Homework 5 - Grant Jackson

October 2, 2024

0.0.1 HW5

- Use the dataset, "Hitters.csv", posted on BB to explain/predict a baseball player's salary using a subset of covariates in the dataset .
- In order to select the best set of covariates, do the following:
  - LASSO Estimations with CV, AIC and BIC
  - Run regular OLS Regression with non-zero covariates indicated by each CV, AIC, BIC (e.g. Regular OLS Regression excluding covariates dropped (are 0) by LASSO CV, Regular OLS Regression excluding covariates dropped (are 0) by LASSO AIC,.....)
  - Produce tables or figures or both to summarize your results (use summarycol)
- For this exercise, you need to take care of missing values and generate dummies for some variables

[1]:
```
# Importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV, LassoLarsIC
import statsmodels.api as sm
from statsmodels.iolib.summary2 import summary_col

# Load the data
import os
os.chdir('C:\\Users\gmoor\Documents\Economic Analytics 1\Data')
hitters_data = pd.read_csv('Hitters.csv')

hitters_data.head()
```

[1]:

	Unnamed: 0	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	\
0	-Andy Allanson	293	66	1	30	29	14	1	293	
1	-Alan Ashby	315	81	7	24	38	39	14	3449	
2	-Alvin Davis	479	130	18	66	72	76	3	1624	
3	-Andre Dawson	496	141	20	65	78	37	11	5628	
4	-Andres Galarraga	321	87	10	39	42	30	2	396	

	CHits	…	CRuns	CRBI	CWalks	League	Division	PutOuts	Assists	Errors	\
0	66	…	30	29	14	A	E	446	33	20	

```
1    835  …    321   414     375       N       W     632       43      10
2    457  …    224   266     263       A       W     880       82      14
3   1575  …    828   838     354       N       E     200       11       3
4    101  …     48    46      33       N       E     805       40       4

   Salary  NewLeague
0     NaN          A
1   475.0          N
2   480.0          A
3   500.0          N
4    91.5          N

[5 rows x 21 columns]
```

[2]:
```
# Display information about the dataset
print("Dataset Info:")
print(hitters_data.info())
print("\nDataset Description:")
print(hitters_data.describe())
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 322 entries, 0 to 321
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  322 non-null    object
 1   AtBat       322 non-null    int64
 2   Hits        322 non-null    int64
 3   HmRun       322 non-null    int64
 4   Runs        322 non-null    int64
 5   RBI         322 non-null    int64
 6   Walks       322 non-null    int64
 7   Years       322 non-null    int64
 8   CAtBat      322 non-null    int64
 9   CHits       322 non-null    int64
 10  CHmRun      322 non-null    int64
 11  CRuns       322 non-null    int64
 12  CRBI        322 non-null    int64
 13  CWalks      322 non-null    int64
 14  League      322 non-null    object
 15  Division    322 non-null    object
 16  PutOuts     322 non-null    int64
 17  Assists     322 non-null    int64
 18  Errors      322 non-null    int64
 19  Salary      263 non-null    float64
 20  NewLeague   322 non-null    object
dtypes: float64(1), int64(16), object(4)
```

```
memory usage: 53.0+ KB
None

Dataset Description:
            AtBat        Hits       HmRun        Runs         RBI       Walks  \
count  322.000000  322.000000  322.000000  322.000000  322.000000  322.000000
mean   380.928571  101.024845   10.770186   50.909938   48.027950   38.742236
std    153.404981   46.454741    8.709037   26.024095   26.166895   21.639327
min     16.000000    1.000000    0.000000    0.000000    0.000000    0.000000
25%    255.250000   64.000000    4.000000   30.250000   28.000000   22.000000
50%    379.500000   96.000000    8.000000   48.000000   44.000000   35.000000
75%    512.000000  137.000000   16.000000   69.000000   64.750000   53.000000
max    687.000000  238.000000   40.000000  130.000000  121.000000  105.000000

            Years       CAtBat        CHits      CHmRun        CRuns  \
count  322.000000    322.00000   322.000000  322.000000   322.000000
mean     7.444099   2648.68323   717.571429   69.490683   358.795031
std      4.926087   2324.20587   654.472627   86.266061   334.105886
min      1.000000     19.00000     4.000000    0.000000     1.000000
25%      4.000000    816.75000   209.000000   14.000000   100.250000
50%      6.000000   1928.00000   508.000000   37.500000   247.000000
75%     11.000000   3924.25000  1059.250000   90.000000   526.250000
max     24.000000  14053.00000  4256.000000  548.000000  2165.000000

             CRBI       CWalks      PutOuts     Assists      Errors  \
count  322.000000   322.000000   322.000000  322.000000  322.000000
mean   330.118012   260.239130   288.937888  106.913043    8.040373
std    333.219617   267.058085   280.704614  136.854876    6.368359
min      0.000000     0.000000     0.000000    0.000000    0.000000
25%     88.750000    67.250000   109.250000    7.000000    3.000000
50%    220.500000   170.500000   212.000000   39.500000    6.000000
75%    426.250000   339.250000   325.000000  166.000000   11.000000
max   1659.000000  1566.000000  1378.000000  492.000000   32.000000

            Salary
count   263.000000
mean    535.925882
std     451.118681
min      67.500000
25%     190.000000
50%     425.000000
75%     750.000000
max    2460.000000
```

[3]:
```
# Handle missing values by dropping rows with missing 'Salary' values
hitters_data_clean = hitters_data.dropna(subset=['Salary'])
```

```
# Generate dummy variables for the categorical columns 'League', 'Division',␣
 ↪and 'NewLeague'
hitters_data_clean = pd.get_dummies(hitters_data_clean, columns=['League',␣
 ↪'Division', 'NewLeague'], drop_first=True)

hitters_data_clean.head()
```

[3]:

	Unnamed: 0	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	…	CRuns	CRBI	CWalks	PutOuts	Assists	Errors	Salary	League_N	Division_W	NewLeague_N
1	-Alan Ashby	315	81	7	24	38	39	14	3449	…	321	414	375	632	43	10	475.0	True	True	True
2	-Alvin Davis	479	130	18	66	72	76	3	1624	…	224	266	263	880	82	14	480.0	False	True	False
3	-Andre Dawson	496	141	20	65	78	37	11	5628	…	828	838	354	200	11	3	500.0	True	False	True
4	-Andres Galarraga	321	87	10	39	42	30	2	396	…	48	46	33	805	40	4	91.5	True	False	True
5	-Alfredo Griffin	594	169	4	74	51	35	11	4408	…	501	336	194	282	421	25	750.0	False	True	False

[5 rows x 21 columns]

[4]:
```
# Prepare the data for LASSO: define X and y
X = hitters_data_clean.drop(columns=['Unnamed: 0', 'Salary'])
y = hitters_data_clean['Salary']

# Convert all columns to numeric, replacing any non-numeric values with NaN
for col in X.columns:
    X[col] = pd.to_numeric(X[col], errors='coerce')

# Drop any rows with NaN values after conversion
X = X.dropna()
y = y[X.index]

print("\nShape of X after cleaning:", X.shape)
print("Shape of y after cleaning:", y.shape)

# Check for any remaining non-numeric data
print("\nData types in X:")
```

```
print(X.dtypes)

# Ensure all data is float64
X = X.astype(float)
y = y.astype(float)
```

```
Shape of X after cleaning: (263, 19)
Shape of y after cleaning: (263,)

Data types in X:
AtBat          int64
Hits           int64
HmRun          int64
Runs           int64
RBI            int64
Walks          int64
Years          int64
CAtBat         int64
CHits          int64
CHmRun         int64
CRuns          int64
CRBI           int64
CWalks         int64
PutOuts        int64
Assists        int64
Errors         int64
League_N        bool
Division_W      bool
NewLeague_N     bool
dtype: object
```

[5]:
```
# Standardize the predictors
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

[6]:
```
# Perform LASSO with cross-validation
lasso_cv = LassoCV(cv=5, random_state=0).fit(X_scaled, y)
lasso_cv_coefs = lasso_cv.coef_

lasso_cv_coefs
```

[6]:
```
array([-226.83018915,  254.82705761,    0.        ,   -0.        ,
          0.        ,  102.1450165 ,  -44.5942132 ,   -0.        ,
          0.        ,   43.36306176,  218.02949977,  123.42065271,
       -138.61749903,   76.10383464,   24.74415993,  -13.25243902,
         16.04102919,  -59.54521229,   -0.        ])
```

[7]:
```
# Perform LASSO using AIC and BIC
lasso_aic = LassoLarsIC(criterion='aic').fit(X_scaled, y)
lasso_bic = LassoLarsIC(criterion='bic').fit(X_scaled, y)

lasso_aic_coefs = lasso_aic.coef_
lasso_bic_coefs = lasso_bic.coef_

print(lasso_aic_coefs)
print(lasso_bic_coefs)
```

```
[-243.86820725  266.40127753    0.            0.            0.
  106.60985958  -47.53200842    0.            0.           47.51250671
  230.64749689  121.58874342 -151.07487186   76.98509399   27.89478629
  -14.45413345   16.40289964  -59.57562111    0.         ]
[  0.          83.79144232   0.            0.            0.
  47.96218008   0.            0.            0.            0.
  67.30564773 133.75587469   0.           61.11779896   0.
   0.           0.         -51.06531782   0.         ]
```

[8]:
```
# Function to run OLS regression and display the summary
def run_ols(X, y):
    X = sm.add_constant(X)  # Add a constant
    ols_model = sm.OLS(y, X).fit()
    return ols_model.summary()

# Get selected columns for each method
cv_selected_columns = X.columns[np.abs(lasso_cv_coefs) > 1e-5]
aic_selected_columns = X.columns[np.abs(lasso_aic_coefs) > 1e-5]
bic_selected_columns = X.columns[np.abs(lasso_bic_coefs) > 1e-5]

# Run OLS regression for each set of selected covariates
print("\nRunning OLS regressions...")
ols_cv_results = run_ols(X[cv_selected_columns], y)
ols_aic_results = run_ols(X[aic_selected_columns], y)
ols_bic_results = run_ols(X[bic_selected_columns], y)

# Print the results
print("\nCV Selected Variables:")
print(cv_selected_columns.tolist())
print("\nAIC Selected Variables:")
print(aic_selected_columns.tolist())
print("\nBIC Selected Variables:")
print(bic_selected_columns.tolist())

print("\nOLS Results with CV-selected variables:")
print(ols_cv_results)
print("\nOLS Results with AIC-selected variables:")
print(ols_aic_results)
```

```
print("\nOLS Results with BIC-selected variables:")
print(ols_bic_results)
```

```
Running OLS regressions…

CV Selected Variables:
['AtBat', 'Hits', 'Walks', 'Years', 'CHmRun', 'CRuns', 'CRBI', 'CWalks',
'PutOuts', 'Assists', 'Errors', 'League_N', 'Division_W']

AIC Selected Variables:
['AtBat', 'Hits', 'Walks', 'Years', 'CHmRun', 'CRuns', 'CRBI', 'CWalks',
'PutOuts', 'Assists', 'Errors', 'League_N', 'Division_W']

BIC Selected Variables:
['Hits', 'Walks', 'CRuns', 'CRBI', 'PutOuts', 'Division_W']

OLS Results with CV-selected variables:
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.540
Model:                            OLS   Adj. R-squared:                  0.516
Method:                 Least Squares   F-statistic:                     22.45
Date:                Wed, 02 Oct 2024   Prob (F-statistic):           5.30e-35
Time:                        23:23:41   Log-Likelihood:                -1878.1
No. Observations:                 263   AIC:                             3784.
Df Residuals:                     249   BIC:                             3834.
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        187.5595     88.572      2.118      0.035      13.114     362.005
AtBat         -2.3080      0.562     -4.104      0.000      -3.416      -1.200
Hits           7.3460      1.718      4.277      0.000       3.963      10.729
Walks          6.0861      1.570      3.876      0.000       2.994       9.178
Years        -13.6050     10.383     -1.310      0.191     -34.055       6.845
CHmRun         0.8363      0.847      0.987      0.324      -0.832       2.505
CRuns          0.9092      0.277      3.287      0.001       0.364       1.454
CRBI           0.3573      0.363      0.986      0.325      -0.357       1.071
CWalks        -0.8392      0.272     -3.084      0.002      -1.375      -0.303
PutOuts        0.2930      0.076      3.839      0.000       0.143       0.443
Assists        0.3148      0.205      1.539      0.125      -0.088       0.718
Errors        -3.2322      4.294     -0.753      0.452     -11.690       5.226
League_N      36.6846     40.735      0.901      0.369     -43.544     116.913
Division_W  -119.6740     39.325     -3.043      0.003    -197.126     -42.222
==============================================================================
Omnibus:                       82.868   Durbin-Watson:                   1.978
```

Prob(Omnibus):	0.000	Jarque-Bera (JB):	427.838
Skew:	1.162	Prob(JB):	1.25e-93
Kurtosis:	8.800	Cond. No.	4.18e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.18e+03. This might indicate that there are strong multicollinearity or other numerical problems.

OLS Results with AIC-selected variables:

OLS Regression Results

Dep. Variable:	Salary	R-squared:	0.540
Model:	OLS	Adj. R-squared:	0.516
Method:	Least Squares	F-statistic:	22.45
Date:	Wed, 02 Oct 2024	Prob (F-statistic):	5.30e-35
Time:	23:23:41	Log-Likelihood:	-1878.1
No. Observations:	263	AIC:	3784.
Df Residuals:	249	BIC:	3834.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	187.5595	88.572	2.118	0.035	13.114	362.005
AtBat	-2.3080	0.562	-4.104	0.000	-3.416	-1.200
Hits	7.3460	1.718	4.277	0.000	3.963	10.729
Walks	6.0861	1.570	3.876	0.000	2.994	9.178
Years	-13.6050	10.383	-1.310	0.191	-34.055	6.845
CHmRun	0.8363	0.847	0.987	0.324	-0.832	2.505
CRuns	0.9092	0.277	3.287	0.001	0.364	1.454
CRBI	0.3573	0.363	0.986	0.325	-0.357	1.071
CWalks	-0.8392	0.272	-3.084	0.002	-1.375	-0.303
PutOuts	0.2930	0.076	3.839	0.000	0.143	0.443
Assists	0.3148	0.205	1.539	0.125	-0.088	0.718
Errors	-3.2322	4.294	-0.753	0.452	-11.690	5.226
League_N	36.6846	40.735	0.901	0.369	-43.544	116.913
Division_W	-119.6740	39.325	-3.043	0.003	-197.126	-42.222

Omnibus:	82.868	Durbin-Watson:	1.978
Prob(Omnibus):	0.000	Jarque-Bera (JB):	427.838
Skew:	1.162	Prob(JB):	1.25e-93
Kurtosis:	8.800	Cond. No.	4.18e+03

Notes:

```
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.18e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

OLS Results with BIC-selected variables:
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.487
Model:                            OLS   Adj. R-squared:                  0.475
Method:                 Least Squares   F-statistic:                     40.55
Date:                Wed, 02 Oct 2024   Prob (F-statistic):           1.49e-34
Time:                        23:23:41   Log-Likelihood:                -1892.2
No. Observations:                 263   AIC:                             3798.
Df Residuals:                     256   BIC:                             3823.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -13.4956     60.148     -0.224      0.823    -131.944     104.953
Hits           2.0083      0.564      3.563      0.000       0.898       3.118
Walks          2.4255      1.198      2.025      0.044       0.067       4.784
CRuns          0.2193      0.191      1.145      0.253      -0.158       0.596
CRBI           0.4398      0.194      2.263      0.024       0.057       0.822
PutOuts        0.2645      0.077      3.431      0.001       0.113       0.416
Division_W  -134.0191     40.573     -3.303      0.001    -213.919     -54.119
==============================================================================
Omnibus:                      109.093   Durbin-Watson:                   1.975
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              719.878
Skew:                           1.512   Prob(JB):                    4.79e-157
Kurtosis:                      10.520   Cond. No.                     2.29e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.29e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[10]:
```python
# Create a summary table using summary_col
models = [ols_cv_model, ols_aic_model, ols_bic_model]
model_names = ['CV', 'AIC', 'BIC']
info_dict = {
    'R-squared': lambda x: f"{x.rsquared:.3f}",
    'Adj. R-squared': lambda x: f"{x.rsquared_adj:.3f}",
    'No. observations': lambda x: f"{int(x.nobs)}",
```

```
    'F-statistic': lambda x: f"{x.fvalue:.3f}",
    'Prob (F-statistic)': lambda x: f"{x.f_pvalue:.3f}",
}

summary_table = summary_col(models,
                            model_names=model_names,
                            stars=True,
                            float_format='%0.4f',
                            info_dict=info_dict)

# Print the summary table
print("\nSummary Table of OLS Regressions:")
print(summary_table)
```

```
Summary Table of OLS Regressions:

============================================================
                         CV          AIC          BIC
------------------------------------------------------------
Assists             0.3148       0.3148
                    (0.2046)     (0.2046)
AtBat               -2.3080***   -2.3080***
                    (0.5624)     (0.5624)
CHmRun              0.8363       0.8363
                    (0.8471)     (0.8471)
CRBI                0.3573       0.3573       0.4398**
                    (0.3625)     (0.3625)     (0.1943)
CRuns               0.9092***    0.9092***    0.2193
                    (0.2766)     (0.2766)     (0.1915)
CWalks              -0.8392***   -0.8392***
                    (0.2721)     (0.2721)
Division_W          -119.6740*** -119.6740*** -134.0191***
                    (39.3248)    (39.3248)    (40.5732)
Errors              -3.2322      -3.2322
                    (4.2944)     (4.2944)
Hits                7.3460***    7.3460***    2.0083***
                    (1.7176)     (1.7176)     (0.5636)
League_N            36.6846      36.6846
                    (40.7347)    (40.7347)
PutOuts             0.2930***    0.2930***    0.2645***
                    (0.0763)     (0.0763)     (0.0771)
R-squared           0.5396       0.5396       0.4873
R-squared Adj.      0.5156       0.5156       0.4752
Walks               6.0861***    6.0861***    2.4255**
                    (1.5701)     (1.5701)     (1.1975)
Years               -13.6050     -13.6050
                    (10.3833)    (10.3833)
```

const	187.5595**	187.5595**	-13.4956
	(88.5718)	(88.5718)	(60.1484)
R-squared	0.540	0.540	0.487
Adj. R-squared	0.516	0.516	0.475
No. observations	263	263	263
F-statistic	22.453	22.453	40.545
Prob (F-statistic)	0.000	0.000	0.000

Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01