In [1]:
```
import pandas as pd

# load dataset
ds = pd.read_csv("Ant18Data.csv")
ds.head()
```

Out[1]:

	COMP	LOC	WMC	DIT	NO
0	org.apache.tools.ant.util.regexp.JakartaRegexp...	-0.451532	-0.654095	-0.658290	-0.14695
1	org.apache.tools.ant.TypeAdapter.java	-0.655176	-0.502131	0.715241	-0.14695
2	org.apache.tools.ant.taskdefs.optional.perforc...	-0.538808	-0.350167	-0.658290	-0.14695
3	org.apache.tools.ant.taskdefs.XSLTLoggerAware....	-0.664128	-0.806060	0.715241	-0.14695
4	org.apache.tools.ant.taskdefs.optional.perforc...	0.074362	0.257691	-0.658290	2.83092

In [85]:
```
#ant18_input_features=["MFA","CAM","MOA","LCOM","AMC","CE","DAM","NOC",'
#ant18_input_features=["CAM","MOA","LCOM","AMC","CE","DAM","NOC","CA","M
#ant18_input_features=["CAM","MOA","LCOM","AMC","CE","DAM","NOC","CA","M
#ant18_input_features=["CAM","MOA","LCOM","AMC","CE","DAM","NOC","CA"]
#ant18_input_features=["CAM","MOA","LCOM","AMC","CE","DAM","NOC"]
#ant18_input_features=["CAM","MOA","LCOM","AMC","CE","DAM"]
#ant18_input_features=["CAM","MOA","LCOM","AMC","CE"]
#ant18_input_features=["CAM","MOA","LCOM","AMC"]
ant18_input_features=["CAM","MOA","LCOM"]#optimal
#ant18_input_features=["CAM","MOA"]
#ant18_input_features=["CAM","LCOM"]
#ant18_input_features=["MOA","LCOM"]



X = ds[ant18_input_features] # Features
y = ds.Sum_Churn # Target variable
```

In [86]:
```
#calculate Variance Inflation Factor
from statsmodels.stats.outliers_influence import variance_inflation_fact
vif_scores = pd.DataFrame()
vif_scores["Attribute"] = X.columns

# calculating VIF for each feature
vif_scores["VIF Scores"] = [variance_inflation_factor(X.values, i) for i

display(vif_scores)
```

	Attribute	VIF Scores
0	CAM	1.361667
1	MOA	1.445768
2	LCOM	1.424016

In [87]:
```
import statsmodels.api as sm
smlog = sm.Logit(y,sm.add_constant(X)).fit(maxiter=4000000)

smlog.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.594605
         Iterations 6
```

Out[87]:

Logit Regression Results

Dep. Variable:	Sum_Churn	**No. Observations:**	711
Model:	Logit	**Df Residuals:**	707
Method:	MLE	**Df Model:**	3
Date:	Wed, 17 Apr 2024	**Pseudo R-squ.:**	0.1231
Time:	22:33:51	**Log-Likelihood:**	-422.76
converged:	True	**LL-Null:**	-482.13
Covariance Type:	nonrobust	**LLR p-value:**	1.441e-25

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0922	0.116	-0.795	0.426	-0.320	0.135
CAM	-0.5191	0.111	-4.671	0.000	-0.737	-0.301
MOA	0.6035	0.157	3.837	0.000	0.295	0.912
LCOM	1.0417	0.482	2.160	0.031	0.096	1.987

In [88]:
```
#odds ratio
import numpy as np
np.exp(smlog.params)
```

Out[88]:
```
const    0.911903
CAM      0.595035
MOA      1.828483
LCOM     2.833893
dtype: float64
```

In [89]:
```
from scipy.stats.distributions import chi2
def likelihood_ratio(reduced_ll, full_ll):
    return(-2*(reduced_ll-full_ll))

afterll=-434.50


beforell=-422.76









LR = likelihood_ratio(afterll,beforell)
p = chi2.sf(LR, 1) # 1 DoF coz diff between variable in model



print(LR)
print(p)
```

```
23.480000000000018
1.2621907160090113e-06
```

In [ ]:

In [ ]:

In [ ]: