MLB Salary Prediction

K Nearest Neighbors Regression

Eduardo Vasquez-Villalpando

Research Intent

- Help players and clubs compensate players **fairly**
- **Question: Is KNN a good method for predicting salaries, and are performance metrics good predictors?**

(Stats) Knowledge is Power

Dataset Description and Sources

- Two Datasets, Merged:
  1. Salaries (2000-2016)
  2. Player Performance (2000-2016)
- Package: **pybaseball**
  - Pulls data from baseball-reference.com
- ***Only* for Pitchers**

Salary Data

	yearID	teamID	lgID	playerID	salary
20624	2010	ARI	NL	abreuto01	407000
20625	2010	ARI	NL	boyerbl01	725000
20626	2010	ARI	NL	drewst01	3400000
20627	2010	ARI	NL	gutieju01	411000
20628	2010	ARI	NL	harenda01	8250000

Pitcher Performance Data

	IDfg	Season	Name	Team	Age	W	L	WAR	ERA	G	...	LA	Barrels	Barrel%	maxEV	HardHit	HardHit%	Events	CStr%	CSW%	xERA
42	60	2001	Randy Johnson	ARI	37	21	6	10.40	2.49	35	...	NaN	NaN	NaN	NaN	NaN	NaN	0	NaN	NaN	NaN
64	60	2000	Randy Johnson	ARI	36	19	7	9.60	2.64	35	...	NaN	NaN	NaN	NaN	NaN	NaN	0	NaN	NaN	NaN
56	60	2004	Randy Johnson	ARI	40	16	14	9.60	2.60	35	...	NaN	NaN	NaN	NaN	NaN	NaN	0	0.18	0.33	NaN
1	200	2000	Pedro Martinez	BOS	28	18	6	9.40	1.74	29	...	NaN	NaN	NaN	NaN	NaN	NaN	0	NaN	NaN	NaN
266	73	2002	Curt Schilling	ARI	35	23	7	9.30	3.23	36	...	NaN	NaN	NaN	NaN	NaN	NaN	0	0.17	0.32	NaN

5 rows × 334 columns

Data Input and Output

- Reduced from 334 Input Variables to 9
- Input Variables:
  - Some arcane Baseball Stats (FIP, inLI…)
  - **All input variables** but Age and Games Played (G) are **career averages**
- **Output Variable**: Salary

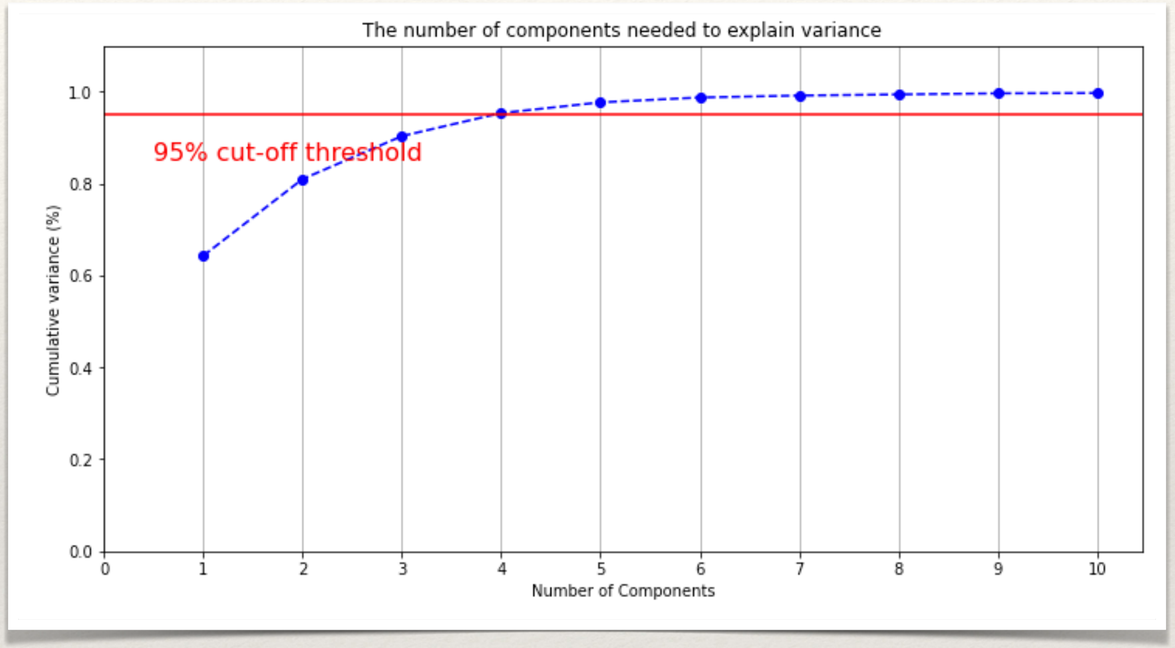
Input Variables

playerID	Age	L	G	SV	SO	FIP	inLI	gmLI	WPA/LI
abbotpa01	33	5.50	63	0.00	109.00	5.06	0.82	0.92	0.57
adamste01	28	8.00	43	0.00	141.00	3.09	1.05	0.99	0.34
alvarhe01	24	10.50	61	0.00	95.00	4.38	0.91	0.87	-0.51
anderbr02	32	10.00	100	0.00	87.00	5.16	0.86	0.84	-0.84
anderbr04	27	10.00	61	0.00	133.00	3.81	0.86	0.88	0.30
...	...	...	...	...	...	...	...	...	...
youngch03	35	7.25	122	0.00	144.00	4.21	0.92	0.88	1.21
zambrca01	28	8.29	221	0.00	175.29	3.90	0.92	0.88	2.15
zambrvi01	29	11.00	65	0.00	122.00	4.68	0.89	0.92	-0.71
zimmejo02	29	8.60	155	0.00	156.80	3.29	0.91	0.87	1.28
zitoba01	34	11.27	373	0.00	153.36	4.34	0.90	0.88	1.10

406 rows × 9 columns

Data Preprocessing

- Reduce from **334 Columns**:
  1. Remove columns with substantial nulls –> **78 Columns**
  2. *Attempted* PCA and…



Reducing to 4 Inputs Seemed Promising

```
1  knn.score(X_test, y_test)
```

```
0.040396341492892285
```

Less-than-stellar R^2

Data Preprocessing

- Reduce from **334 Columns**:
  1. Remove columns with substantial nulls –> **78 Columns**
  2. ~~*Attempted* PCA and…~~ failed miserably
  3. Enter **Sequential Feature Selector**
     - Sk-learn's "forward selection"
     - End result: 9 **Input Variables**

The Final Dataset

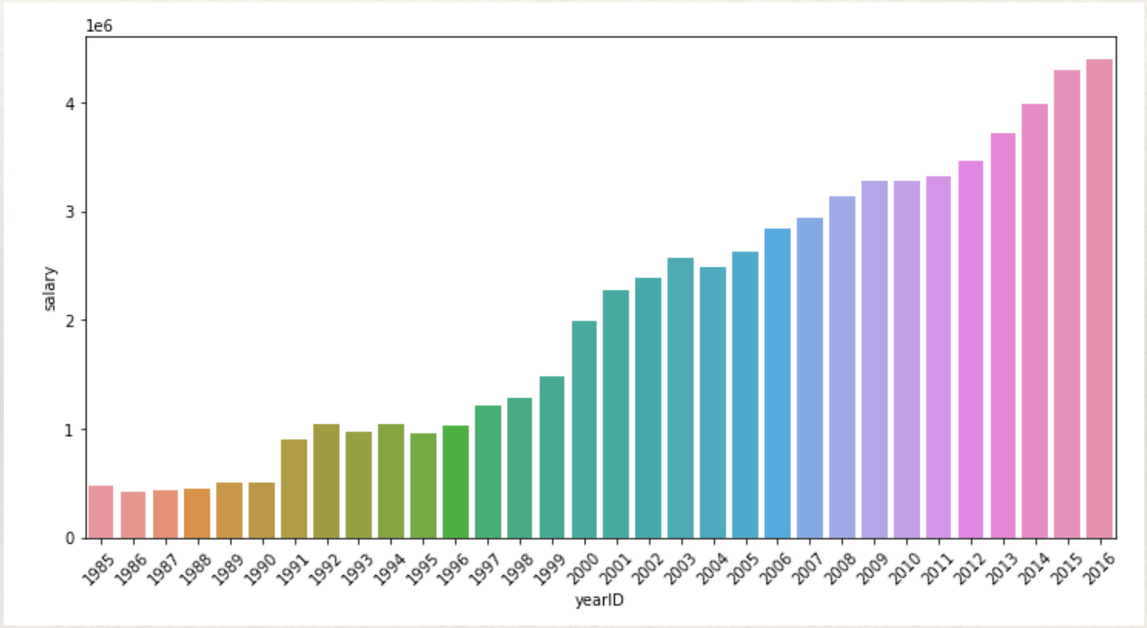
playerID	Age	L	G	SV	SO	FIP	inLI	gmLI	WPA/LI	Salary
abbotpa01	33	5.50	63	0.00	109.00	5.06	0.82	0.92	0.57	992500.00
adamste01	28	8.00	43	0.00	141.00	3.09	1.05	0.99	0.34	2600000.00
alvarhe01	24	10.50	61	0.00	95.00	4.38	0.91	0.87	-0.51	504150.00
anderbr02	32	10.00	100	0.00	87.00	5.16	0.86	0.84	-0.84	2333333.33
anderbr04	27	10.00	61	0.00	133.00	3.81	0.86	0.88	0.30	5200000.00
...	...	...	...	...	...	...	...	...	...	...
youngch03	35	7.25	122	0.00	144.00	4.21	0.92	0.88	1.21	687500.00
zambrca01	28	8.29	221	0.00	175.29	3.90	0.92	0.88	2.15	8314285.71
zambrvi01	29	11.00	65	0.00	122.00	4.68	0.89	0.92	-0.71	1200000.00
zimmejo02	29	8.60	155	0.00	156.80	3.29	0.91	0.87	1.28	6413000.00
zitoba01	34	11.27	373	0.00	153.36	4.34	0.90	0.88	1.10	8957727.27

406 rows × 10 columns

Salary *not* scaled

Exploratory Data Analysis

- The average player made **$3,341,567**
- The top 5% of players made **$9,681,105**
- EDA Takeaway:
  - **Rich Get Richer**
  - League Minimum hasn't increased relatively.
  - Some teams have deeper pockets



Clear relationship between Year and Avg. Salary

Data Splitting

- Straightforward:
  - Split into train, test (70%, 30%)
  - Small number of observations (284, 122), so cross-validation likely needed

Model Training Pipeline

Sequential Feature Selector: **Forward Selection**

KNeighborsRegressor

96 Candidate Models

```
[886]:
pipeline = Pipeline(
    [
    ('selector',SequentialFeatureSelector(n.KNeighborsRegressor(), cv = 3)),
    ('model',n.KNeighborsRegressor())
    ]
)
search = ms.GridSearchCV(
    estimator = pipeline,
    param_grid = {'selector__n_features_to_select':[i for i in range(2,10)],
                  'model__n_neighbors' : [i for i in range(3, 15)]},
    scoring="neg_mean_squared_error",
    cv=3,
    verbose=3
)
search.fit(X_train,y_train)
search.best_params_
search.best_score_

Fitting 3 folds for each of 96 candidates, totalling 288 fits
[CV 1/3] END model__n_neighbors=3, selector__n_features_to_select=2; total time=   0.9s
[CV 2/3] END model__n_neighbors=3, selector__n_features_to_select=2; total time=   1.1s
[CV 3/3] END model__n_neighbors=3, selector__n_features_to_select=2; total time=   0.8s
[CV 1/3] END model__n_neighbors=3, selector__n_features_to_select=3; total time=   0.9s
```

Final Model Parameters: 9 Features, 11 Neighbors

```
search.best_params_

{'model__n_neighbors': 11, 'selector__n_features_to_select': 9}
```

Model Evaluation: Not Terrible!

- *Decent* evaluation metrics
- No signficant overfitting
- *Magel, Hoffman* produced **a better model**: R^2 of .68
- In general, significant outliers were ***undervalued***.

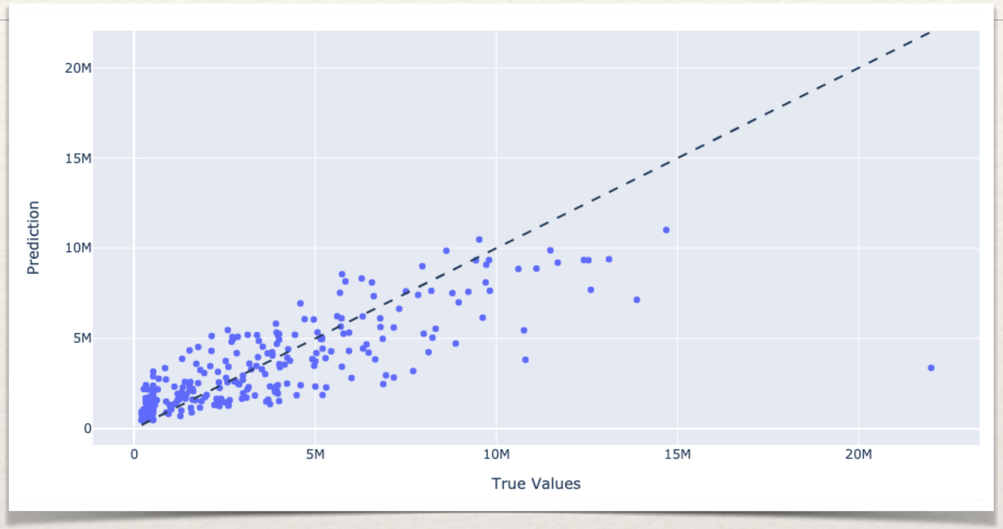
Final Model's Metrics

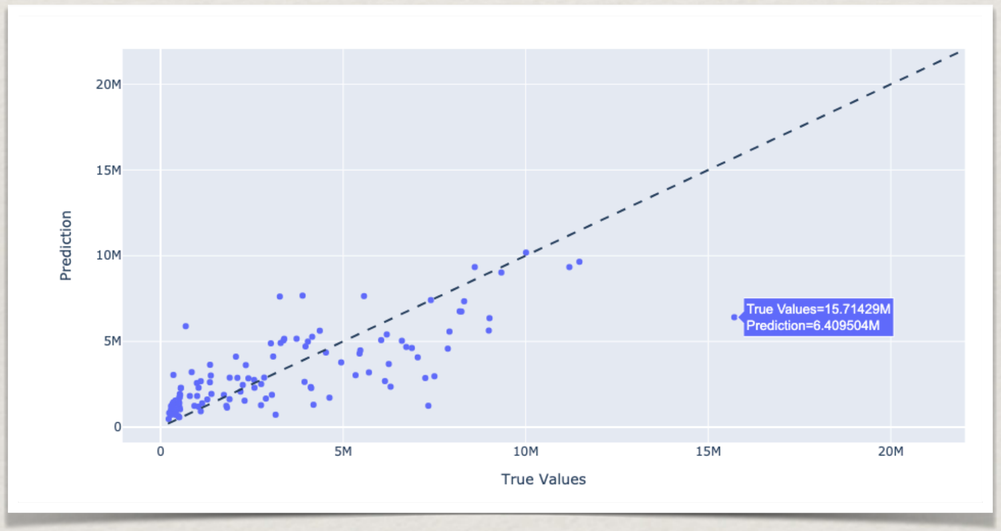
	Training	Test
R^2	0.64	0.6
RMSE	1.98262e+06	1.96281e+06

Model Evaluation: Not Terrible!

- *Decent* evaluation metrics
- No signficant overfitting
- *Magel, Hoffman* produced **a better model**: R^2 of .68
- In general, significant outliers were ***undervalued***.

Good Fit on Training Set





Decent Fit on Test Set

Conclusions

- Feature selection favored reliability in leverage.
- More informative input variables necessary for outliers
  - **Popularity**: Jersey Sales? Accolades?
  - **Contract Year Performance:** Hochberg (2011) found contract year performance to be overweighted
- **Teams Overpay or Underpay Veterans**
  - Players paid significantly more and less than predicted were above average in age.

playerID	Age	L	G	SV	SO	FIP	inLI	gmLI	WPA/LI	Salary
abbotpa01	33	5.50	63	0.00	109.00	5.06	0.82	0.92	0.57	992500.00

Saves + Last 3 Variables are Leverage Stats

```
rf = sklearn.ensemble.RandomForestRegressor()
rf_param_grid = [{'n_estimators' : [3,10,30], 'max_features' : [2,4,6,8]},
                 {'bootstrap' : [False], 'n_estimators' : [3,10], 'max_features' : [2,3,4]}]

grid_search = ms.GridSearchCV(rf, rf_param_grid, cv = 8)
grid_search.fit(X_train_2, y_train)
print("Best Parameters for a CV Random Forests Model:", grid_search.best_params_)
grid_search.best_estimator_.fit(X_train_2, y_train)
print("R^2 for Random Forests", round(grid_search.best_estimator_.score(X_test_2, y_test), 2))

```

```
Best Parameters for a CV Random Forests Model: {'max_features': 2, 'n_estimators': 30}
R^2 for Random Forests 0.58
```

Better Than Random Forests…Barely