Predicting NBA Players' 2k21 Ratings

By Ben Huston and Eli Standard

Motivation for our Project

Problem Statement

Are NBA statistics good predictors for overall (OVR) scores for professional basketball players in the video game NBA 2k21?

- **Predict an NBA player's rating** in the 2k21 video game **based on real stats** from the 2020-2021 NBA season.
- NBA statistics will be predictor variables for a **linear regression model.**
- Fit two different linear models based on defensive and offensive statistics.
- **Draw conclusions** about whether defensive or offensive statistics are more important when we rate superstars in the NBA.
- Final Objective: Predict the NBA 2k21 overall ratings for **rookies** using our linear model - could provide insight into which rookies might develop into NBA stars.

Problem Statement

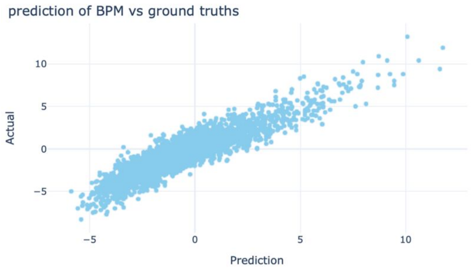
Are NBA statistics good predictors for overall (OVR) scores for professional basketball players in the video game NBA 2k21?

Combine 2 Datasets:

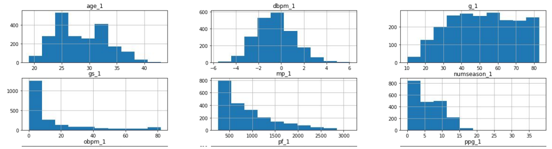
Real NBA Stats	+	Video Game OVR Scores
basketballreference.com		**Scraped**

Related Work

1. Learn linear regression using scikit-learn and NBA data: Data science with sports.



2. What makes a player score? NBA ppg predictions from a college data scientist.



3. Using NCAA Stats to Predict NBA Draft Order.

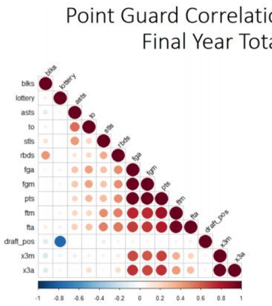


Figure A13

Data Collection and Cleaning

Real NBA Stats - **www.basketball-reference.com**

2020-21 season, Per-Game Stats

Player Per Game Share & Export ▾ ☑ When table is sorted, hide non-qualifiers for rate stats Glossary Hide Partial Rows

Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
1	Precious Achiuwa	PF	21	MIA	61	4	12.1	2.0	3.7	.544	0.0	0.0	.000	2.0	3.7	.546	.544	0.9	1.8	.509	1.2	2.2	3.4	0.5	0.3	0.5	0.7	1.5	5.0
2	Jaylen Adams	PG	24	MIL	7	0	2.6	0.1	1.1	.125	0.0	0.3	.000	0.1	0.9	.167	.125	0.0	0.0		0.0	0.4	0.4	0.3	0.0	0.0	0.0	0.1	0.3
3	Steven Adams	C	27	NOP	58	58	27.7	3.3	5.3	.614	0.0	0.1	.000	3.3	5.3	.620	.614	1.0	2.3	.444	3.7	5.2	8.9	1.9	0.9	0.7	1.3	1.9	7.6
4	Bam Adebayo	C	23	MIA	64	64	33.5	7.1	12.5	.570	0.0	0.1	.250	7.1	12.4	.573	.571	4.4	5.5	.799	2.2	6.7	9.0	5.4	1.2	1.0	2.6	2.3	18.7
5	LaMarcus Aldridge	C	35	TOT	26	23	25.9	5.4	11.4	.473	1.2	3.1	.388	4.2	8.3	.505	.525	1.6	1.8	.872	0.7	3.8	4.5	1.9	0.4	1.1	1.0	1.8	13.5
5	LaMarcus Aldridge	C	35	SAS	21	18	25.9	5.5	11.8	.464	1.3	3.6	.360	4.2	8.2	.509	.518	1.5	1.8	.838	0.8	3.7	4.5	1.7	0.4	0.9	1.0	1.7	13.7
5	LaMarcus Aldridge	C	35	BRK	5	5	26.0	5.0	9.6	.521	0.8	1.0	.800	4.2	8.6	.488	.563	2.0	2.0	1.000	0.4	4.4	4.8	2.6	0.6	2.2	1.4	2.2	12.8
6	Ty-Shon Alexander	SG	22	PHO	15	0	3.1	0.2	0.8	.250	0.1	0.6	.222	0.1	0.2	.333	.333	0.1	0.1	.500	0.1	0.5	0.7	0.4	0.0	0.1	0.2	0.1	0.6
7	Nickeil Alexander-Walker	SG	22	NOP	46	13	21.9	4.2	10.0	.419	1.7	4.8	.347	2.5	5.2	.485	.502	1.0	1.4	.727	0.3	2.8	3.1	2.2	1.0	0.5	1.5	1.9	11.0
8	Grayson Allen	SG	25	MEM	50	38	25.2	3.5	8.3	.418	2.1	5.5	.391	1.3	2.8	.471	.547	1.6	1.8	.868	0.4	2.8	3.2	2.2	0.9	0.2	1.0	1.4	10.6
9	Jarrett Allen	C	22	TOT	63	45	29.6	4.7	7.7	.618	0.1	0.3	.316	4.6	7.3	.631	.624	3.2	4.6	.703	3.1	6.9	10.0	1.7	0.5	1.4	1.6	1.5	12.8
9	Jarrett Allen	C	22	BRK	12	5	26.7	3.7	5.4	.677	0.0	0.0		3.7	5.4	.677	.677	3.8	5.1	.754	3.2	7.3	10.4	1.7	0.6	1.6	1.8	1.8	11.2
9	Jarrett Allen	C	22	CLE	51	40	30.3	5.0	8.2	.609	0.1	0.4	.316	4.9	7.8	.623	.616	3.1	4.5	.690	3.1	6.8	9.9	1.7	0.5	1.4	1.5	1.5	13.2
10	Al-Farouq Aminu	PF	30	TOT	23	14	18.9	1.7	4.3	.384	0.3	1.6	.216	1.3	2.7	.484	.424	0.8	1.0	.818	1.0	3.8	4.8	1.3	0.8	0.4	1.2	1.3	4.4
10	Al-Farouq Aminu	PF	30	ORL	17	14	21.6	2.1	5.2	.404	0.4	1.8	.226	1.7	3.4	.500	.444	0.8	1.0	.824	1.2	4.2	5.4	1.7	1.0	0.5	1.5	1.3	5.5
10	Al-Farouq Aminu	PF	30	CHI	6	0	11.2	0.3	1.7	.200	0.2	1.0	.167	0.2	0.7	.250	.250	0.7	0.8	.800	0.3	2.8	3.2	0.3	0.3	0.0	0.5	1.2	1.5

700 Players,   30 Features

STL% BLK% DWS DBPM

Rk	Player	STL%	BLK%	DWS	DBPM
1	Precious Achiuwa	1.3	4.0	1.0	-0.5
2	Jaylen Adams	0.0	0.0	0.0	-4.6
3	Steven Adams	1.6	2.2	1.7	0.1
4	Bam Adebayo	1.7	3.2	3.2	2.0
5	LaMarcus Aldridge	0.8	3.7	0.6	-0.2
5	LaMarcus Aldridge	0.7	2.8	0.5	-0.7
5	LaMarcus Aldridge	1.1	7.4	0.2	2.1
6	Ty-Shon Alexander	0.0	1.9	0.0	-1.7
7	Nickeil Alexander-Walker	2.2	2.1	1.0	0.1
8	Grayson Allen	1.7	0.6	1.2	0.1
9	Jarrett Allen	0.8	4.3	2.1	-0.2
9	Jarrett Allen	1.1	5.2	0.5	0.9
9	Jarrett Allen	0.8	4.1	1.6	-0.4
10	Al-Farouq Aminu	2.1	1.9	0.5	1.1
10	Al-Farouq Aminu	2.3	2.2	0.4	1.3
10	Al-Farouq Aminu	1.4	0.0	0.1	0.2
11	Kyle Anderson	2.1	2.7	2.7	1.9
12	Giannis Antetokounmpo	1.7	3.2	3.3	2.8

+ 4 Features

Data Collection and Cleaning

Basic Stats

	Rk	Player	Pos	Age	Tm	G	GS
0	1	Precious Achiuwa\achiupr01	PF	21.0	MIA	61.0	4.0
1	2	Jaylen Adams\adamsja01	PG	24.0	MIL	7.0	0.0
2	3	Steven Adams\adamsst01	C	27.0	NOP	58.0	58.0
3	4	Bam Adebayo\adebaba01	C	23.0	MIA	64.0	64.0
4	5	LaMarcus Aldridge\aldrila01	C	35.0	TOT	26.0	23.0

	...	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
0	...	0.509	1.2	2.2	3.4	0.5	0.3	0.5	0.7	1.5	5.0
1	...	NaN	0.0	0.4	0.4	0.3	0.0	0.0	0.0	0.1	0.3
2	...	0.444	3.7	5.2	8.9	1.9	0.9	0.7	1.3	1.9	7.6
3	...	0.799	2.2	6.7	9.0	5.4	1.2	1.0	2.6	2.3	18.7
4	...	0.872	0.7	3.8	4.5	1.9	0.4	1.1	1.0	1.8	13.5

Advanced Stats

	Rk	Player	STL%	BLK%	DWS	DBPM
0	1	Precious Achiuwa\achiupr01	1.3	4.0	1.0	-0.5
1	2	Jaylen Adams\adamsja01	0.0	0.0	0.0	-4.6
2	3	Steven Adams\adamsst01	1.6	2.2	1.7	0.1
3	4	Bam Adebayo\adebaba01	1.7	3.2	3.2	2.0
4	5	LaMarcus Aldridge\aldrila01	0.8	3.7	0.6	-0.2
...	...	...	...	...	...	...
700	536	Delon Wright\wrighde01	3.0	1.3	0.5	0.6
701	537	Thaddeus Young\youngth01	2.2	2.1	2.2	1.4
702	538	Trae Young\youngtr01	1.2	0.5	1.3	-1.7
703	539	Cody Zeller\zelleco01	1.3	1.7	1.1	-0.2
704	540	Ivica Zubac\zubaciv01	0.7	3.4	2.1	0.4

705 rows × 6 columns

Data Collection and Cleaning

Combined, Basic + Advanced

	Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	...	AST	STL	BLK	TOV	PF	PTS	STL%	BLK%	DWS	DBPM
0	1	Precious Achiuwa\achiupr01	PF	21.0	MIA	61.0	4.0	12.1	2.0	3.7	...	0.5	0.3	0.5	0.7	1.5	5.0	1.3	4.0	1.0	-0.5
1	2	Jaylen Adams\adamsja01	PG	24.0	MIL	7.0	0.0	2.6	0.1	1.1	...	0.3	0.0	0.0	0.0	0.1	0.3	0.0	0.0	0.0	-4.6
2	3	Steven Adams\adamsst01	C	27.0	NOP	58.0	58.0	27.7	3.3	5.3	...	1.9	0.9	0.7	1.3	1.9	7.6	1.6	2.2	1.7	0.1
3	4	Bam Adebayo\adebaba01	C	23.0	MIA	64.0	64.0	33.5	7.1	12.5	...	5.4	1.2	1.0	2.6	2.3	18.7	1.7	3.2	3.2	2.0
4	5	LaMarcus Aldridge\aldrila01	C	35.0	TOT	26.0	23.0	25.9	5.4	11.4	...	1.9	0.4	1.1	1.0	1.8	13.5	0.8	3.7	0.6	-0.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
700	536	Delon Wright\wrighde01	PG	28.0	SAC	27.0	8.0	25.8	3.9	8.3	...	3.6	1.6	0.4	1.3	1.1	10.0	3.0	1.3	0.5	0.6
701	537	Thaddeus Young\youngth01	PF	32.0	CHI	68.0	23.0	24.3	5.4	9.7	...	4.3	1.1	0.6	2.0	2.2	12.1	2.2	2.1	2.2	1.4
702	538	Trae Young\youngtr01	PG	22.0	ATL	63.0	63.0	33.7	7.7	17.7	...	9.4	0.8	0.2	4.1	1.8	25.3	1.2	0.5	1.3	-1.7
703	539	Cody Zeller\zelleco01	C	28.0	CHO	48.0	21.0	20.9	3.8	6.8	...	1.8	0.6	0.4	1.1	2.5	9.4	1.3	1.7	1.1	-0.2
704	540	Ivica Zubac\zubaciv01	C	23.0	LAC	72.0	33.0	22.3	3.6	5.5	...	1.3	0.3	0.9	1.1	2.6	9.0	0.7	3.4	2.1	0.4

705 rows × 34 columns

Data Collection and Cleaning

```
# Create a new column in the new dataframe that is just the player's name (without the 9 character ID code)
dfNew['Name'] = dfNew['Player'].str[:-10]
# See if it worked
print(dfNew.iloc[1:5, :])
dfNew.info()
```

```
0          Precious Achiuwa\achiupr01
1              Jaylen Adams\adamsja01
2              Steven Adams\adamsst01
3               Bam Adebayo\adebaba01
4          LaMarcus Aldridge\aldrila01
                     ...
701          Thaddeus Young\youngth01
702              Trae Young\youngtr01
703             Cody Zeller\zelleco01
704             Ivica Zubac\zubaciv01
705                               NaN
Name: Player, Length: 706, dtype: object
```

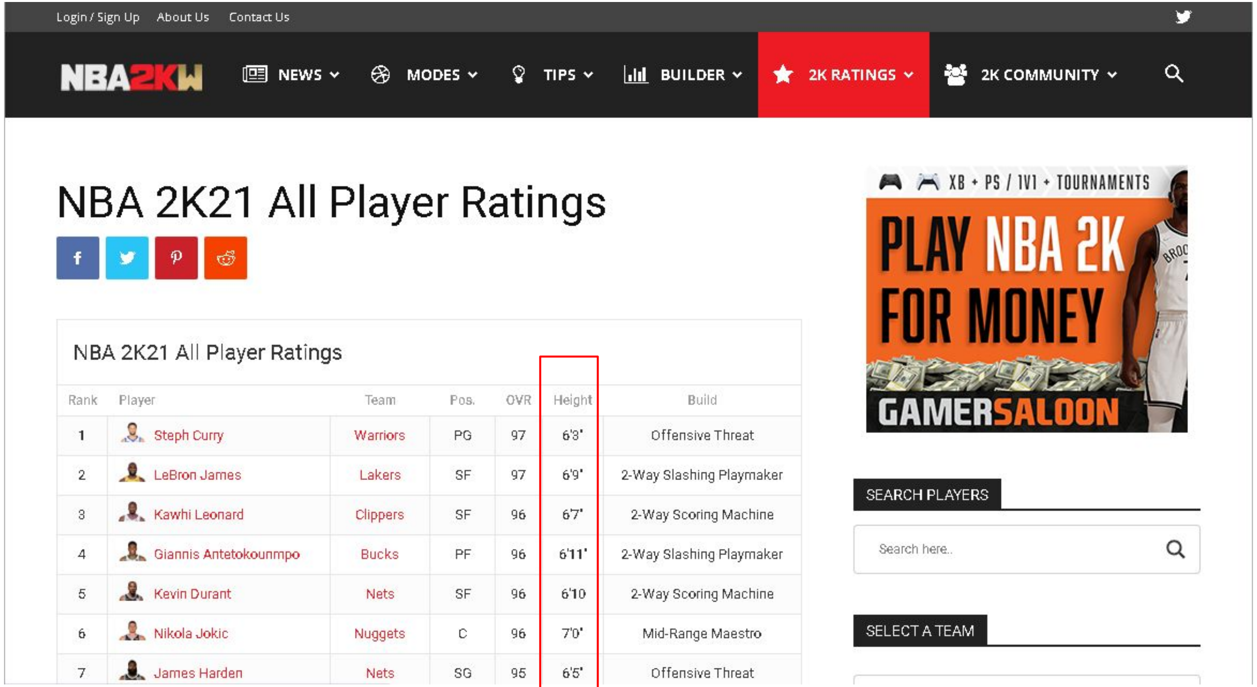
Clean Names

```
0          precious achiuwa
1              jaylen adams
2              steven adams
3               bam adebayo
4          lamarcus aldridge
                 ...
700            delon wright
701          thaddeus young
702              trae young
703             cody zeller
704             ivica zubac
Name: Name, Length: 705, dtype: object
```

Data Collection and Cleaning

Video Game OVR Scores

Website 1



NBA 2K21 All Player Ratings

Rank	Player	Team	Pos.	OVR	Height	Build
1	Steph Curry	Warriors	PG	97	6'3'	Offensive Threat
2	LeBron James	Lakers	SF	97	6'9'	2-Way Slashing Playmaker
3	Kawhi Leonard	Clippers	SF	96	6'7'	2-Way Scoring Machine
4	Giannis Antetokounmpo	Bucks	PF	96	6'11'	2-Way Slashing Playmaker
5	Kevin Durant	Nets	SF	96	6'10	2-Way Scoring Machine
6	Nikola Jokic	Nuggets	C	96	7'0'	Mid-Range Maestro
7	James Harden	Nets	SG	95	6'5'	Offensive Threat

Data Collection and Cleaning

Individual Player HTML Entry:

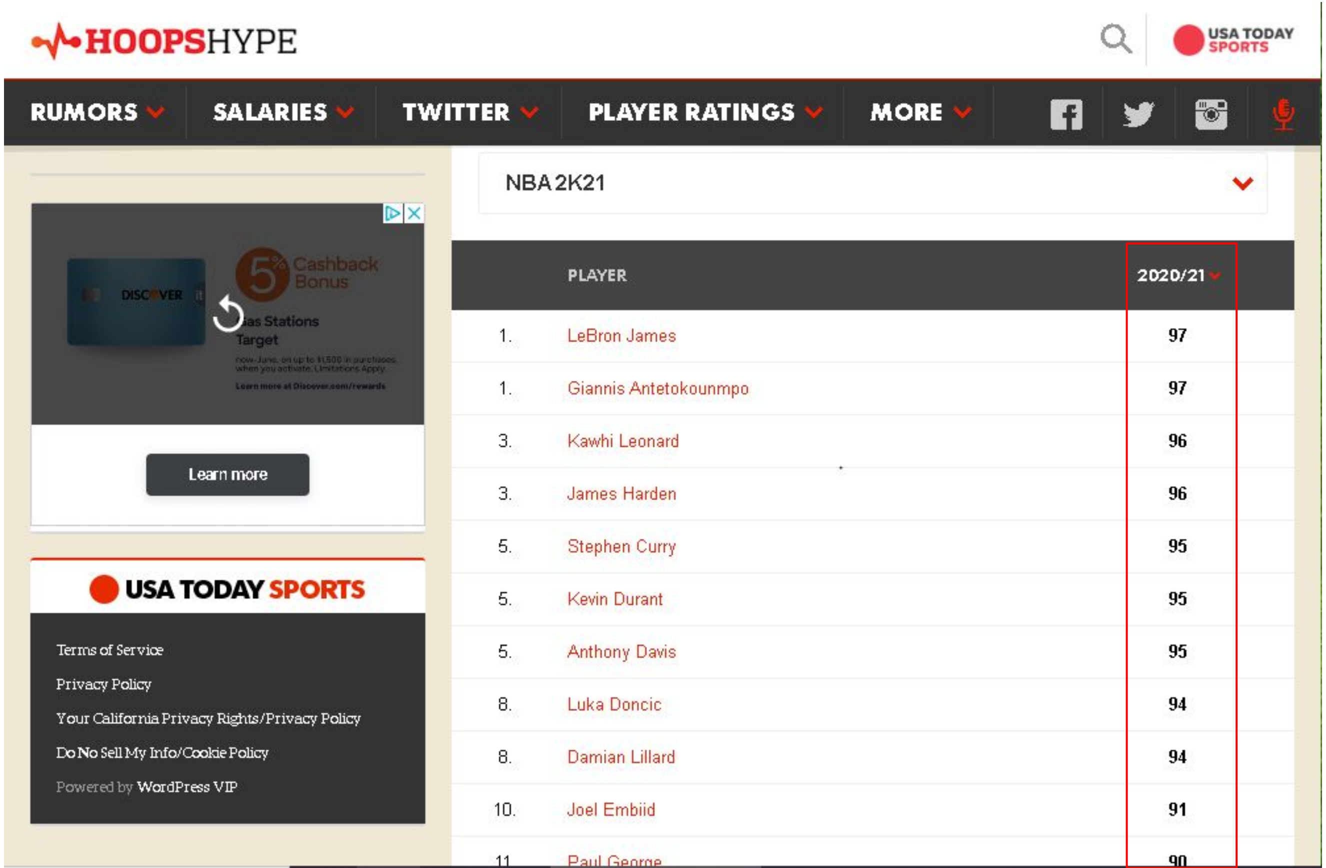
```
<tr>
<td class='3D"rank"' data-value='3D""'>1.</td>
<td class='3D"name"' data-value='3D""'>
<a href='3D"https://hoopshype.com/player/giannis-antetokounmp=' o="">
                                                                            Giannis Antetokounmpo
</a>
</td>
<td class='3D"value' data-value='3D"97"' hh-salaries-sorted="">

</td>
</tr>
```

Extremely messy, too hard to scrape

Data Collection and Cleaning

Website 2



Data Collection and Cleaning

Scrape OVR Score

```
# Start with OVR Score
overallrows = soup.find_all(lambda tag: tag.name == 'td' and tag.get('class') == ['3D"value' ])
i=0
for ovr in overallrows:
    print(ovr)
    text = ovr.renderContents()
    overallrows[i]=text
    i=i+1
```

```
# Write a regex to get rid of leading and trailing stuff around the two digits that make up the OVR
for ovr in overallrows:
    ovr=str(ovr)
    regex=r"[0-9]*[0-9]"
    ovr=re.search(regex,ovr).group(0)
    print(ovr)
```

Data Collection and Cleaning

Scrape Player Name

```
# Now work on getting the player name
namerows = soup.find_all(lambda tag: tag.name == 'td' and tag.get('class') == ['3D"name"'])
# Need to find a way to get the 'a' tag from the td.
for name in namerows:
    for a in name.find_all('a', href=True):
        try:
            a=str(a)
            regex=r"[a-z]+-[a-z]+"
            regexName = re.search(regex, a).group(0)
            regexName = str(regexName)
            regexName = regexName.replace("-", " ")
            print(regexName)
        except AttributeError:
            a=str(a)
            regex= r"[a-z]+-[a-z]+"
            regexName = re.search(regex, a)
            regexName = str(regexName)
            regexName = regexName.replace("-", " ")
            print(regexName)
```

Data Collection and Cleaning

```python
list_names = []
list_ovr = []

for name in namerows:
    for a in name.find_all('a', href=True):
        try:
            a=str(a)
            regex= r"[a-z]+-[a-z]+"
            urlname = re.search(regex, a).group(0)
            urlname = str(urlname)
            urlname = urlname.replace("-", " ")
            #Add it to the dataframe
            list_names.append({'Name': urlname})

        except AttributeError:
            a=str(a)
            regex= r"[a-z]+-[a-z]+"
            urlname = re.search(regex, a)
            urlname = str(urlname)
            urlname = urlname.replace("-", " ")
            #Add it to the dataframe
            list_names.append({'Name': urlname})
for ovr in overallrows:
    ovr=str(ovr)
    #now need to write a regex to get rid of leading and trailing stuff around the two #'s
    regex=r"[0-9]*[0-9]"
    ovr=re.search(regex,ovr).group(0)
    list_ovr.append(ovr)
```

	Name	OVR
0	lebron james	97
1	giannis antetokounmpo	97
2	kawhi leonard	96
3	james harden	96
4	stephen curry	95
...	...	...
580	marial shayok	67
581	justin robinson	67
582	robert franks	67
583	antonius cleveland	67
584	kobi simmons	66

585 rows × 2 columns

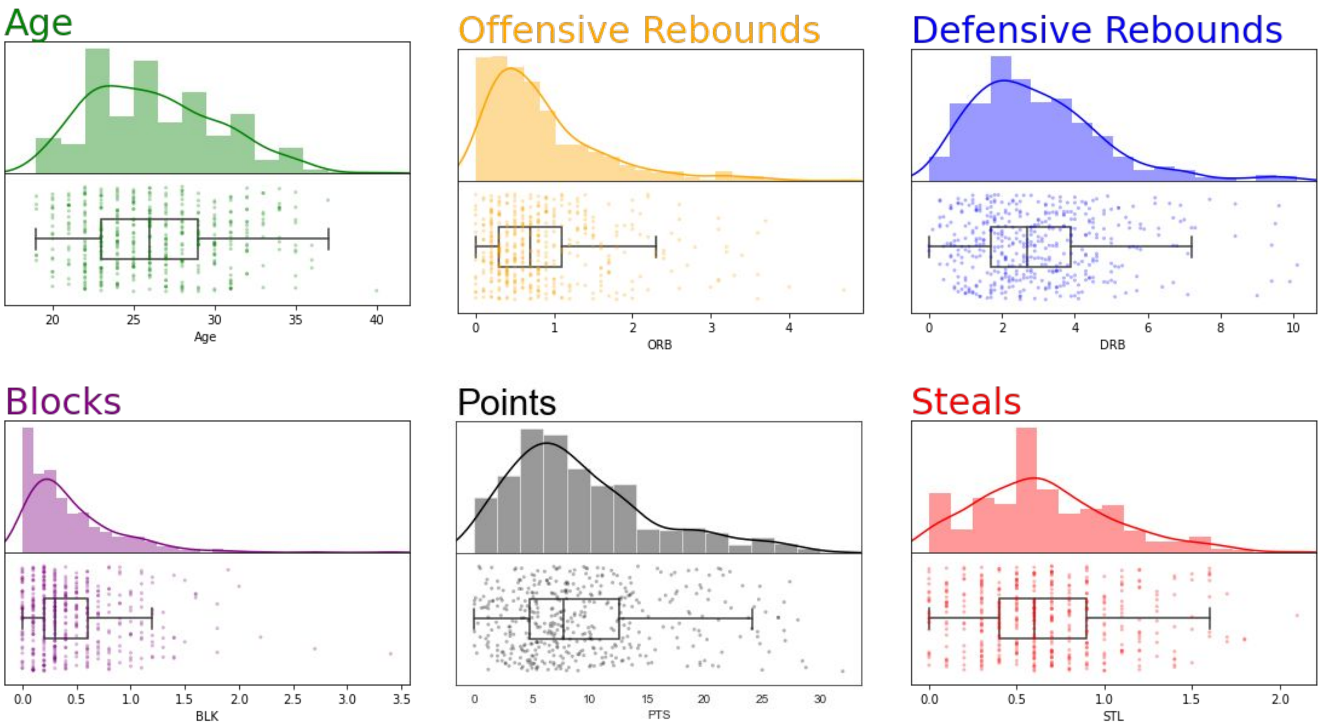
Data Collection and Cleaning

Merged and Cleaned

	Name	OVR	Pos	Tm	Age	G	GS	MP	FG	FGA	...	AST	STL	BLK	STL%	BLK%	DWS	DBPM	TOV	PF	PTS
0	lebron james	97	PG	LAL	36.0	45.0	45.0	33.4	9.4	18.3	...	7.8	1.1	0.6	1.6	1.5	2.6	2.3	3.7	1.6	25.0
1	giannis antetokounmpo	97	PF	MIL	26.0	61.0	61.0	33.0	10.3	18.0	...	5.9	1.2	1.2	1.7	3.2	3.3	2.8	3.4	2.8	28.1
2	kawhi leonard	96	SF	LAC	29.0	52.0	52.0	34.1	8.9	17.5	...	5.2	1.6	0.4	2.3	1.1	2.4	1.3	2.0	1.6	24.8
3	james harden	96	PG-SG	TOT	31.0	44.0	43.0	36.6	7.8	16.7	...	10.8	1.2	0.8	1.6	1.8	1.7	1.0	4.0	2.3	24.6
4	james harden	96	SG	HOU	31.0	8.0	8.0	36.3	7.5	16.9	...	10.4	0.9	0.8	1.1	1.8	0.2	-0.5	4.3	1.8	24.8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
451	jared harper	67	PG	NYK	23.0	8.0	0.0	2.0	0.0	0.5	...	0.1	0.0	0.0	0.0	0.0	0.0	-5.5	0.4	0.1	0.4
452	jarrell brantley	67	PF	UTA	24.0	28.0	0.0	4.9	0.9	1.9	...	0.5	0.3	0.1	2.5	1.2	0.2	1.6	0.3	0.6	2.3
453	gabe vincent	67	PG	MIA	24.0	50.0	7.0	13.1	1.8	4.7	...	1.3	0.4	0.0	1.6	0.3	0.5	-0.9	0.7	1.6	4.8
454	justin robinson	67	PG	OKC	23.0	9.0	0.0	9.8	0.8	2.3	...	1.0	0.3	0.0	1.6	0.0	0.0	-1.0	0.2	1.1	2.3
455	robert franks	67	PF	ORL	24.0	7.0	0.0	14.4	1.9	4.0	...	0.7	0.4	0.4	1.4	2.7	0.1	-0.8	0.3	1.1	6.1

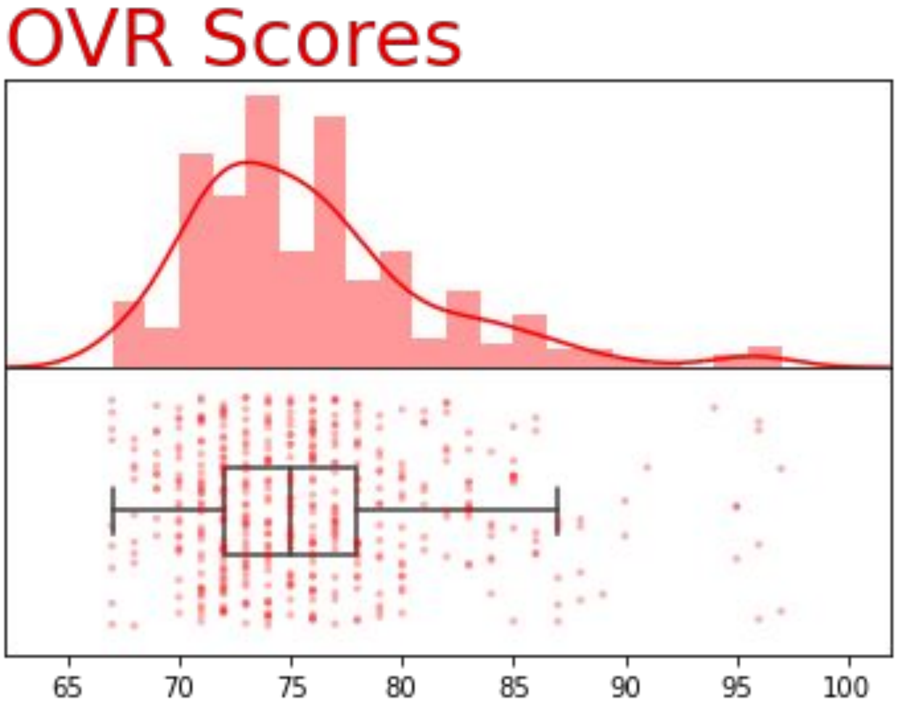
456 rows × 34 columns

Data Exploration



Right skew

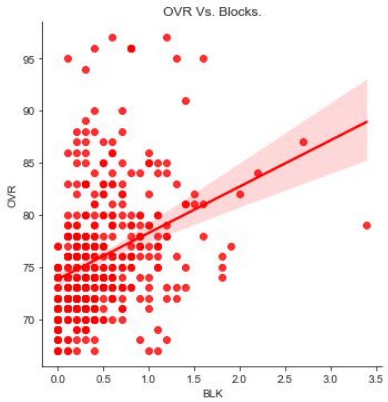
Data Exploration

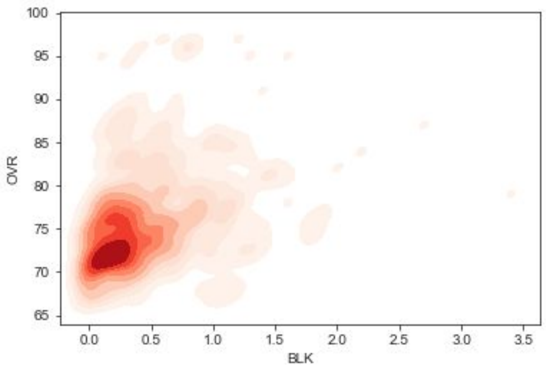


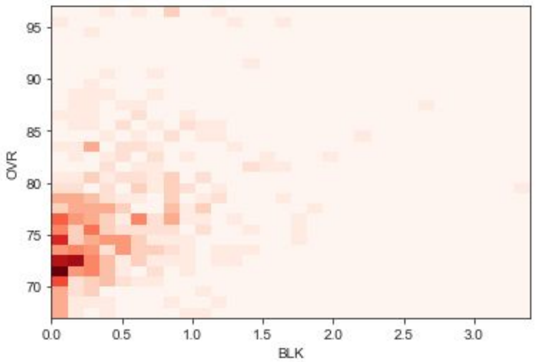
	OVR
count	456.000000
mean	75.875000
std	5.613147
min	67.000000
25%	72.000000
50%	75.000000
75%	78.000000
max	97.000000

Data Exploration

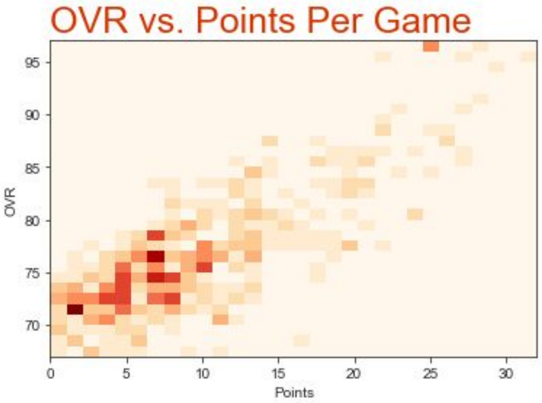
400 points, Overplotting

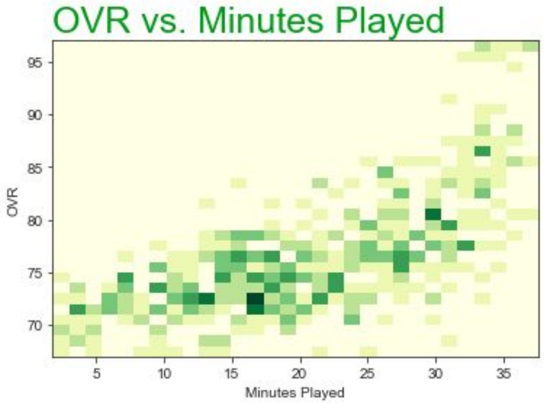


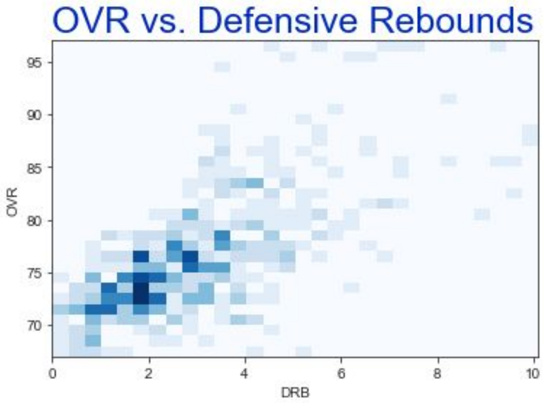




Data Exploration

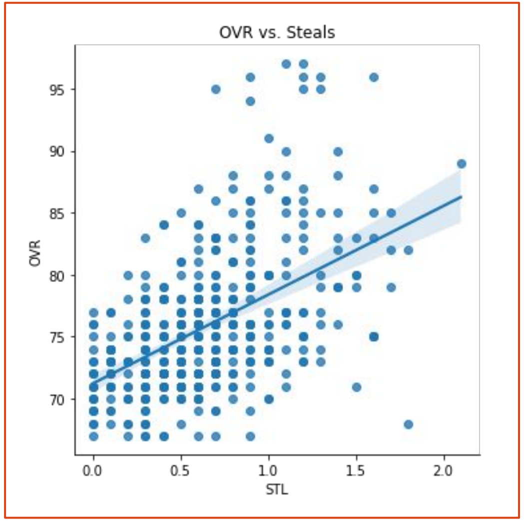






Data Exploration

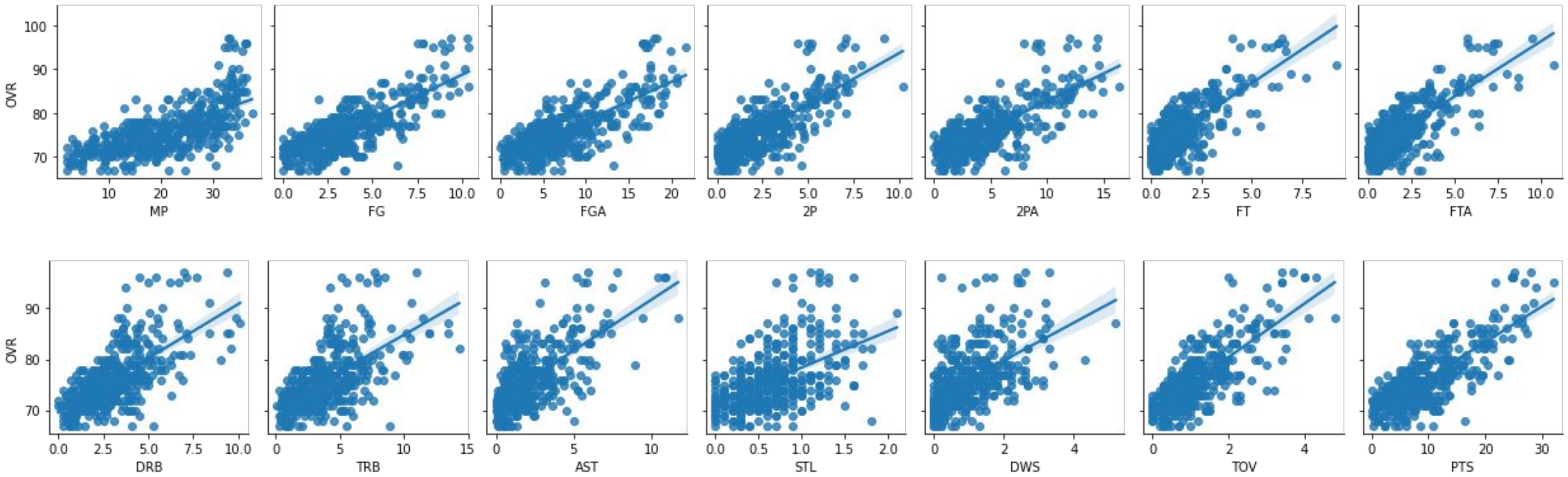
OVR vs. Steals



OVR vs. All Variables

Data Exploration

14 Most Linear Features, **Fit a Linear Model**



Minutes played, Field goals, Field goal attempts, 2 pointers, 2 point attempts, Free throws, Free throw attempts, Defensive rebounds, Total rebounds, Assists, Steals, Defensive win share, Turnovers, Points

Modeling

```python
#Now that our data was fully cleaned, we began the process of fitting a linear model.
from sklearn.model_selection import train_test_split
train, val = train_test_split(dfCleaned, train_size=0.8, random_state=42)
```

```python
train # 336 rows × 31 columns
val # 85 rows × 31 columns
```

6 Defensive Features

```python
# Split the data into dataframes of predicting variables and the OVR that we will attempt to predict.
def select_columns(data, *columns):
    return data.loc[:, columns]

X_train = select_columns(train,
                         'STL%',
                         'BLK%',
                         'DWS',
                         'DBPM',
                         'STL',
                         'BLK')
Y_train = train.loc[:,'OVR']

X_val = select_columns(val,
                         'STL%',
                         'BLK%',
                         'DWS',
                         'DBPM',
                         'STL',
                         'BLK')
y_val = val.loc[:,'OVR']
```

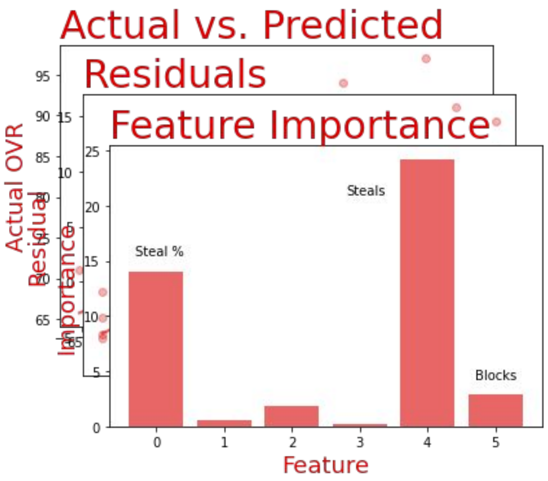
Modeling

```
# Import linear model and Standard Scaler to normalize the data
from sklearn import linear_model as lm
from sklearn.preprocessing import StandardScaler
linear_model = lm.LinearRegression()
#Now normalize the data
scaler = StandardScaler()
scaler.fit(X_train)
X_train=scaler.transform(X_train)
#Normalize the validation data
scaler2 = StandardScaler()
scaler2.fit((X_val))
X_val=scaler2.transform(X_val)


linear_model.fit(X_train, Y_train)
y_fitted = linear_model.predict(X_train)
y_predicted = linear_model.predict(X_val)
```

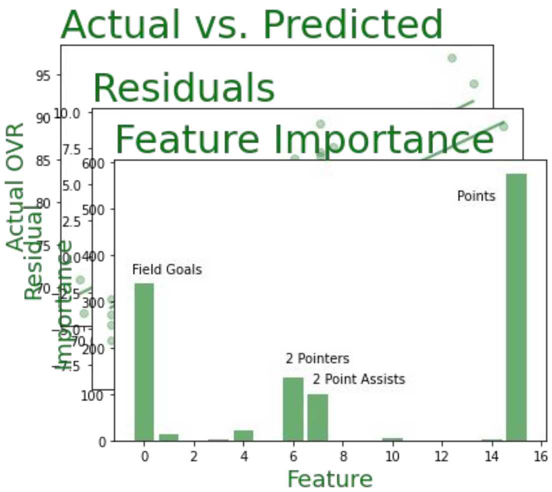
Modeling

Defense



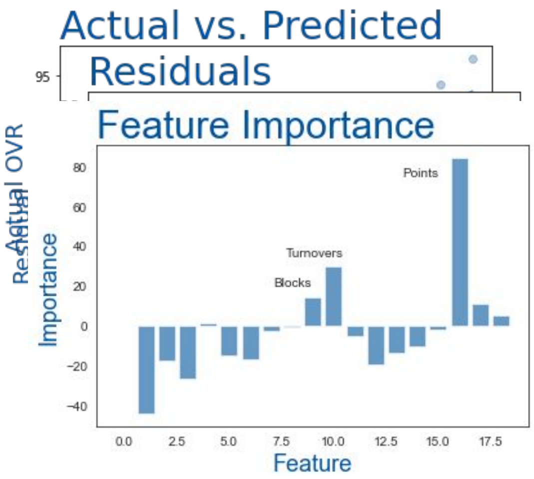
6 features

Offense



16 features

Mixed



19 features

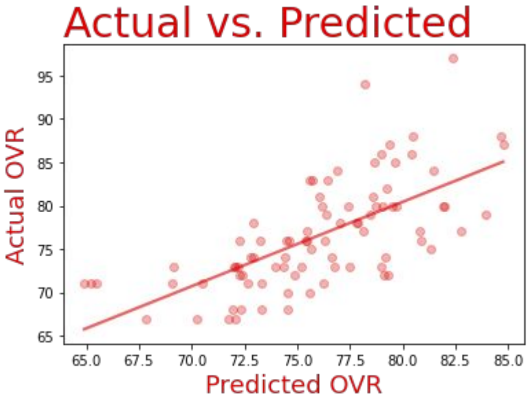
Data Exploration and Modeling

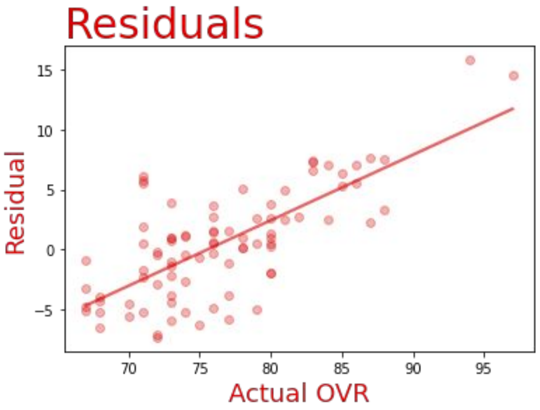
6 Defensive Features

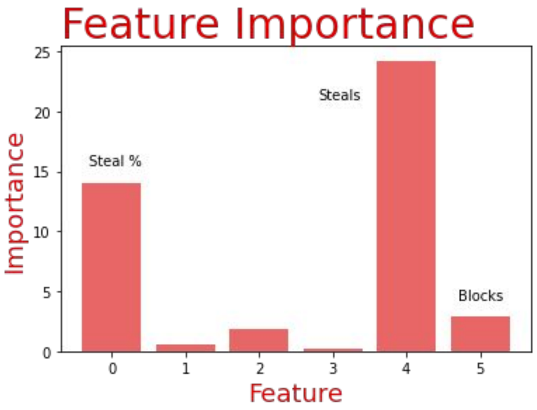
R^2 = **0.47**

Validation error: **4.63**

STL, STL%, BLK







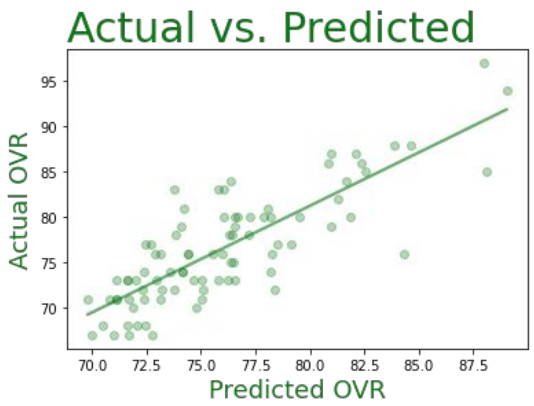
Data Exploration and Modeling

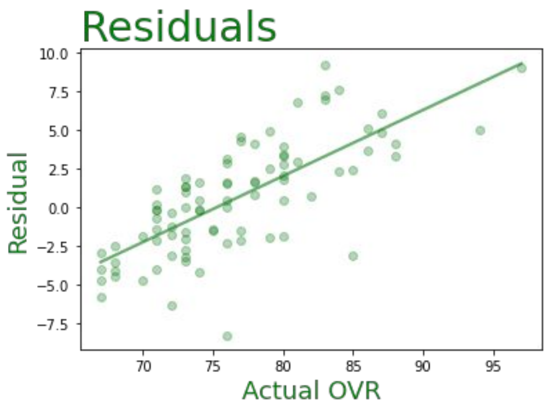
16 Offensive Features

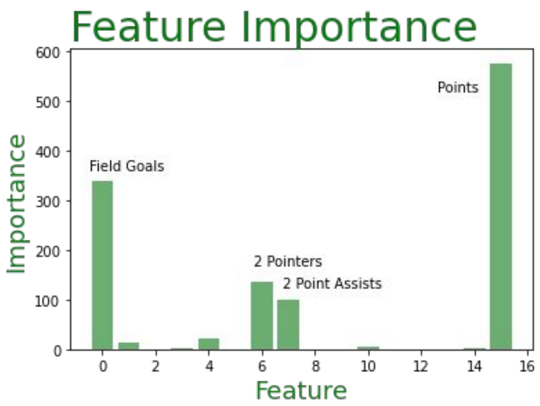
R^2 = **0.70**

Validation error: **3.64**

PTS, FG, 2PT, 2PA







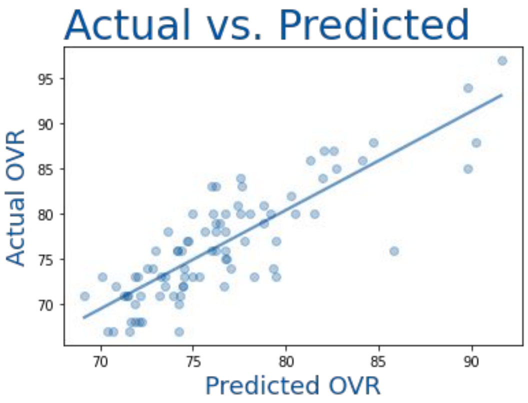
Data Exploration and Modeling

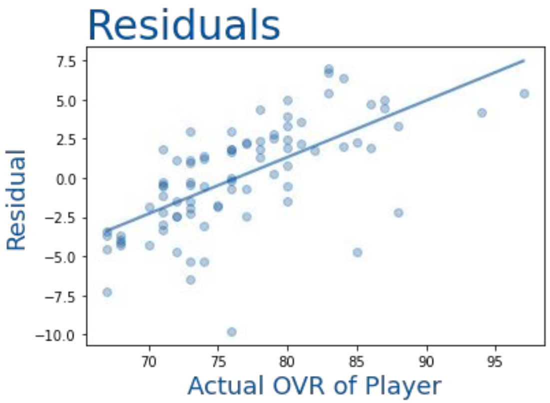
19 Mixed Features

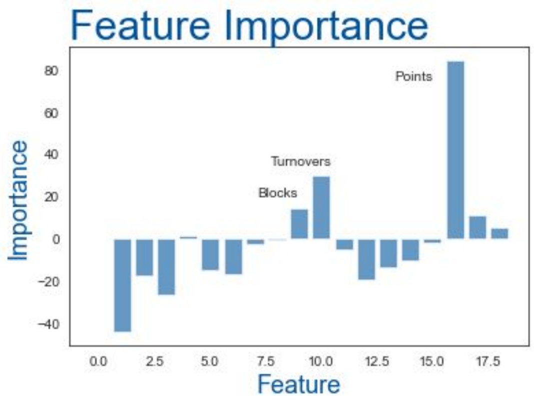
R^2 = **0.74**

Validation error: **3.40**

PTS, TOV, BLK



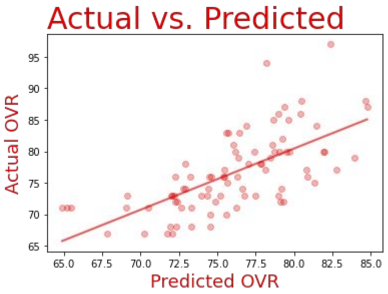


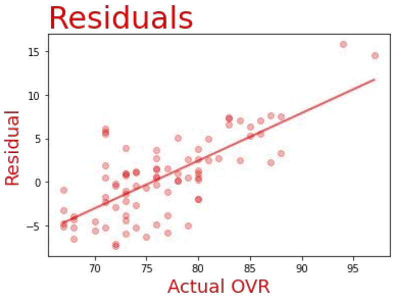


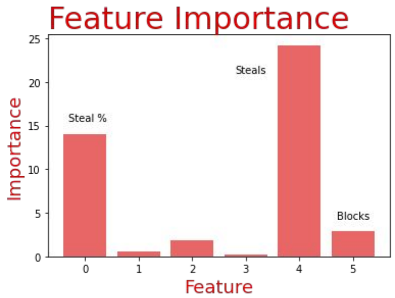
Defense

R^2 = **0.47**

Error = **4.63**



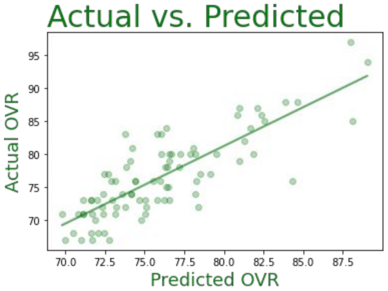


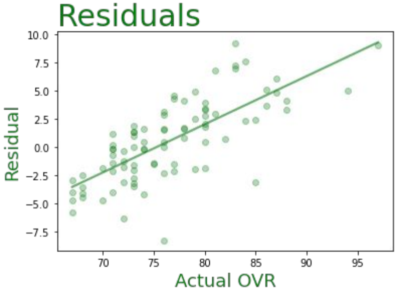


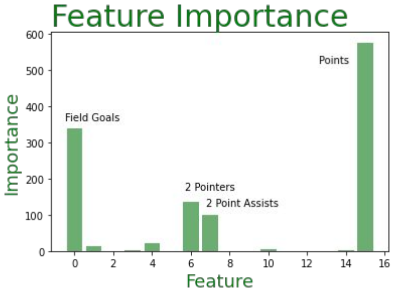
Offense

R^2 = **0.70**

Error = **3.64**



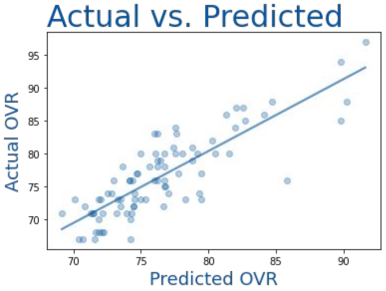


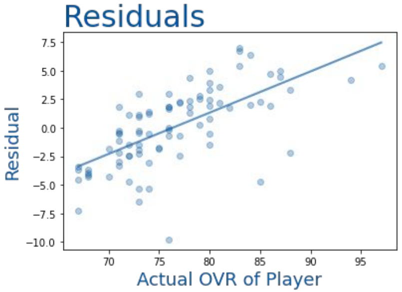


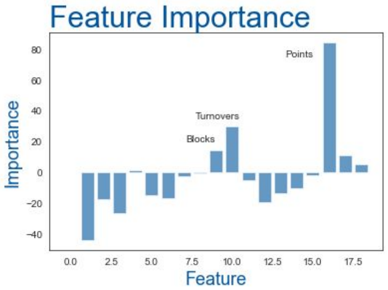
Mixed

R^2 = **0.74**

Error = **3.40**







Analysis of Results

```
#The next step in evaluating and tuning our linear regression models was to perform hypothesis tests on the variables.
from scipy import stats
params = np.append(linear_model.intercept_,linear_model.coef_)
predictions = linear_model.predict(X_train)
new_X = np.append(np.ones((len(X_train),1)), X_train, axis=1)
M_S_E = (sum((Y_train-predictions)**2))/(len(new_X)-len(new_X[0]))
v_b = M_S_E*(np.linalg.inv(np.dot(new_X.T,new_X)).diagonal())
s_b = np.sqrt(v_b)
t_b = params/ s_b
p_val =[2*(1-stats.t.cdf(np.abs(i),(len(new_X)-len(new_X[0])))) for i in t_b]
p_val = np.round(p_val,3)
p_val2=p_val[1:]
```

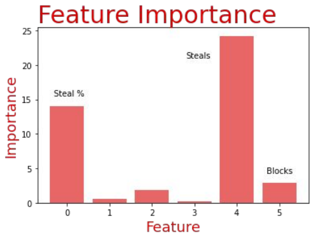
```
featurnames1=['STL%','BLK%','DWS','DBPM','STL','BLK']
```

```
hyptest=pd.DataFrame([featurnames1,p_val2])
hyptest
```

Analysis of Results

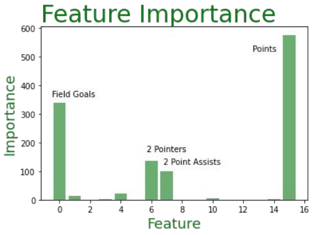
Results of hypothesis testing on defensive linear model:

	0	1	2	3	4	5
0	STL%	BLK%	DWS	DBPM	STL	BLK
1	0.0	0.333	0.004	0.287	0.0	0.021



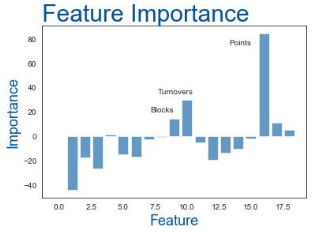
Results of hypothesis testing on offensive linear model:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%	ORB	AST	PTS
1	0.294	0.877	0.808	0.789	0.663	0.089	0.242	0.549	0.314	0.995	0.69	0.673	0.228	0.662	0.0	0.258

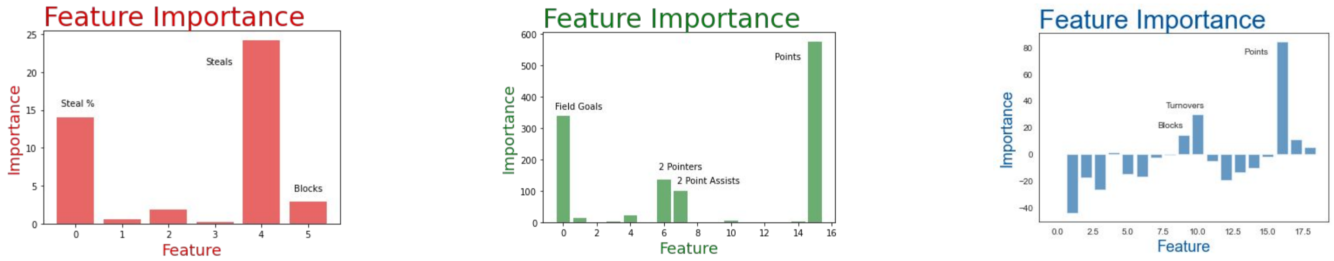


Results of hypothesis testing on rookie linear model:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	G	FG	FGA	3P	3PA	FT	FTA	ORB	STL	BLK	TOV	PF	FG%	3P%	FT%	MP	PTS	TRB	AST
1	0.997	0.942	0.887	0.787	0.985	0.951	0.857	0.366	0.97	0.091	0.426	0.641	0.0	0.0	0.167	0.96	0.959	0.683	0.829



Discussions and Conclusions



Defense	Perm. imp.	Offense	Perm. imp.	Mixed	Perm. imp.
Steals	24.23527	Points	576.25272	Points	83.99036
Steal %	13.98909	Field Goals	340.02493	Turnovers	29.76969
Blocks	2.92047	2 Pointers	137.05598	Blocks	14.30556
DWS	1.82538	2 PA	98.61912	Total Rebounds	10.80018

Discussions and Conclusions

The **mixed set** has the most features, highest R-squared value, and least validation error, so it likely has the best predicting power.

The permutation test for this model and the offensive model show that **POINTS** is by far the statistic that contributes most to a player's 2K Overall Score.

When **defensive** statistics were the only predictors, **STEALS** contribute the most to OVR. However, these stats don't really contribute to the prediction of OVR scores at all when considered with offensive stats.

94 Rookies

Rookies

Career BAA/NBA Stats | Share & Export ▾ | Glossary

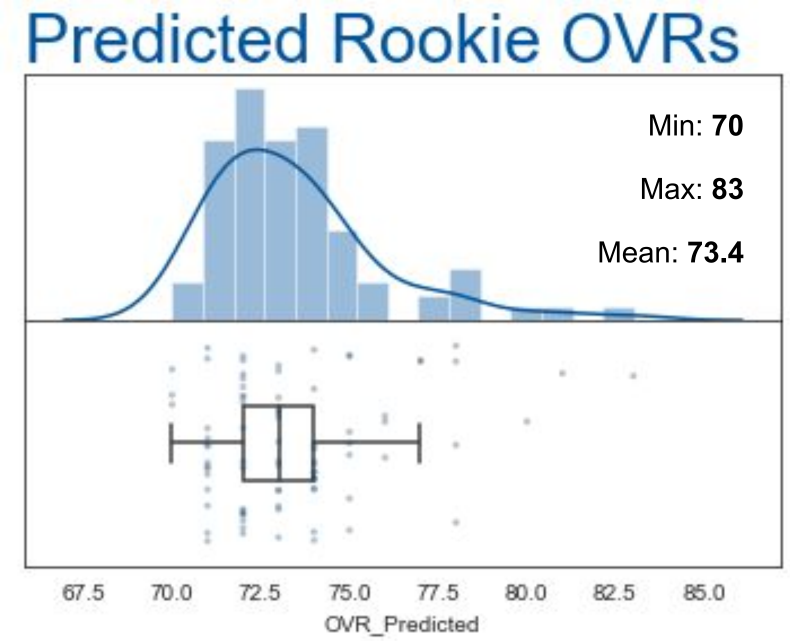
					Totals																Shooting			Per Game			
Rk	Player	Debut	Age	Yrs	G	MP	FG	FGA	3P	3PA	FT	FTA	ORB	TRB	AST	STL	BLK	TOV	PF	PTS	FG%	3P%	FT%	MP	PTS	TRB	AST
1	Precious Achiuwa	Dec 23, '20, MIA @ ORL	21	1	61	737	124	228	0	1	56	110	73	208	29	20	28	43	91	304	.544	.000	.509	12.1	5.0	3.4	0.5
2	Ty-Shon Alexander	Dec 27, '20, PHO @ SAC	22	1	15	47	3	12	2	9	1	2	2	10	6	0	1	3	2	9	.250	.222	.500	3.1	0.6	0.7	0.4
3	Cole Anthony	Dec 23, '20, ORL vs. MIA	20	1	47	1273	219	552	58	172	109	131	38	221	192	30	18	106	98	605	.397	.337	.832	27.1	12.9	4.7	4.1
4	Deni Avdija	Dec 23, '20, WAS @ PHI	20	1	54	1257	130	312	53	168	29	45	22	262	63	32	15	33	140	342	.417	.315	.644	23.3	6.3	4.9	1.2
5	Udoka Azubuike	Dec 23, '20, UTA @ POR	21	1	15	57	4	9	0	0	8	10	4	13	0	1	4	3	9	16	.444		.800	3.8	1.1	0.9	0.0
6	LaMelo Ball	Dec 23, '20, CHO @ CLE	19	1	51	1469	293	672	92	261	125	165	63	302	313	81	18	145	136	803	.436	.352	.758	28.8	15.7	5.9	6.1
7	Desmond Bane	Dec 23, '20, MEM vs. SAS	22	1	68	1519	234	499	117	271	40	49	31	210	118	41	16	59	125	625	.469	.432	.816	22.3	9.2	3.1	1.7
8	Saddiq Bey	Dec 26, '20, DET vs. CLE	21	1	70	1909	279	691	175	460	124	147	43	318	95	52	14	60	110	857	.404	.380	.844	27.3	12.2	4.5	1.4
9	Tyler Bey	Jan 13, '21, DAL @ CHO	22	1	18	71	7	22	1	4	3	5	8	19	3	0	1	3	6	18	.318	.250	.600	3.9	1.0	1.1	0.2
10	Keljin Blevins	Dec 23, '20, POR vs. UTA	25	1	17	75	5	20	2	8	0	0	3	10	4	2	0	5	8	12	.250	.250		4.4	0.7	0.6	0.2
11	Amida Brimah	Apr 25, '21, IND @ ORL	26	1	5	29	5	8	0	0	3	3	2	8	1	0	5	4	4	13	.625		1.000	5.8	2.6	1.6	0.2
12	Armoni Brooks	Apr 9, '21, HOU @ LAC	22	1	20	520	78	192	60	157	7	12	10	68	30	12	5	22	34	223	.406	.382	.583	26.0	11.2	3.4	1.5
13	Elijah Bryant	May 16, '21, MIL @ CHI	25	1	1	32	6	13	1	5	3	3	2	6	3	0	1	4	4	16	.462	.200	1.000	32.0	16.0	6.0	3.0
14	Facundo Campazzo	Dec 23, '20, DEN vs. SAC	29	1	65	1425	120	315	76	216	80	91	22	134	232	79	14	73	132	396	.381	.352	.879	21.9	6.1	2.1	3.6
15	Devin Cannady	Apr 7, '21, ORL vs. WAS	24	1	8	74	11	28	6	16	6	7	0	5	1	5	1	2	9	34	.393	.375	.857	9.3	4.3	0.6	0.1
16	Vernon Carey Jr.	Dec 30, '20, CHO @ DAL	19	1	19	115	18	36	1	7	9	11	6	27	2	1	5	5	13	46	.500	.143	.818	6.1	2.4	1.4	0.1

Rookies

83 predicted values

```
[74.28143308 72.25745467 78.24337848 73.62560901 81.01851321 74.40943221
 75.49197863 71.18271871 75.04059784 83.01421313 73.48671779 71.07344152
 72.21717762 71.01471889 74.08539871 72.5650785  80.16367368 71.38500422
 73.76320982 72.65077901 72.2790663  71.53206582 71.75695789 77.815327
 72.86144235 74.36460335 73.56241389 76.93439369 71.88177198 71.79135728
 71.37780967 71.09528871 73.37028749 69.53224418 72.82411673 71.6812578
 74.00382853 72.26955454 75.68893613 70.43096712 72.72728058 75.36642376
 75.36042228 71.85163978 74.28053193 71.71550442 73.59308006 71.34096805
 70.7077147  71.70144583 70.70519435 73.2810116  74.06281784 74.24891374
 72.69475908 77.59771245 71.48939445 72.10174407 77.16166353 71.0873449
 73.09746268 74.70872382 71.38079546 73.59085252 71.81773964 74.95393719
 71.94280558 71.96302249 70.82006552 72.94791417 75.68515051 75.43142004
 72.81651941 72.52621795 71.18465366 74.14265929 72.63880721 72.4396465
 75.60824628 74.12390327 70.46951617 77.72970357 71.93866188]
```

Discussion and Conclusions



Cole Anthony, predicted 78 overall

Top Rookie Predictions

83

Keljin Blevins

81

Udoka Azubuike

80

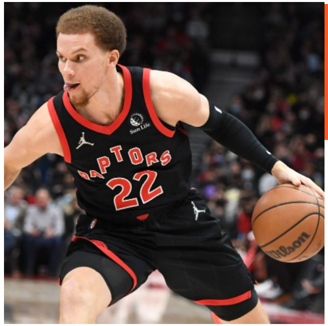
Nate Darling

78

Jaden Mcdaniels

78

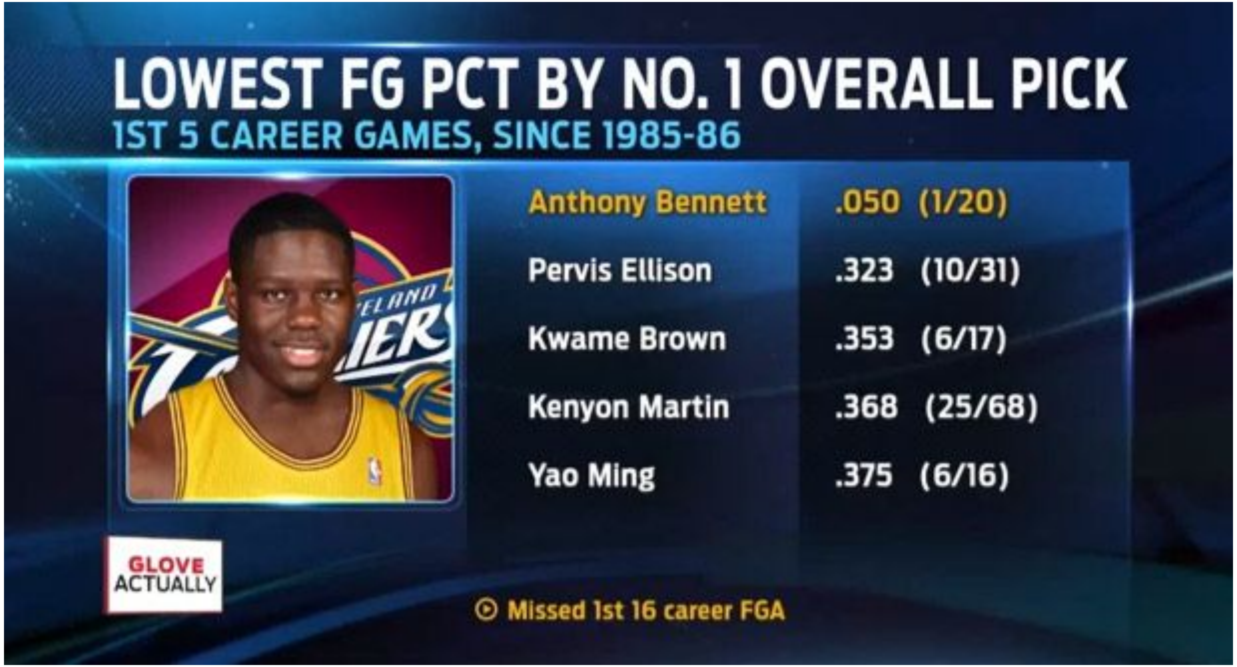
Jae'Sean Tate



78

Malachi Flynn

Limitations



Stats aren't everything.

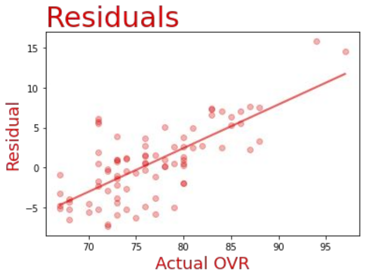
Limitations

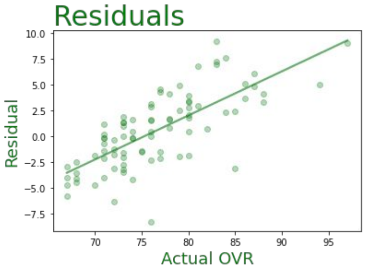
1. A lot of time was spent **cleaning**, so not as much was available for analysis

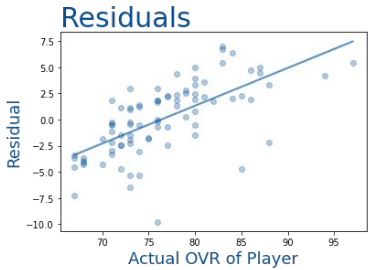
2. We only looked at **one season** of player data because we only wanted to scrape one website for OVR scores.

3. We used linear regression because there was a linear relationship between most of our features and OVR. However, lots of these stats were not normally distributed.

Linear trend in residual plots - Use a different model







Future Work

1. Fit a **different model** to our data set.

2. Using seasons other than 2020-21. How do predictions change?

3. Combining data from **more than one season**.

4. Predicting rookies OVR from years ago and comparing them to their current 2k21 OVRs. How good is our model?

5. Predicting **positions** or some other form of grouping based on per-game stats.

References

1. https://www.walmart.com/ip/NBA-2K21-2K-PlayStation-4/885514502 (picture first slide)
2. https://i.ytimg.com/vi/ogHdj5l4ydI/maxresdefault.jpg (picture second slide)
3. https://2kmtcentral.com/21/players/8312/lebron-james (picture second slide)
4. Brown, George, et al. “Using NCAA Stats to Predict NBA Draft Order.” *Playing Numbers*, 4 Mar. 2020, https://www.playingnumbers.com/2020/03/using-ncaa-stats-to-predict-nba-draft-order/#:~:text=This%20model%20.
5. Liu, Michael. “What Makes a Player Score? NBA PPG Predictions from a College Data Scientist.” *Medium*, Medium, 30 June 2020, https://medium.com/@michaelliu36/what-makes-a-player-score-nba-ppg-predictions-from-a-college-data-scientist-3086f9fbb23.
6. Hwang, JP. “Learn Linear Regression Using Scikit-Learn and NBA Data: Data Science with Sports.” *Medium*, Towards Data Science, 18 Sept. 2020, https://towardsdatascience.com/learn-linear-regression-using-scikit-learn-and-nba-data-data-science-with-sports-9908b0f6a031.
7. BasketballReference.com
8. https://nba2kw.com/list/nba-2k21-all-player-ratings/
9. https://ballislife.com/anthony-bennett-scores-1st-field-goal/
10. https://www.orlandopinstripedpost.com/2022/1/29/22907800/orlando-magic-cole-anthony-nba-slam-dunk-contest
11. https://basketballsocietyonline.com/jaden-mcdaniels-scouting-report
12. https://scarletandgame.com/2021/06/18/ohio-state-basketball-jaesean-tate-makes-rookie-team/
13. https://www.sltrib.com/sports/jazz/2022/03/25/utah-jazz-center-udoka/
14. https://ca.movies.yahoo.com/toronto-raptors-mismanaged-malachi-flynn-031726019.html