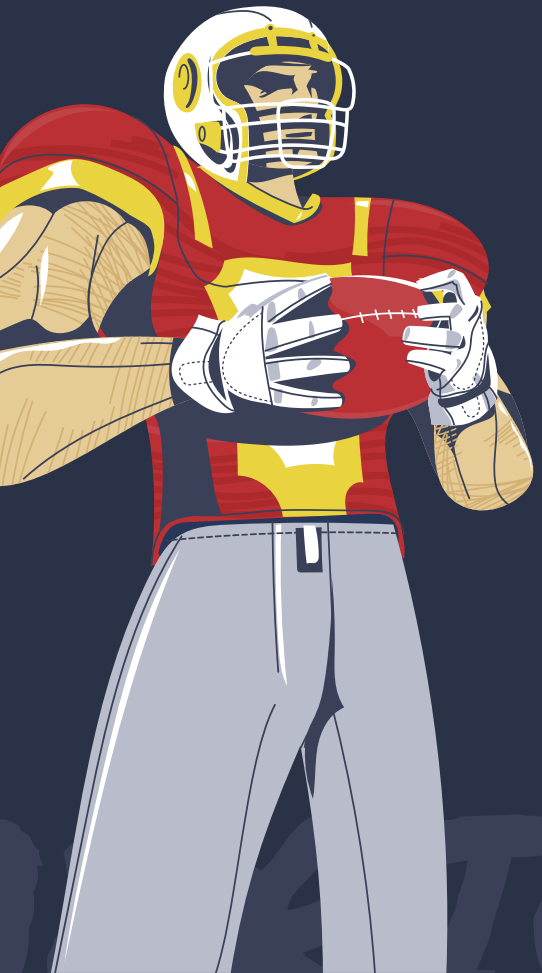
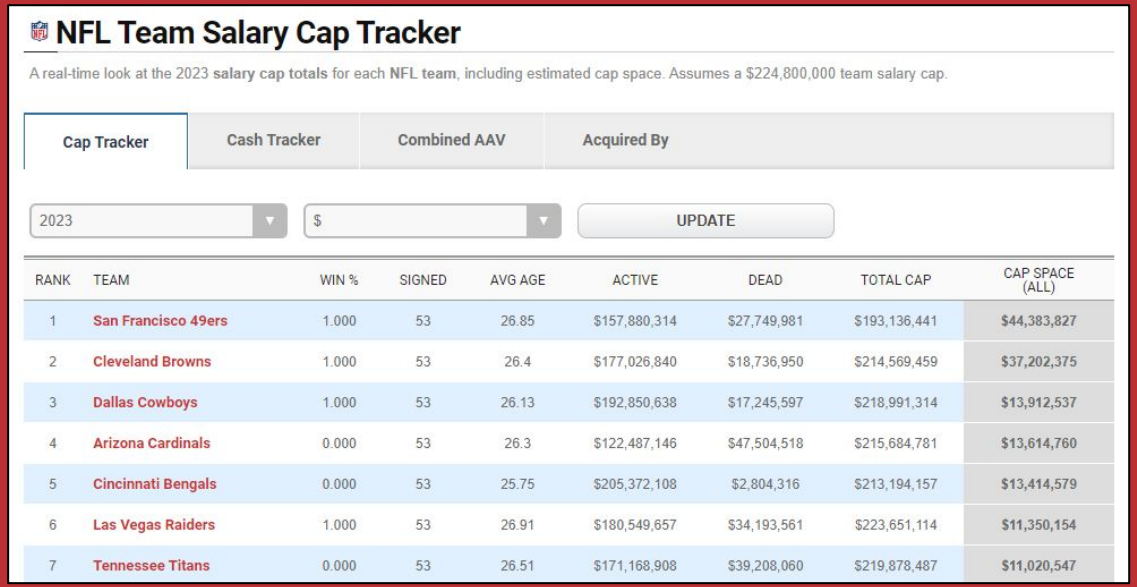


PREDICTING NFL VESTED VETERANS

Billie Kim, Casey Ng, Ethan Panal
Group 28

Problem

- NFL Team General Managers (GMs) manage multi-million dollar budgets
- NFL Players become Vested Veterans after 3 years in the league
- **Solution**: Aid GMs with budget strategy with predictions if a rookie will make it long enough to be a vested veteran
- **Dataset:** Offensive rookie data from 1918-2019



NFL Team Salary Cap Tracker

A real-time look at the 2023 **salary cap totals** for each **NFL team**, including estimated cap space. Assumes a $224,800,000 team salary cap.

Cap Tracker | Cash Tracker | Combined AAV | Acquired By

2023 | $ | UPDATE

RANK	TEAM	WIN %	SIGNED	AVG AGE	ACTIVE	DEAD	TOTAL CAP	CAP SPACE (ALL)
1	San Francisco 49ers	1.000	53	26.85	$157,880,314	$27,749,981	$193,136,441	$44,383,827
2	Cleveland Browns	1.000	53	26.4	$177,026,840	$18,736,950	$214,569,459	$37,202,375
3	Dallas Cowboys	1.000	53	26.13	$192,850,638	$17,245,597	$218,991,314	$13,912,537
4	Arizona Cardinals	0.000	53	26.3	$122,487,146	$47,504,518	$215,684,781	$13,614,760
5	Cincinnati Bengals	0.000	53	25.75	$205,372,108	$2,804,316	$213,194,157	$13,414,579
6	Las Vegas Raiders	1.000	53	26.91	$180,549,657	$34,193,561	$223,651,114	$11,350,154
7	Tennessee Titans	0.000	53	26.51	$171,168,908	$39,208,060	$219,878,487	$11,020,547

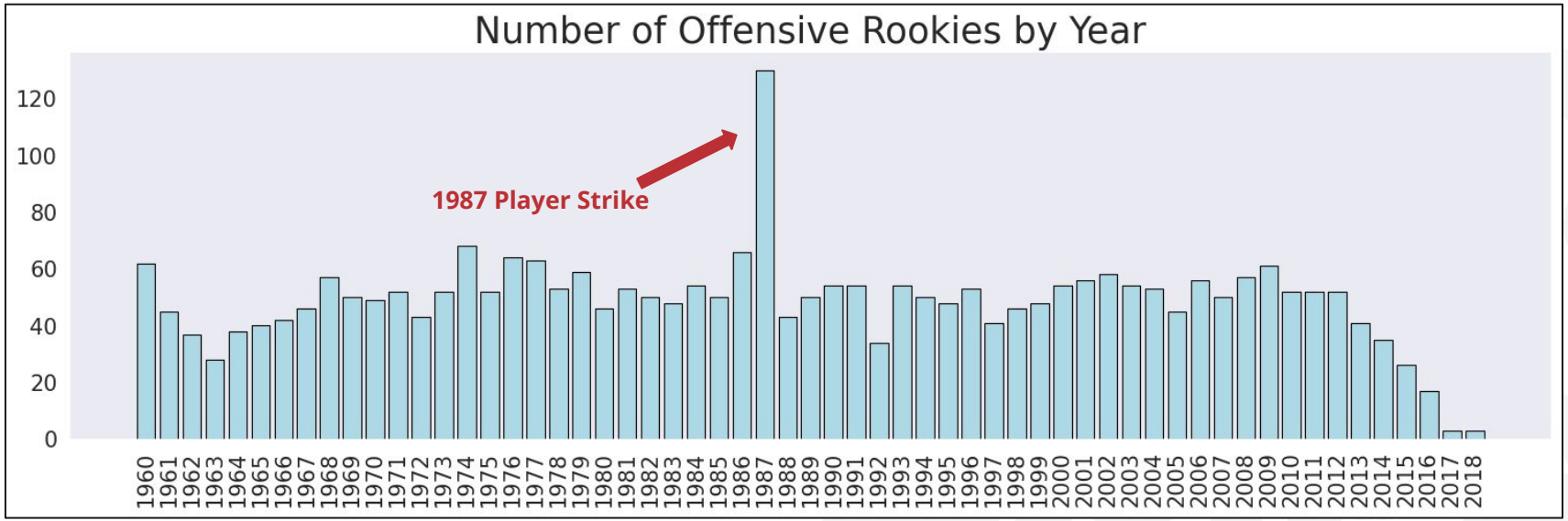
Data Cleaning/Wrangling

- Merged datasets: receiving + rushing + basic info
- Convert data types
- Standardize values
- Impute missing data
- Transformed # of years played to boolean if > 3 years

	Player_Id	Year	Team	Games_Played	Attempts	Yards_Rush	Average_Rush	Long_Rush	TDs_Rush	First_Downs_Rush	...	Yards_Receptions	Average_Receptions	L
0	a-b-brown	1989	NewYorkJets	16	12	63	5.3	17	0	0	...	10	2.5	
1	a-d-whitfield	1965	DallasCowboys	2	1	0	0	0	0	0	...	0	0	
2	a-j-jenkins	2012	SanFrancisco49ers	3	0	0	0	0	0	0	...	0	0	
3	aaron-bailey-2	1994	IndianapolisColts	13	0	0	0	0	0	0	...	30	15	
4	aaron-brooks	1999	GreenBayPackers	0	0	0	0	0	0	0	...	0	0	

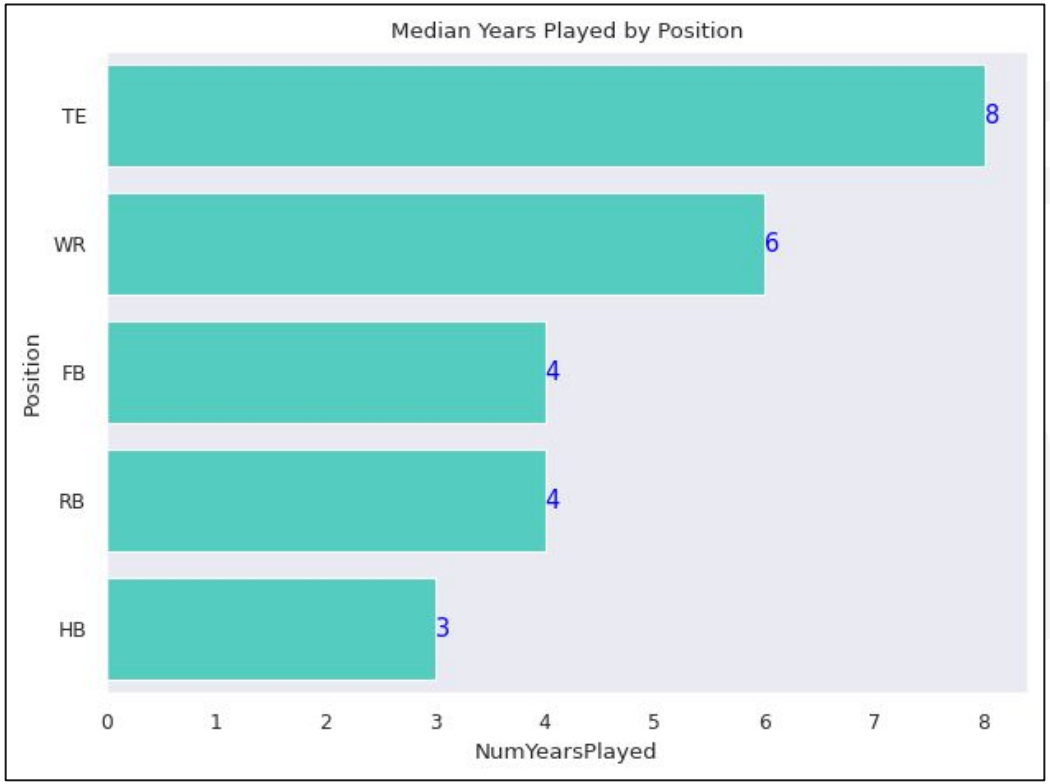
5 rows × 25 columns

EDA



- 3000+ retired offensive player stats from their rookie seasons

EDA

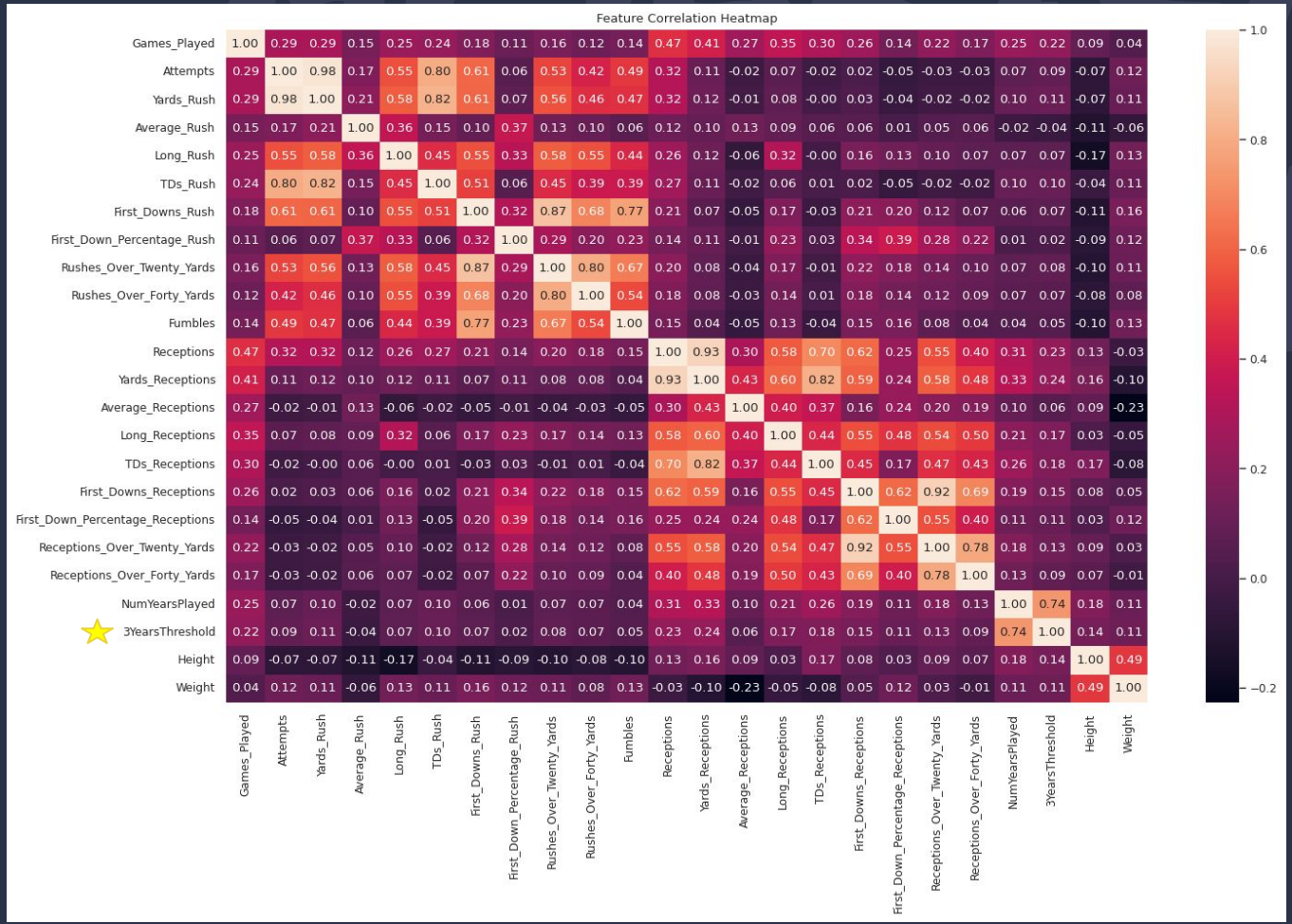


- Offensive players tend to last 3-8 years in league
- Running backs have the shortest careers

Correlation Heatmap

Correlation Patterns:

Rushing Stats
Vs.
Receiving Stats



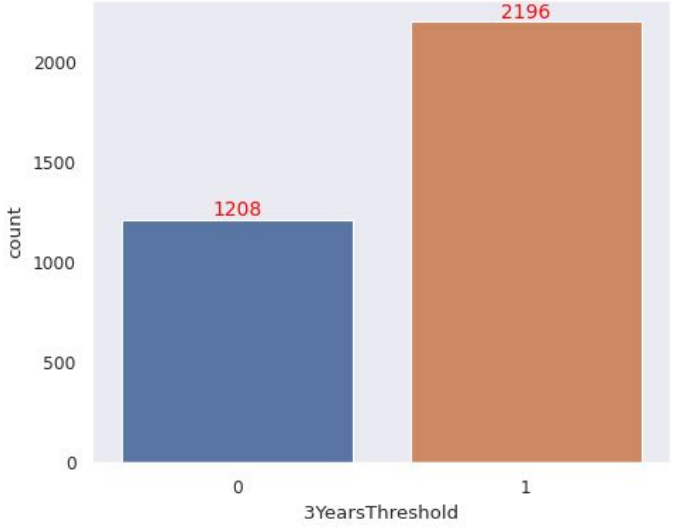
Feature Selection

- Domain Knowledge
- ANOVA F-Tests
- Random Forest Feature Importance

Final Features:
Weight, Games_Played, Yards_Receptions, Yards_Rush, Average_Rush, Attempts, Receptions, Height, Long_Receptions, Average_Receptions, Long_Rush

	ANOVA	Random Forest
Weight	25.765948	0.126059
Games_Played	122.562439	0.108469
Yards_Receptions	164.037910	0.103428
Yards_Rush	29.793872	0.084811
Attempts	19.116489	0.080928
Average_Rush	1.417779	0.079855
Receptions	159.192758	0.077155
Average_Receptions	9.301793	0.076650
Height	44.161096	0.069651
Long_Receptions	79.716845	0.049955
Long_Rush	13.974476	0.045311
TDs_Rush	23.883879	0.026140
TDs_Receptions	89.453975	0.019827
First_Downs_Receptions	54.415906	0.017637
First_Downs_Rush	12.129420	0.015883
Receptions_Over_Twenty_Yards	41.218663	0.007487
Rushes_Over_Twenty_Yards	17.685941	0.006805
Fumbles	7.628763	0.003949

Baseline Accuracy




```
[284] df['3YearsThreshold'].value_counts(normalize=True)

      1    0.645123
      0    0.354877
      Name: 3YearsThreshold, dtype: float64
```

Zero-Rate Classifier Accuracy (Baseline) = 64.51 %

*** Theoretical Baseline for Classification ***

ML Algorithm Comparison

```
LogisticRegression()
Accuracy Score:  0.678015608759227

DecisionTreeClassifier()
Accuracy Score:  0.6372769233479556

RandomForestClassifier()
Accuracy Score:  0.7132718496203513

KNeighborsClassifier()
Accuracy Score:  0.68262424466642

GradientBoostingClassifier()
Accuracy Score:  0.7191418705847119

GaussianNB()
Accuracy Score:  0.5789167239222733

SVC()
Accuracy Score:  0.7073921392456354
```

Method: *Looping thru Mean Cross-Validation Accuracy Scores*

Base Models

TRAINING ACCURACY, TESTING ACCURACY:

1. **Random Forest** (0.9987405541561712, 0.7064579256360078)
2. **Gradient Boosting** (0.781696053736356, 0.7211350293542075)
3. **Support Vector Machine** (0.7246011754827876, 0.7045009784735812)

Hyperparameter Tuning

SVC Best Hyperparameters:

- 'C': 10
- 'gamma': 'scale'
- 'kernel': 'rbf'

Best CV Score: **0.7150**

Accuracy on Test Set: **0.6967**

Baseline Accuracy: 0.7074

```
svc_param_grid = {
    'C': [0.1, 1, 10, 100, 1000],
    'gamma': ['scale', 'auto'],
    'kernel': ['rbf', 'poly', 'sigmoid']
}
```

Hyperparameter Tuning

Gradient Boost Best Hyperparameters:

- subsample: 0.5
- n_estimators: 2000
- min_samples_split: 5
- min_samples_leaf: 8
- max_features: 'log2'
- max_depth: None
- loss: 'log_loss'
- learning_rate: 0.001

Best CV Score: **0.7254**

Accuracy on Test Set: **0.7221**

Baseline Accuracy: 0.7191

```
gb_param_grid = {
    'loss':['log_loss', 'exponential'],
    'learning_rate':[0.001, 0.01, 0.1, 0.2],
    'subsample':[0.5, 0.8, 1.0],
    'max_depth': [20, 40, 60, 80, 100, None],
    'max_features': ['sqrt', 'log2', None],
    'min_samples_leaf': [1, 2, 4, 8],
    'min_samples_split': [2, 5, 10],
    'n_estimators': [100, 400, 1000, 2000]
}
```

Hyperparameter Tuning

Random Forest Best Hyperparameters:

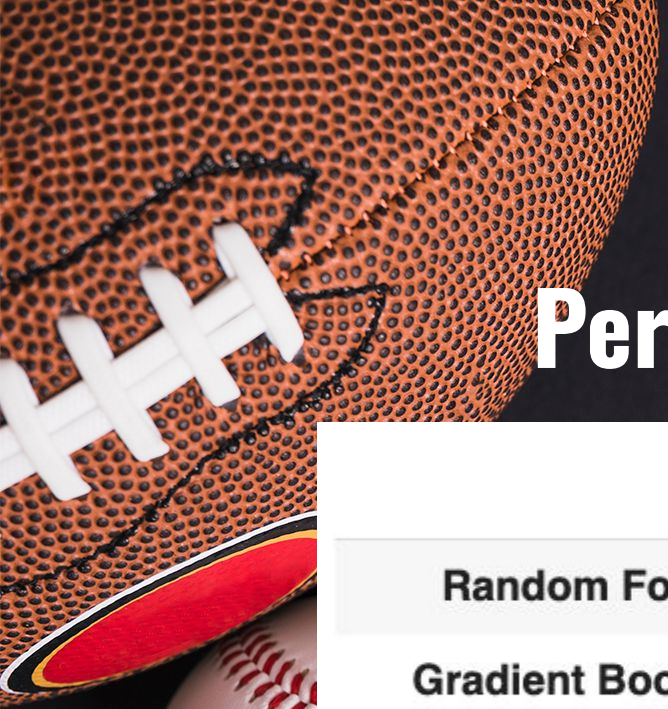
- n_estimators: 400
- min_samples_split: 10
- min_samples_leaf: 8
- max_features: 'sqrt'
- max_depth: 60
- criterion: 'log_loss'
- bootstrap: True

Best CV Score: **0.7313**

Accuracy on Test Set: **0.7290**

Baseline Accuracy: 0.7132

```
rf_param_grid = {
    'criterion': ['gini', 'entropy', 'log_loss'],
    'bootstrap': [True, False],
    'max_depth': [20, 40, 60, 80, 100, None],
    'max_features': ['auto', 'sqrt', 'log2', None],
    'min_samples_leaf': [1, 2, 4, 8],
    'min_samples_split': [2, 5, 10],
    'n_estimators': [100, 200, 400, 800, 1200, 2000]
}
```



Performance Metrics Evaluations

	Accuracy	Recall	Precision	F1 Score
Random Forest	0.728963	0.869231	0.746367	0.803127
Gradient Boosting	0.722114	0.873846	0.737662	0.800000
Support Vector Machine	0.696673	0.836923	0.727273	0.778255

CONCLUSION/NEXT STEPS

SINGLE MODEL?

- LESS EXPENSIVE
- SIMPLICITY
- GENERALLY, LOWER PERFORMANCE
- NEED LESS DATA

SPECIALIZED MODELS?

- MORE EXPENSIVE
- MORE COMPLEX
- GENERALLY, HIGHER PERFORMANCE
- NEED MORE DATA

Final Deliverable - 2020 Rookies

	Player_Id	Prediction
1	a-j-dillon	1
7	adrian-killins-jr	1
28	anthony-mcfarland-2	0
33	antonio-gandy-golden	1
34	antonio-gibson-2	1
35	antonio-williams	1
53	brandon-aiyuk	1
67	cam-akers	1
74	ceedee-lamb	1
78	chase-claypool	1
90	clyde-edwards-helaire	1
93	cole-kmet	1
103	d-andre-swift	1
122	darnell-mooney	1