In [62]:

```
#Upload Packages to Use
import pandas as pd
import numpy as np
from sklearn.metrics import classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

In [3]:

```
#Deploy CSV

dply_csv = pd.read_csv('deploy.csv')
dply_csv.head()
```

Out[3]:

	Velo	SpinRate	HorzBreak	InducedVertBreak
0	94.72	2375.0	3.10	18.15
1	95.25	2033.0	11.26	14.50
2	92.61	2389.0	11.00	21.93
3	94.94	2360.0	6.84	18.11
4	97.42	2214.0	16.70	13.38

In [83]:

```
#Training CSV

trn_csv = pd.read_csv('training.csv')


trn_new = trn_csv[np.isfinite(trn_csv).all(1)]

trn_new.head()
```

Out[83]:

	InPlay	Velo	SpinRate	HorzBreak	InducedVertBreak
0	0	95.33	2893.0	10.68	21.33
1	0	94.41	2038.0	17.13	5.77
2	0	90.48	2183.0	6.61	15.39
3	0	93.04	2279.0	9.33	14.57
4	0	95.17	2384.0	6.99	17.62

In [82]:

```
#Create InPlay column for the deploy csv

dply_csv['InPlay'] = trn_new['InPlay']

dply_csv.head()
```

Out[82]:

	Velo	SpinRate	HorzBreak	InducedVertBreak	InPlay
0	94.72	2375.0	3.10	18.15	0.0

	Velo	SpinRate	HorzBreak	InducedVertBreak	InPlay
1	95.25	2033.0	11.26	14.50	0.0
2	92.61	2389.0	11.00	21.93	0.0
3	94.94	2360.0	6.84	18.11	0.0
4	97.42	2214.0	16.70	13.38	0.0

In [85]:

```
#Clean Deploy csv of NaN or Infinite values
dply_new = dply_csv[np.isfinite(dply_csv).all(1)]
```

In [88]:

```
#Train Deploy dataframe with logistic regression model
x = dply_new.drop(['InPlay'], axis=1)
y = dply_new['InPlay']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=0)


lr = LogisticRegression()

lr.fit(x_train, y_train)
```

Out[88]:

```
LogisticRegression()
```

In [106…

```
#Predict Probability of In Play vs. Out Play
predict = lr.predict(x_test)
data = print(classification_report(y_test, predict, zero_division = 0))
```

```
              precision    recall  f1-score   support

         0.0       0.74      1.00      0.85      2205
         1.0       0.00      0.00      0.00       790

    accuracy                           0.74      2995
   macro avg       0.37      0.50      0.42      2995
weighted avg       0.54      0.74      0.62      2995
```