JQA Logistic Regression

June 18, 2022

[176]:

```
#Importing the Libraries

import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
```

[177]:

```
#Importing the Dataset
df = pd.read_excel('/Users/AmrinSinghDhillon/Desktop/Book2.xlsx')
```

[178]:

```
df
```

[178]:

```
       is_in_play pitch  three_plus batter_stance pitcher_throws  strikes  \
0               1    SL         0.0             R              L      0.0
1               1    CU         0.0             R              R      1.0
2               1    FA         0.0             R              L      2.0
3               1    SI         0.0             R              L      1.0
4               1    SL         0.0             L              R      2.0
…             …     …           …           …             …       …
99995           1    CH         0.0             R              R      2.0
99996           1    CH         0.0             L              R      1.0
99997           1    FA         0.0             R              L      0.0
99998           0    FA         NaN             R              L      0.0
99999           1    FA         0.0             L              R      1.0

       balls  outs  pitch_plate_location_x  pitch_plate_location_z  \
0        1.0     2                  -0.356                   1.840
1        1.0     2                   0.095                   2.140
2        2.0     0                  -0.556                   2.515
3        1.0     1                  -0.212                   3.107
4        1.0     1                  -0.248                   3.437
```

```
…      …   …                     …                     …
99995    2.0     1                   1.028                    2.376
99996    0.0     2                  -0.784                    2.604
99997    0.0     1                  -0.213                    3.022
99998    0.0     1                  -0.018                    2.732
99999    2.0     2                  -0.008                    2.659

       pitch_initial_speed  pitch_arc_break_x  pitch_arc_break_z  \
0                     87.9           0.404316           -6.27682
1                     81.0           4.821380          -13.18070
2                     91.8           2.652270           -3.52130
3                     92.2           5.926690           -3.88490
4                     83.1          -1.288710           -9.83063
…                    …                …                …
99995                 74.0          -0.175697           -7.11359
99996                 83.9           1.872990           -6.53433
99997                 92.7           2.753500           -4.84395
99998                 87.5           1.426750           -3.95007
99999                 91.4           1.835650           -4.32523

       pitch_spin_rate  inning  pitch_per_atbat  home_team_runs  \
0             1025.700       3                2               1
1             2256.700       7                3               2
2             2457.170       5                5               0
3             3510.320       6                3               0
4              478.459       8                7               4
…                …     …              …              …
99995         1811.010       7                7               5
99996         1417.850       4                2               3
99997         1907.690       9                1               6
99998         2183.500       1                1               0
99999         1919.020       4                4               1

       away_team_runs
0                   0
1                   3
2                   1
3                   0
4                   2
…               …
99995               4
99996               4
99997               2
99998               0
99999              10

[100000 rows x 18 columns]
```

[179]:

```
#Viewing the Columns

df = df.dropna()
print(df.shape)
print(list(df.columns))
```

```
(99254, 18)
['is_in_play', 'pitch', 'three_plus', 'batter_stance', 'pitcher_throws',
'strikes', 'balls', 'outs', 'pitch_plate_location_x', 'pitch_plate_location_z',
'pitch_initial_speed', 'pitch_arc_break_x', 'pitch_arc_break_z',
'pitch_spin_rate', 'inning', 'pitch_per_atbat', 'home_team_runs',
'away_team_runs']
```

[184]:

```
#Turning Categorical Variables into Binary

cat_vars=['pitch','three_plus','batter_stance','pitcher_throws']
for var in cat_vars:
    cat_list='var'+'_'+var
    cat_list = pd.get_dummies(df[var], prefix=var)
```

[189]:

```
df=df1
```

[190]:

```
df
```

[190]:

```
       is_in_play pitch  three_plus batter_stance pitcher_throws  strikes  \
0               1    SL         0.0             R              L      0.0
1               1    CU         0.0             R              R      1.0
2               1    FA         0.0             R              L      2.0
3               1    SI         0.0             R              L      1.0
4               1    SL         0.0             L              R      2.0
…           …   …         …           …            …      …
99994           1    SI         0.0             L              L      2.0
99995           1    CH         0.0             R              R      2.0
99996           1    CH         0.0             L              R      1.0
99997           1    FA         0.0             R              L      0.0
99999           1    FA         0.0             L              R      1.0

       balls  outs  pitch_plate_location_x  pitch_plate_location_z  …  \
0        1.0     2                  -0.356                   1.840  …
1        1.0     2                   0.095                   2.140  …
2        2.0     0                  -0.556                   2.515  …
3        1.0     1                  -0.212                   3.107  …
4        1.0     1                  -0.248                   3.437  …
…      …   …                     …                     …  …
99994    3.0     0                  -0.014                   3.006  …
99995    2.0     1                   1.028                   2.376  …
99996    0.0     2                  -0.784                   2.604  …
```

```
99997    0.0     1                  -0.213                   3.022  …
99999    2.0     2                  -0.008                   2.659  …

       pitch_FA  pitch_FC  pitch_SI  pitch_SL  three_plus_0.0  three_plus_1.0  \
0             0         0         0         1               1               0
1             0         0         0         0               1               0
2             1         0         0         0               1               0
3             0         0         1         0               1               0
4             0         0         0         1               1               0
…         …       …       …       …             …             …
99994         0         0         1         0               1               0
99995         0         0         0         0               1               0
99996         0         0         0         0               1               0
99997         1         0         0         0               1               0
99999         1         0         0         0               1               0

       batter_stance_L  batter_stance_R  pitcher_throws_L  pitcher_throws_R
0                    0                1                 1                 0
1                    0                1                 0                 1
2                    0                1                 1                 0
3                    0                1                 1                 0
4                    1                0                 0                 1
…                …              …               …               …
99994                1                0                 1                 0
99995                0                1                 0                 1
99996                1                0                 0                 1
99997                0                1                 1                 0
99999                1                0                 0                 1

[99254 rows x 30 columns]
```

[191]:

```
cat_vars=['three_plus','batter_stance','pitcher_throws','pitch']
df_vars=df.columns.values.tolist()
to_keep=[i for i in df_vars if i not in cat_vars]
```

[192]:

```
#Viewing New Categories with Binary Variables

df=df[to_keep]
df.columns.values
```

[192]:

```
array(['is_in_play', 'strikes', 'balls', 'outs', 'pitch_plate_location_x',
       'pitch_plate_location_z', 'pitch_initial_speed',
       'pitch_arc_break_x', 'pitch_arc_break_z', 'pitch_spin_rate',
       'inning', 'pitch_per_atbat', 'home_team_runs', 'away_team_runs',
       'pitch_CH', 'pitch_CU', 'pitch_FA', 'pitch_FC', 'pitch_SI',
       'pitch_SL', 'three_plus_0.0', 'three_plus_1.0', 'batter_stance_L',
       'batter_stance_R', 'pitcher_throws_L', 'pitcher_throws_R'],
      dtype=object)
```

[193]:

```
#Identifying the Dependent and Independent Variables

X = df.loc[:, df.columns != 'is_in_play']
y = df.loc[:, df.columns == 'is_in_play']
```

[195]:

```
#Splitting the Data into Train and Test Sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

```
/Users/amrinsinghdhillon/opt/anaconda3/lib/python3.8/site-
packages/sklearn/utils/validation.py:1111: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
/Users/amrinsinghdhillon/opt/anaconda3/lib/python3.8/site-
packages/sklearn/linear_model/_logistic.py:444: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[195]:

```
LogisticRegression()
```

[196]:

```
#Printing Accuracy of the Classifier

y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.
 ↪format(logreg.score(X_test, y_test)))
```

```
Accuracy of logistic regression classifier on test set: 0.68
```

[197]:

```
#Viewing the Classification Report

from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support
```

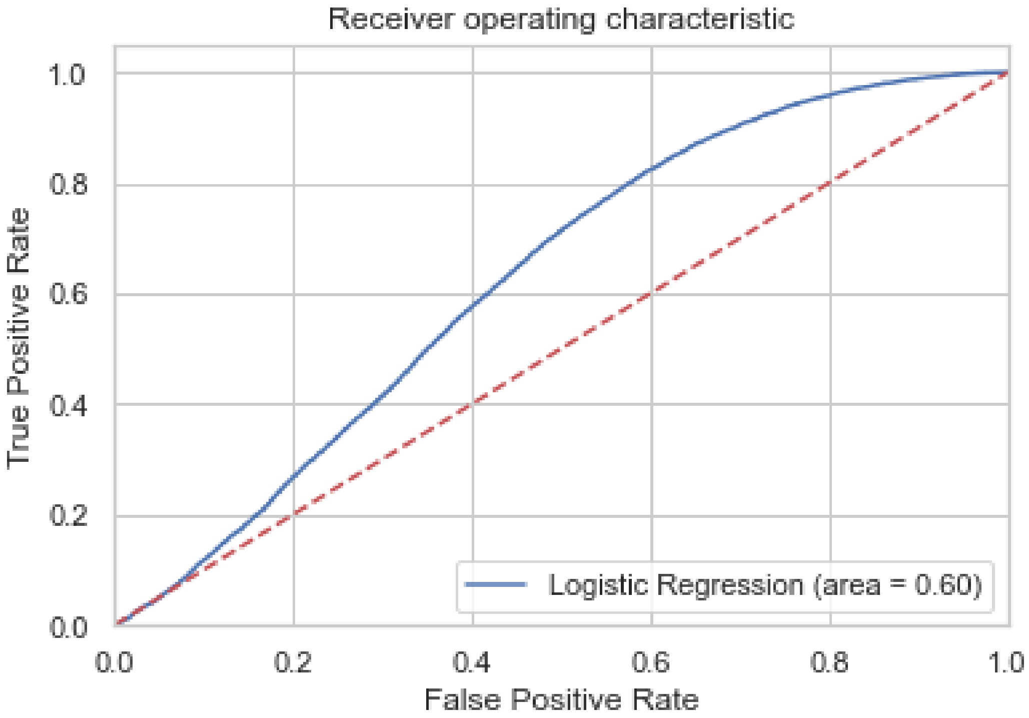
```
           0       0.69      0.27      0.39     11254
           1       0.68      0.93      0.78     18523

    accuracy                           0.68     29777
   macro avg       0.68      0.60      0.58     29777
weighted avg       0.68      0.68      0.63     29777
```

[198]:

```python
#Creating the ROC Curve

from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, logreg.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```



[ ]:

```
```