0
                                                          . . .
                                                              5.2
                                                                   8.1
1.3
3 34
       34
          32.0
                 3.2
                       8.4 0.384
                                   1.4 3.4
                                             0.426
                                                    1.8
                                                              3.2
                                                                   4.0
1.4
4 39
       25
          24.0
                 3.7
                       7.6 0.493
                                  1.9 3.5 0.547
                                                    1.8
                                                         . . .
                                                              3.2 3.9
1.0
        BLK
             TOV
                   PF
                        PTS
                               SOS
   STL
                                              name
0
  0.8
       3.7
             1.9
                  2.7
                       14.1
                              4.50
                                     chet-holmaren
                              7.26
1
  1.1
       0.9
            2.4
                 1.9
                       17.2
                                    paolo-banchero
  0.8
       2.1
             2.2
                  2.7
                       12.0
                              7.86
                                       jalen-duren
3
  0.7
       0.2
             1.5
                  1.7
                       10.1
                             11.02
                                     caleb-houstan
4 0.5
       0.6
             0.6
                  1.1
                       10.4
                              7.26
                                        aj-griffin
[5 rows x 26 columns]
chl=prospects.loc[[0]].values.flatten().tolist()
name=chl.pop()
achl=[chl]
ynew = rfc.predict(achl)
ynew, name
(array([ True]), 'chet-holmgren')
prospects.dropna(inplace=True)
prospects=prospects.reset index(drop=True)
df list5=[]
df=[]
for i in range(len(prospects)):
    plist=prospects.loc[[i]].values.flatten().tolist()
    name=plist.pop()
    parray=[plist]
    ynew = gbc.predict proba(parray)
    ylist=ynew.tolist()
    ylist=ylist[0]
    #df["name"] = plist.pop() #add player name to dataframe
    #df["lottery prob"] = ylist.pop()
    df.append(name)
    df.append(ylist.pop())
    #df list5.append(df)
    print(ynew,name)
[[0.13694653 0.86305347]] chet-holmgren
[[0.35865844 0.64134156]] paolo-banchero
[[0.32427004 0.67572996]] jalen-duren
```

```
[[0.96324324 0.03675676]] caleb-houstan
[[0.48338117 0.51661883]] aj-griffin
[[0.8507817 0.1492183]] patrick-baldwinjr
[[0.21480413 0.78519587]] jaden-ivey
[[0.85142094 0.14857906]] peyton-watson
[[0.76896919 0.23103081]] tyty-washingtonjr
[[0.49508863 0.50491137]] kennedy-chandler
[[0.14717415 0.85282585]] nolan-hickman
[[0.60389648 0.39610352]] id-davison
[[0.98268289 0.01731711]] allen-flanigan
[[0.97854359 0.02145641]] johnny-juzang
[[0.75616143 0.24383857]] marcus-bagley
[[0.79591399 0.20408601]] julian-champagnie
[[0.70837284 0.29162716]] gabe-brown
[[0.89705985 0.10294015]] mark-williams
[[0.62267199 0.37732801]] matthew-mayer
[[0.66721309 0.33278691]] jabari-walker
[[0.16258231 0.83741769]] keegan-murray
[[0.7946896 0.2053104]] max-abmas
[[0.73710987 0.26289013]] caleb-love
[[0.81059227 0.18940773]] ochai-agbaji
[[0.96737749 0.03262251]] collin-gillespie
[[0.64409835 0.35590165]] kendall-brown
[[0.94542676 0.05457324]] matthew-cleveland
[[0.8549522 0.1450478]] walker-kessler
[[0.98181191 0.01818809]] taevion-kinsey
[[0.96630583 0.03369417]] jamaree-bouyea
[[0.61291264 0.38708736]] drew-timme
[[0.96847587 0.03152413]] josiah-jordan-james
[[0.63746849 0.36253151]] trayce-jackson-davis
[[0.96069979 0.03930021]] dalen-terry
[[0.46760666 0.53239334]] malaki-branham
[[0.92952245 0.07047755]] jalen-wilson
[[0.87886422 0.12113578]] ron-harperir
[[0.13742008 0.86257992]] bryce-mcgowens
[[0.93222982 0.06777018]] ej-liddell
[[0.91489753 0.08510247]] scotty-pippenjr
[[0.29446372 0.70553628]] kofi-cockburn
[[0.97613121 0.02386879]] taran-armstrong
[[0.99069255 0.00930745]] isaiah-mobley
[[0.87483201 0.12516799]] jahvon-guinerly
[[0.99200043 0.00799957]] marcus-sasser
[[0.80705167 0.19294833]] david-roddy
[[0.83822783 0.16177217]] dereon-seabron
[[0.87271505 0.12728495]] hyunjung-lee
[[0.81327779 0.18672221]] justin-lewis
[[0.77390809 0.22609191]] terguavion-smith
[[0.85138221 0.14861779]] jaylin-williams
[[0.86513647 0.13486353]] jaylin-williams
[[0.94872858 0.05127142]] tevin-brown
```

```
[[0.98827423 0.01172577]] iverson-molinar
[[0.0846073 0.9153927]] keon-ellis
[[0.9859432 0.0140568]] jordan-hall
[[0.94123388 0.05876612]] christian-braun
[[0.9266844 0.0733156]] ryan-rollins
[[0.83856882 0.16143118]] blake-wesley
[[0.9858812 0.0141188]] jake-laravia
[[0.9477659 0.0522341]] pete-nance
[[0.98205569 0.01794431]] trevion-williams
[[0.85314669 0.14685331]] josh-minott
[[0.98792285 0.01207715]] andrew-nembhard
[[0.5372853 0.4627147]] tari-eason
[[0.73939688 0.26060312]] trevor-keels
[[0.75953322 0.24046678]] julian-strawther
[[0.98186583 0.01813417]] wendell-moorejr
[[0.98686309 0.01313691]] alondes-williams
[[0.82137082 0.17862918]] christian-koloko
[[0.34967901 0.65032099]] jeremy-sochan
[[0.92336123 0.07663877]] harrison-ingram
[[0.97574703 0.02425297]] tyler-burton
[[0.36153445 \ 0.63846555]] jabari-smith
[[0.81151773 0.18848227]] johnny-davis
def every second element(values):
    second values = []
    for index in range(1, len(values), 2):
        second_values.append(values[index])
    return second values
lot prob list=every second element(df)
res = [i for i in df if i not in lot prob list]
len(res)
75
len(lot prob list)
75
d = {'Prospect':res,'Lottery Probability':lot prob list}
lottery = pd.DataFrame(d)
lottery.sort_values(by=['Lottery_Probability'],ascending=False).head(1
            Prospect Lottery Probability
54
          keon-ellis
                                 0.915393
```

```
0
                                  0.863053
       chet-holmgren
37
      bryce-mcgowens
                                  0.862580
10
       nolan-hickman
                                  0.852826
20
       keegan-murray
                                  0.837418
6
          jaden-ivey
                                  0.785196
40
       kofi-cockburn
                                  0.705536
2
         jalen-duren
                                  0.675730
70
       jeremy-sochan
                                  0.650321
1
      paolo-banchero
                                  0.641342
73
        jabari-smith
                                  0.638466
34
      malaki-branham
                                  0.532393
4
          aj-griffin
                                  0.516619
9
    kennedy-chandler
                                  0.504911
64
          tari-eason
                                  0.462715
feat_importances = pd.Series(gbc.feature_importances_,
index=X.columns)
feat importances.nlargest(10).plot(kind='barh')
plt.style.use('fivethirtyeight')
plt.title("Top 10 important features")
plt.show()
```

