Logistic Regression in Python

Learning Python

In [282]:
```
import pandas as pd
import numpy as np
import os
from sklearn.model_selection import RepeatedKFold, cross_val_score
from sklearn.linear_model import LogisticRegression
```

In [283]:
```
os.chdir("C:\\Users\\Matt\\Documents\\Python_Projects")
```

In [284]:
```
pwd
```

Out[284]:
```
'C:\\Users\\Matt\\Documents\\Python_Projects'
```

In [285]:
```
baseball_train = pd.read_csv(r"baseball_train.csv",index_col=0,
                             dtype={'Opp': 'category', 'Result': 'category',
'Name': 'category'}, header=0)
baseball_test = pd.read_csv(r"baseball_test.csv",index_col=0,
                            dtype={'Opp': 'category', 'Result': 'category', 'N
ame': 'category'}, header=0)
print(baseball_test.head())
encoded_categories = dict(enumerate(baseball_test.Name.cat.categories))
print(encoded_categories)
```

```
      Opp  DR   IP   H  R  ER  BB  SO  HR  HBP  ...  CS  PO  X2B  X3B  IBB  \
788   DET   4  4.2   3  5   5   7   5   2    0  ...   0   0    0    0    0
1463  DET   5  6.0   7  2   2   1   5   0    0  ...   1   0    1    0    0
1272  TOR   4  9.0  10  5   5   6   6   0    0  ...   1   0    1    0    0
639   PIT   5  7.0   2  0   0   5   6   0    0  ...   1   0    0    0    0
41    ATL   4  2.1   0  0   0   1   0   0    0  ...   0   0    0    0    0

      GDP  SF  ROE  Result   Name
788     0   0    0       L  Nolan
1463    1   0    0       L  Tommy
1272    1   0    0       L  Tommy
639     1   0    0       W  Nolan
41      0   0    1       W  Nolan

[5 rows x 28 columns]
{0: 'Nolan', 1: 'Tommy'}
```

In [286]:
```
X = baseball_train.iloc[:,:-1]
X = X.drop(['Opp','Result'],axis=1)
y = baseball_train.iloc[:,-1]

# Create logistic regression
logit = LogisticRegression(fit_intercept=True)

# Create repeated kfold
rkf = RepeatedKFold(n_splits=5, n_repeats=10, random_state=21191)

# Do repeated k-fold cross-validation
cv_results = cross_val_score(logit,
                             X,
                             y,
                             cv=rkf,
                             scoring="roc_auc")
```

Repeated K-Fold Cross Validation

In [287]:
```
print(cv_results.min())
print(np.percentile(cv_results, 25))
print(cv_results.mean())
print(np.percentile(cv_results, 50))
print(np.percentile(cv_results, 75))
print(cv_results.max())
```

```
0.952828605002518
0.9613817469462649
0.9669297585406852
0.9662099882688117
0.9729831391245957
0.9800403395243298
```

In [288]:
```
model = logit.fit(X,y)

intercept = model.intercept_[0]

print("intercept = {}".format(intercept))
for idx, col_name in enumerate(X.columns):
    print("{} = {}".format(col_name, model.coef_[0][idx]))
```

```
intercept = 1.2946166338126635
DR = 0.011732883825072156
IP = -0.23226847104529907
H = 0.20288283686541814
R = -0.34776710286808105
ER = 0.0028960956992052796
BB = -0.3200886417413454
SO = -0.3132016093632194
HR = 0.08526653709614487
HBP = 0.0710202234769784
ERA = -0.10336335576210107
BF = -0.21558563340441994
GB = 0.726929231994203
FB = -0.09150477706222566
LD = 0.30735242703011767
PU = -0.4865137756269206
Unk = 0.5177317713205674
SB = -0.5546505737669927
CS = 0.25899883409993363
PO = -0.8674866862900049
X2B = 0.0006321819834886793
X3B = -0.04613833936871116
IBB = 0.571862684825111
GDP = 0.19170268600140106
SF = -0.07018569499658994
ROE = -0.03986918217233402
```

In [289]:
```
Xnew = baseball_test.iloc[:,:-1]
Xnew = Xnew.drop(['Opp','Result'],axis=1)
yTrue = baseball_test.iloc[:,-1]

# make a prediction
ynew = model.predict(Xnew)
# show the inputs and predicted outputs
#for i in range(len(Xnew)):
#    print("Predicted=%s" % (ynew[i]))

baseball = {'predicted': ynew, 'truth': yTrue}
pd.DataFrame(data=baseball)
```

Out[289]:

	predicted	truth
788	Nolan	Nolan
1463	Tommy	Tommy
1272	Nolan	Tommy
639	Nolan	Nolan
41	Tommy	Nolan
391	Nolan	Nolan
779	Nolan	Nolan
1457	Tommy	Tommy
496	Nolan	Nolan
678	Nolan	Nolan
358	Nolan	Nolan
67	Nolan	Nolan
1185	Tommy	Tommy
1096	Nolan	Tommy
946	Nolan	Tommy
911	Nolan	Tommy
1542	Tommy	Tommy
324	Nolan	Nolan
955	Tommy	Tommy
206	Nolan	Nolan