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

In [2]:
```
df=pd.read_csv('df_autonomus.csv')
df
col1=list(df.self_driving)
col1
binary=[]
```

In [3]:
```
for i in col1:
    if i=="Yes":
        binary.append(1)
    else:
        binary.append(0)
binary
```

Out[3]:
```
[0,
 1,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 1,
 0,
 1,
 0,
 0,
 0,
 0,
 0,
 0,
 1,
 0,
 0,
 0,
 0,
 1,
 1,
 1,
 0,
 0,
 0,
 0,
 1,
 0,
 0,
 1,
 0,
 0,
 0,
 1,
 0,
 0,
 1,
 0,
 0,
 0,
 1,
 0,
 0,
 1,
 0,
 0,
 0,
 1,
 0,
 1,
 1,
 0,
 0,
 1,
 0,
 0,
 0,
 1,
 1,
 1,
```

1,
1,
1,
1,
0,
1,
1,
0,
1,
0,
0,
0,
1,
1,
0,
1,
1,
1,
1,
1,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
1,
1,
0,
0,
0,
1,
0,
1,
1,
1,
1,
1,
1,
1,
0,
1,
1,
1,
1,
1,
0,
1,
0,
0,
0,
1,
1,
0,
1,
1,
0,
1,
0,
1,
0,
1,

```
1,
0,
1,
1,
1,
1,
1,
0,
0,
1,
0,
1,
0,
0,
0,
1,
1,
0,
0,
0,
1,
0,
1,
1,
1,
1,
1,
0,
0,
1,
0,
0,
0,
1,
1,
1,
0,
1,
1,
0,
0,
0,
1,
0,
1,
0,
1,
0,
1,
1,
0,
1,
1,
1,
0,
0,
0,
1,
0,
1,
0,
1,
0,
0,
1,
1,
0,
1,
```

0,
1,
0,
0,
0,
1,
0,
0,
0,
1,
1,
1,
1,
1,
0,
0,
0,
0,
1,
0,
1,
1,
0,
1,
0,
0,
1,
1,
0,
0,
0,
0,
0,
1,
0,
0,
0,
0,
0,
0,
1,
1,
0,
0,
0,
0,
0,
1,
1,
0,
0,
0,
0,
0,
0,
0,
1,
0,
1,
1,
1,
0,
1,
1,
1,
0,
0,

```
1,
1,
1,
1,
0,
1,
0,
1,
0,
1,
0,
1,
0,
1,
1,
1,
0,
1,
0,
0,
1,
0,
0,
0,
1,
0,
1,
0,
0,
1,
0,
0,
0,
0,
0,
0,
0,
0,
1,
0,
1,
1,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
1,
1,
1,
0,
1,
0,
1,
1,
1,
1,
1,
0,
1,
1,
```

```
1,
1,
1,
1,
0,
1,
1,
0,
1,
0,
1,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
1,
0,
1,
0,
1,
1,
1,
0,
0,
0,
1,
1,
0,
1,
0,
1,
0,
1,
1,
0,
1,
1,
1,
0,
0,
1,
1,
1,
1,
0,
1,
0,
1,
1,
1,
0,
0,
1,
0,
1,
0,
```

0,
1,
1,
1,
1,
1,
1,
1,
0,
0,
0,
1,
0,
0,
0,
0,
0,
0,
0,
1,
0,
0,
0,
0,
0,
0,
1,
0,
0,
0,
1,
1,
0,
1,
0,
0,
1,
0,
0,
0,
1,
0,
1,
0,
0,
0,
1,
1,
1,
0,
1,
1,
1,
0,
1,
0,
0,
1,
0,
0,
0,
0,
1,
0,
0,
0,
0,
0,
1,
0,
0,
0,
0,
0,
0,
1,

1,
1,
1,
1,
0,
1,
1,
1,
1,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
1,
0,
0,
0,
1,
0,
0,
0,
1,
0,
1,
1,
1,
1,
0,
0,
1,
0,
0,
0,
1,
0,
0,
0,
0,
1,
0,
0,
0,
0,
0,
1,
0,
1,
1,
1,
1,
1,
1,
0,
0,
0,
1,
1,
1,
0,
1,

0,
1,
1,
0,
0,
0,
0,
1,
1,
0,
1,
1,
1,
0,
1,
0,
0,
0,
0,
1,
0,
0,
1,
1,
1,
0,
1,
0,
0,
1,
1,
0,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
0,
1,
0,
1,
0,
1,
0,
0,
1,
1,
1,
1,
0,
1,
1,
0,
1,

```
1,
1,
0,
0,
0,
0,
0,
1,
1,
1,
0,
0,
1,
0,
1,
1,
0,
1,
1,
1,
1,
1,
1,
1,
1,
0,
0,
0,
0,
1,
1,
0,
0,
1,
1,
0,
0,
0,
1,
0,
0,
1,
1,
0,
1,
0,
1,
0,
0,
0,
0,
0,
0,
0,
1,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
1,
0,
1,
```

0,
1,
1,
0,
1,
0,
1,
1,
1,
0,
1,
0,
0,
1,
0,
0,
0,
1,
1,
0,
1,
0,
1,
1,
1,
1,
0,
1,
1,
1,
0,
1,
1,
1,
1,
0,
1,
0,
1,
1,
0,
0,
0,
1,
1,
1,
0,
1,
1,
1,
1,
1,
0,
1,
0,
0,
1,
1,
1,
0,
1,
0,
0,
0,
1,
1,

```
1,
0,
0,
0,
0,
0,
0,
0,
0,
1,
1,
1,
0,
0,
1,
1,
1,
0,
1,
0,
1,
0,
0,
0,
1,
0,
0,
1,
0,
1,
0,
1,
0,
0,
0,
1,
0,
0,
0,
0,
1,
0,
0,
0,
1,
0,
0,
1,
0,
0,
0,
1,
1,
0,
0,
0,
1,
1,
0,
1,
```

0,
1,
0,
1,
0,
1,
0,
0,
1,
0,
0,
0,
1,
0,
1,
1,
1,
0,
1,
1,
0,
1,
0,
1,
0,
0,
0,
1,
0,
0,
0,
0,
0,
1,
0,
1,
0,
1,
0,
1,
1,
1,
0,
0,
1,
0,
0,
0,
1,
1,
0,
1,
0,
1,
1,
0,
0,
0,
0,
0,
0,
1,
0,
1,
1,

```
1,
0,
0,
0,
1,
1,
0,
1,
1,
0,
0,
0,
1,
1,
0,
1,
1,
0,
1,
1,
0,
0,
1,
0,
1,
1,
0,
1,
1,
1,
0,
1,
0,
0,
1,
1,
0,
0,
1,
0,
0,
0,
0,
0,
0,
0,
1,
0,
1,
0,
1,
0,
1,
1,
0,
1,
1,
1,
0,
1,
0,
1,
1,
0,
0,
1,
```

0,
0,
1,
1,
0,
1,
1,
0,
1,
1,
0,
0,
0,
0,
1,
0,
0,
1,
0,
0,
0,
0,
0,
1,
0,
0,
0,
0,
1,
0,
0,
1,
0,
0,
0,
0,
1,
1,
0,
0,
0,
1,
1,
0,
0,
0,
1,
0,
0,
0,
1,
0,
0,
0,
0,
0,
1,
0,
0,
1,
0,

```
1,
0,
0,
0,
1,
0,
0,
1,
1,
0,
0,
0,
0,
0,
0,
1,
0,
1,
0,
0,
0,
0,
0,
0,
1,
0,
1,
1,
1,
1,
0,
0,
1,
0,
0,
1,
0,
0,
0,
0,
0,
1,
...]
```

In [4]:
```
df['new']=binary
df.head(10)
df.columns
```

Out[4]:
```
Index(['self_driving', 'safety_concern', 'carshare_level', 'rideshare_level',
       'pevowner', 'income', 'householdvehicles', 'housing', 'parkingtype',
       'new'],
      dtype='object')
```

In [5]:
```
resp=df['new']
import statsmodels.formula.api as smf
model = smf.logit(formula = 'new ~ safety_concern +carshare_level+ rideshare_level+
model.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.615207
         Iterations 6
```

Out[5]:

Logit Regression Results

Dep. Variable:	new	**No. Observations:**	3614
Model:	Logit	**Df Residuals:**	3598
Method:	MLE	**Df Model:**	15
Date:	Sun, 20 Nov 2022	**Pseudo R-squ.:**	0.1009
Time:	19:49:00	**Log-Likelihood:**	-2223.4
converged:	True	**LL-Null:**	-2472.8
Covariance Type:	nonrobust	**LLR p-value:**	9.765e-97

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.6738	0.201	-3.352	0.001	-1.068	-0.280
safety_concern[T.Disagree]	0.9348	0.130	7.169	0.000	0.679	1.190
safety_concern[T.Neutral]	-0.7997	0.183	-4.367	0.000	-1.159	-0.441
carshare_level[T.Interested in participating]	-1.1363	0.141	-8.064	0.000	-1.413	-0.860
carshare_level[T.Not participating]	-0.6290	0.086	-7.286	0.000	-0.798	-0.460
rideshare_level[T.Interested in participating]	-0.2853	0.108	-2.641	0.008	-0.497	-0.074
rideshare_level[T.Not participating]	-0.3080	0.090	-3.415	0.001	-0.485	-0.131
pevowner[T.Yes]	-0.7011	0.146	-4.794	0.000	-0.988	-0.414
income[T.Low income]	0.3845	0.121	3.185	0.001	0.148	0.621
income[T.Middle income]	0.3431	0.101	3.382	0.001	0.144	0.542
householdvehicles[T.Three plus]	0.2899	0.108	2.674	0.007	0.077	0.502
householdvehicles[T.Two vehicle]	0.0926	0.086	1.076	0.282	-0.076	0.261
housing[T.Mobile based home]	-0.0124	0.277	-0.045	0.964	-0.556	0.531
housing[T.Others]	0.2625	0.580	0.453	0.651	-0.874	1.400
housing[T.Single family]	-0.1062	0.116	-0.913	0.361	-0.334	0.122
parkingtype[T.Shared parking]	-0.0326	0.114	-0.287	0.774	-0.255	0.190

INTERPRETATION Since p value <=0.05 for safety_concern,carshare_level,rideshare_level,pevowner,income and householdvehicles(Three plus) so these independent variables are significant

new model(discarding non significant values)

In [7]:

```
# For the following columns the influence on the 'new' column or self driving
# will be zero or minimum since the values from 0.025 to 0.975 are changing
# from negative to positive.

#householdvehicles[T.Two vehicle]
#housing[T.Mobile based home]
#housing[T.Others]
```

```
#housing[T.Single family]
#parkingtype[T.Shared parking]
```

In [6]:
```
model1 = smf.logit(formula = 'new ~ safety_concern +carshare_level+ rideshare_level
model1.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.615388
         Iterations 6
```

Out[6]:

Logit Regression Results

Dep. Variable:	new	**No. Observations:**	3614
Model:	Logit	**Df Residuals:**	3602
Method:	MLE	**Df Model:**	11
Date:	Sun, 20 Nov 2022	**Pseudo R-squ.:**	0.1006
Time:	19:49:03	**Log-Likelihood:**	-2224.0
converged:	True	**LL-Null:**	-2472.8
Covariance Type:	nonrobust	**LLR p-value:**	1.058e-99

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.7517	0.173	-4.356	0.000	-1.090	-0.413
safety_concern[T.Disagree]	0.9306	0.130	7.150	0.000	0.675	1.186
safety_concern[T.Neutral]	-0.8031	0.183	-4.390	0.000	-1.162	-0.445
carshare_level[T.Interested in participating]	-1.1349	0.141	-8.061	0.000	-1.411	-0.859
carshare_level[T.Not participating]	-0.6282	0.086	-7.285	0.000	-0.797	-0.459
rideshare_level[T.Interested in participating]	-0.2828	0.108	-2.619	0.009	-0.494	-0.071
rideshare_level[T.Not participating]	-0.3056	0.090	-3.400	0.001	-0.482	-0.129
pevowner[T.Yes]	-0.7029	0.146	-4.808	0.000	-0.989	-0.416
income[T.Low income]	0.3967	0.120	3.320	0.001	0.162	0.631
income[T.Middle income]	0.3480	0.101	3.439	0.001	0.150	0.546
householdvehicles[T.Three plus]	0.2634	0.105	2.511	0.012	0.058	0.469
householdvehicles[T.Two vehicle]	0.0730	0.084	0.872	0.383	-0.091	0.237

In this output, all independent variables are statistically significant and the signs are logical, so this model can be used for further diagnosis.

In [7]:
```
#calculating odds ratio
import numpy as np
conf = model1.conf_int()
conf['OR'] = model1.params
conf.columns = ['2.5%', '97.5%', 'OR']
print(np.exp(conf))
```

```
                                                 2.5%     97.5%        OR
Intercept                                    0.336222  0.661340  0.471548
safety_concern[T.Disagree]                   1.964963  3.272932  2.535979
safety_concern[T.Neutral]                    0.312969  0.641128  0.447943
carshare_level[T.Interested in participating]  0.243954  0.423611  0.321468
carshare_level[T.Not participating]          0.450588  0.631803  0.533556
rideshare_level[T.Interested in participating]  0.609923  0.931284  0.753666
rideshare_level[T.Not participating]         0.617666  0.878570  0.736657
pevowner[T.Yes]                              0.371810  0.659434  0.495161
income[T.Low income]                         1.176418  1.879326  1.486901
income[T.Middle income]                      1.161440  1.727050  1.416286
householdvehicles[T.Three plus]              1.059466  1.598391  1.301323
householdvehicles[T.Two vehicle]             0.912944  1.267628  1.075766
```

In [8]:
```
#Predicting probabilities
from sklearn.metrics import confusion_matrix
predicted_values1 = model1.predict()
threshold=0.5
predicted_class1=np.zeros(predicted_values1.shape)
predicted_class1[predicted_values1>threshold]=1
cm1 = confusion_matrix(df['new'],predicted_class1)
print('Confusion Matrix : \n', cm1)
```

```
Confusion Matrix :
 [[1514  534]
 [ 691  875]]
```

In [5]:
```
# From confusion matrix we can say that 1514 are correctly predicted
# as not self_driving(True Negative) and 875 correctly predicted
# as self_driving (True Positive).

# 534 were predicted as self driving but all these were observed
#as non-self driving.(False Positive)

# 691 instances were predicted as non-self driving
#but were self-driving(False Negative)
```

In [6]:
```
#From confusion matrix we can say that 1514 are correctly
#predicted as not self_driving and 875 correctly predicted
#self_driving

#The precision and recall values of the model are routinely
#assessed in a classification model.

#Precision tells us what percentage of predicted positive cases
#are correctly predicted.
#Recall tells us what percentage of actual positive cases are correctly predicted.
```

In [9]:
```
from sklearn.metrics import classification_report
print(classification_report(df['new'],predicted_class1))
```

```
              precision    recall  f1-score   support

           0       0.69      0.74      0.71      2048
           1       0.62      0.56      0.59      1566

    accuracy                           0.66      3614
   macro avg       0.65      0.65      0.65      3614
weighted avg       0.66      0.66      0.66      3614
```

In [4]:
```
# Accuracy=66%
# Precision is 62% and Recall is 56% for self driving affinity
```

In [3]:
```
#The precision and recall values of the model
#are routinely assessed in a classification model.

#Precision tells us what percentage of predicted positive cases
#are correctly predicted.
#Recall tells us what percentage of actual positive cases are correctly predicted.
```

In [10]:
```
import numpy as np
import pandas as pd
```

In [11]:
```
df=pd.read_csv('df_autonomus.csv')
df
col1=list(df.self_driving)
col1

MNLogit=[]
```

In [12]:
```
for i in col1:
    if i=='Yes':
        MNLogit.append(1)
    elif i== 'Neutral':
        MNLogit.append(2)
    else:
        MNLogit.append(3)
MNLogit
```

Out[12]:
```
[3,
 1,
 3,
 3,
 2,
 2,
 3,
 2,
 3,
 2,
 3,
 3,
 2,
 1,
 2,
 1,
 3,
 2,
 2,
 2,
 3,
 1,
 2,
 2,
 2,
 1,
 1,
 1,
 3,
 2,
 2,
 3,
 1,
 2,
 2,
 1,
 2,
 3,
 1,
 3,
 2,
 1,
 3,
 3,
 1,
 1,
 2,
 1,
 2,
 3,
 3,
 1,
 2,
 1,
 1,
 2,
 3,
 1,
 2,
 2,
 3,
 1,
 1,
 1,
```

1,
1,
1,
1,
2,
1,
1,
3,
1,
2,
3,
1,
1,
2,
1,
1,
1,
1,
3,
3,
3,
2,
2,
2,
2,
2,
2,
3,
2,
1,
1,
2,
3,
3,
1,
2,
1,
1,
1,
1,
1,
1,
2,
1,
1,
1,
1,
1,
3,
1,
3,
3,
3,
1,
1,
3,
1,
1,
3,
3,
1,
3,
1,
3,
1,

1,
2,
1,
1,
1,
1,
3,
3,
1,
3,
3,
1,
3,
3,
2,
1,
1,
3,
3,
3,
1,
3,
1,
1,
1,
1,
1,
2,
3,
1,
3,
3,
2,
1,
1,
1,
2,
1,
1,
1,
3,
3,
3,
3,
1,
3,
1,
3,
1,
3,
1,
2,
1,
1,
2,
2,
1,
1,
3,
2,
2,
1,
2,
1,
2,
1,
2,
2,
1,
1,
3,
1,

2,
1,
3,
2,
2,
1,
3,
3,
3,
1,
1,
1,
1,
1,
2,
2,
3,
3,
1,
2,
1,
1,
2,
1,
3,
3,
1,
1,
3,
2,
2,
3,
1,
2,
3,
3,
3,
2,
1,
1,
2,
2,
2,
2,
1,
1,
3,
3,
2,
3,
2,
2,
3,
1,
3,
1,
1,
1,
3,
1,
1,
1,
2,
3,

1,
1,
1,
1,
2,
1,
3,
1,
3,
1,
3,
1,
2,
1,
1,
1,
3,
1,
3,
2,
1,
3,
3,
2,
1,
3,
1,
3,
3,
1,
3,
3,
3,
3,
2,
3,
3,
1,
2,
1,
1,
3,
3,
3,
2,
3,
3,
3,
3,
2,
1,
1,
1,
2,
1,
3,
1,
1,
1,
1,
1,
3,
1,
1,

1,
1,
1,
1,
3,
1,
1,
2,
1,
3,
1,
3,
2,
2,
3,
3,
3,
2,
3,
3,
2,
3,
3,
1,
3,
1,
2,
1,
1,
1,
2,
3,
3,
1,
1,
2,
1,
3,
1,
2,
1,
1,
2,
1,
1,
1,
2,
3,
1,
1,
1,
1,
2,
1,
2,
1,
1,
1,
3,
3,
1,
2,
1,
3,

3,
1,
1,
1,
1,
1,
1,
1,
2,
3,
3,
1,
2,
3,
3,
2,
2,
2,
1,
3,
3,
3,
2,
3,
1,
2,
3,
3,
1,
1,
2,
1,
2,
3,
1,
2,
3,
1,
2,
1,
3,
2,
1,
1,
1,
2,
1,
1,
1,
1,
3,
1,
3,
3,
1,
3,
2,
3,
1,
2,
3,
2,
3,
2,
1,

1,
1,
1,
1,
3,
1,
1,
1,
1,
3,
2,
3,
2,
3,
3,
2,
3,
3,
3,
3,
2,
1,
3,
3,
2,
1,
3,
3,
2,
1,
2,
1,
1,
1,
1,
3,
2,
1,
2,
2,
1,
3,
3,
3,
1,
3,
2,
3,
3,
1,
2,
1,
1,
1,
1,
1,
2,
3,
3,
1,
1,
1,
2,
1,

2,
1,
1,
3,
2,
2,
3,
1,
1,
2,
2,
1,
1,
1,
1,
2,
1,
2,
2,
3,
3,
2,
1,
2,
2,
1,
1,
3,
2,
1,
1,
1,
3,
3,
1,
3,
3,
1,
1,
1,
2,
2,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
2,
2,
1,
2,
1,
3,
1,
2,
2,
3,
1,
1,
1,
1,
1,
3,
1,
1,
1,
3,
1,

1,
1,
3,
3,
3,
3,
3,
1,
1,
1,
2,
2,
1,
3,
1,
1,
2,
1,
1,
1,
1,
1,
1,
1,
2,
2,
2,
3,
1,
1,
2,
2,
1,
1,
3,
3,
2,
1,
2,
2,
1,
1,
2,
1,
2,
1,
2,
2,
2,
2,
2,
2,
1,
3,
2,
3,
3,
3,
3,
3,
3,
3,
1,
2,
1,

2,
1,
1,
3,
1,
3,
1,
1,
1,
3,
1,
2,
2,
1,
3,
2,
3,
1,
1,
3,
1,
2,
1,
1,
1,
3,
1,
1,
1,
2,
1,
1,
1,
1,
3,
1,
3,
1,
1,
3,
3,
2,
1,
1,
1,
2,
1,
1,
1,
1,
1,
3,
1,
3,
3,
1,
1,
1,
2,
1,
3,
3,
3,
1,
1,

1,
3,
2,
3,
2,
2,
3,
3,
3,
1,
1,
1,
3,
2,
1,
1,
1,
2,
1,
3,
1,
3,
3,
2,
1,
3,
2,
1,
2,
1,
3,
1,
3,
3,
3,
1,
3,
3,
3,
3,
1,
3,
2,
2,
1,
3,
3,
1,
2,
2,
2,
1,
2,
3,
2,
1,
1,
3,
2,
3,
1,
1,
3,
1,

3,
1,
3,
1,
3,
1,
3,
2,
1,
3,
2,
2,
2,
1,
2,
1,
1,
1,
3,
1,
1,
2,
1,
2,
1,
2,
2,
2,
2,
1,
3,
2,
3,
3,
1,
3,
1,
3,
1,
3,
1,
1,
1,
1,
2,
3,
1,
3,
3,
3,
1,
1,
1,
2,
1,
2,
1,
1,
3,
2,
2,
3,
3,
3,
1,
3,
1,
1,

1,
3,
3,
3,
1,
1,
2,
1,
1,
2,
3,
3,
1,
1,
3,
1,
1,
3,
1,
1,
3,
3,
1,
3,
1,
1,
2,
1,
1,
1,
2,
1,
3,
3,
1,
1,
3,
3,
1,
3,
2,
3,
3,
2,
2,
1,
2,
1,
2,
1,
2,
1,
1,
3,
1,
1,
2,
1,
2,
1,
1,
3,
2,
1,

2,
3,
1,
1,
3,
1,
1,
3,
1,
1,
2,
3,
3,
2,
1,
3,
2,
1,
3,
2,
2,
3,
3,
1,
2,
3,
3,
3,
1,
3,
2,
1,
2,
3,
1,
3,
2,
3,
3,
1,
1,
3,
2,
1,
1,
3,
2,
2,
1,
3,
3,
3,
1,
2,
3,
3,
3,
2,
1,
3,
3,
3,
1,
3,

```
1,
3,
3,
3,
1,
3,
3,
1,
1,
3,
3,
2,
2,
2,
1,
2,
1,
2,
2,
3,
3,
3,
1,
3,
1,
1,
1,
1,
3,
3,
1,
2,
3,
1,
3,
2,
2,
2,
3,
1,
...]
```

In [13]:
```
df['MNLogit_new']=MNLogit
df.head(10)
df.columns
```

Out[13]:
```
Index(['self_driving', 'safety_concern', 'carshare_level', 'rideshare_level',
       'pevowner', 'income', 'householdvehicles', 'housing', 'parkingtype',
       'MNLogit_new'],
      dtype='object')
```

In [15]:
```
from statsmodels.discrete.discrete_model import MNLogit
formula = 'MNLogit_new ~ safety_concern +carshare_level+ rideshare_level+pevowner+
model_MNL = MNLogit.from_formula(formula=formula , data = df ).fit()
print(model_MNL.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.958242
         Iterations 6
                          MNLogit Regression Results
==============================================================================
Dep. Variable:            MNLogit_new   No. Observations:                 3614
Model:                        MNLogit   Df Residuals:                     3582
Method:                           MLE   Df Model:                           30
Date:                Sun, 20 Nov 2022   Pseudo R-squ.:                 0.09351
Time:                        19:51:21   Log-Likelihood:                -3463.1
converged:                       True   LL-Null:                       -3820.3
Covariance Type:            nonrobust   LLR p-value:                4.721e-131
==============================================================================
================================
                                  MNLogit_new=2       coef    std err          z
P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
--------------------------------
Intercept                                        -1.4423      0.294     -4.903
0.000      -2.019      -0.866
safety_concern[T.Disagree]                        0.2115      0.218      0.972
0.331      -0.215       0.638
safety_concern[T.Neutral]                         2.1754      0.255      8.533
0.000       1.676       2.675
carshare_level[T.Interested in participating]     0.7351      0.184      3.999
0.000       0.375       1.095
carshare_level[T.Not participating]               0.5655      0.111      5.072
0.000       0.347       0.784
rideshare_level[T.Interested in participating]    0.0518      0.138      0.374
0.708      -0.219       0.323
rideshare_level[T.Not participating]             -0.0454      0.116     -0.390
0.697      -0.274       0.183
pevowner[T.Yes]                                   0.2319      0.200      1.161
0.246      -0.159       0.623
income[T.Low income]                             -0.1716      0.156     -1.101
0.271      -0.477       0.134
income[T.Middle income]                          -0.1523      0.133     -1.147
0.251      -0.412       0.108
householdvehicles[T.Three plus]                  -0.2729      0.141     -1.941
0.052      -0.549       0.003
householdvehicles[T.Two vehicle]                 -0.0653      0.109     -0.599
0.549      -0.279       0.148
housing[T.Mobile based home]                     -0.0654      0.384     -0.170
0.865      -0.818       0.688
housing[T.Others]                                 0.4040      0.617      0.655
0.512      -0.804       1.612
housing[T.Single family]                          0.2379      0.150      1.591
0.112      -0.055       0.531
parkingtype[T.Shared parking]                     0.2292      0.143      1.605
0.109      -0.051       0.509
------------------------------------------------------------------------------
--------------------------------
                                  MNLogit_new=3       coef    std err          z
P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
--------------------------------
Intercept                                         0.5373      0.219      2.457
0.014       0.109       0.966
safety_concern[T.Disagree]                       -1.3226      0.137     -9.688
0.000      -1.590      -1.055
safety_concern[T.Neutral]                         0.1995      0.194      1.029
0.304      -0.181       0.580
carshare_level[T.Interested in participating]     1.3260      0.150      8.813
0.000       1.031       1.621
```

```
carshare_level[T.Not participating]                0.6690      0.097      6.886
0.000       0.479       0.859
rideshare_level[T.Interested in participating]     0.4749      0.123      3.857
0.000       0.234       0.716
rideshare_level[T.Not participating]               0.5506      0.104      5.316
0.000       0.348       0.754
pevowner[T.Yes]                                    0.8897      0.154      5.775
0.000       0.588       1.192
income[T.Low income]                              -0.5079      0.135     -3.775
0.000      -0.772      -0.244
income[T.Middle income]                           -0.4464      0.111     -4.015
0.000      -0.664      -0.228
householdvehicles[T.Three plus]                   -0.2947      0.122     -2.406
0.016      -0.535      -0.055
householdvehicles[T.Two vehicle]                  -0.1050      0.097     -1.079
0.280      -0.296       0.086
housing[T.Mobile based home]                       0.0355      0.312      0.114
0.909      -0.576       0.647
housing[T.Others]                                 -1.6333      1.109     -1.473
0.141      -3.807       0.541
housing[T.Single family]                           0.0160      0.131      0.123
0.902      -0.240       0.272
parkingtype[T.Shared parking]                     -0.1056      0.129     -0.819
0.413      -0.358       0.147
================================================================================
================================
```

In [ ]:

```
# for both MNL =2 and MNL=3 tables some variables are not signiganct
# hence i would later be fitting a regression model between those parameters that
# matter

# The values ranging from negative to positive in 0.025 and 0.975
# have no or minimum effect on our response variable

# only safety_concern[T.Neutral] and carshare_level are significant when
# compared to MNL_new = 1 for those who are taking neutral stand(MNL_new=2) in adop
# autonomous vehicles(P<0.05)

# Income[T.Low income] has odds ratio of e^-0.1716 = 0.8423 for MNL_new = 2
# and Income[T.Low income] has odds ratio of e^-0.5079 = 0.6017
# For both the values the lower income will reduce the odds for adoption
# of autonomous vehicles because the odds ratio values are less than 1.

#householdvehicles[T.Two vehicle],housing[T.Mobile based home],housing[T.Others]
#housing[T.Single family],parkingtype[T.Shared parking] are not significant
# with respect to People saying 'Yes' or MNL_new =1
```

In [ ]:

In [16]:

```
model_MNL.params
np.exp(model_MNL.params)
```

Out[16]:

	0	1
Intercept	0.236390	1.711381
safety_concern[T.Disagree]	1.235509	0.266433
safety_concern[T.Neutral]	8.805343	1.220761
carshare_level[T.Interested in participating]	2.085759	3.766129
carshare_level[T.Not participating]	1.760378	1.952310
rideshare_level[T.Interested in participating]	1.053182	1.607932
rideshare_level[T.Not participating]	0.955638	1.734321
pevowner[T.Yes]	1.260941	2.434345
income[T.Low income]	0.842348	0.601763
income[T.Middle income]	0.858758	0.639910
householdvehicles[T.Three plus]	0.761154	0.744734
householdvehicles[T.Two vehicle]	0.936783	0.900284
housing[T.Mobile based home]	0.936711	1.036154
housing[T.Others]	1.497748	0.195292
housing[T.Single family]	1.268627	1.016157
parkingtype[T.Shared parking]	1.257552	0.899802

In [17]:
```
#Marginal Effects
print(model_MNL.get_margeff().summary())
```

MNLogit Marginal Effects

```
====================================
Dep. Variable:            MNLogit_new
Method:                           dydx
At:                           overall
================================================================================
================================
                                 MNLogit_new=1      dy/dx    std err          z
P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
--------------------------------
safety_concern[T.Disagree]                          0.1533      0.029      5.204
0.000       0.096       0.211
safety_concern[T.Neutral]                          -0.2100      0.040     -5.243
0.000      -0.288      -0.131
carshare_level[T.Interested in participating]      -0.2339      0.029     -7.944
0.000      -0.292      -0.176
carshare_level[T.Not participating]                -0.1345      0.018     -7.488
0.000      -0.170      -0.099
rideshare_level[T.Interested in participating]     -0.0659      0.023     -2.855
0.004      -0.111      -0.021
rideshare_level[T.Not participating]               -0.0674      0.019     -3.512
0.000      -0.105      -0.030
pevowner[T.Yes]                                    -0.1348      0.032     -4.261
0.000      -0.197      -0.073
income[T.Low income]                                0.0803      0.026      3.114
0.002       0.030       0.131
income[T.Middle income]                             0.0707      0.022      3.260
0.001       0.028       0.113
householdvehicles[T.Three plus]                     0.0613      0.023      2.647
0.008       0.016       0.107
householdvehicles[T.Two vehicle]                    0.0191      0.018      1.038
0.299      -0.017       0.055
housing[T.Mobile based home]                        0.0009      0.060      0.016
0.987      -0.116       0.118
housing[T.Others]                                   0.1772      0.162      1.092
0.275      -0.141       0.495
housing[T.Single family]                           -0.0222      0.025     -0.892
0.372      -0.071       0.027
parkingtype[T.Shared parking]                      -0.0057      0.024     -0.236
0.814      -0.053       0.042
--------------------------------------------------------------------------------
--------------------------------
                                 MNLogit_new=2      dy/dx    std err          z
P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
--------------------------------
safety_concern[T.Disagree]                          0.1279      0.031      4.148
0.000       0.067       0.188
safety_concern[T.Neutral]                           0.3258      0.034      9.692
0.000       0.260       0.392
carshare_level[T.Interested in participating]       0.0199      0.025      0.807
0.419      -0.028       0.068
carshare_level[T.Not participating]                 0.0405      0.016      2.563
0.010       0.010       0.071
rideshare_level[T.Interested in participating]     -0.0260      0.020     -1.308
0.191      -0.065       0.013
rideshare_level[T.Not participating]               -0.0466      0.017     -2.783
0.005      -0.079      -0.014
pevowner[T.Yes]                                    -0.0275      0.027     -1.015
0.310      -0.081       0.026
income[T.Low income]                                0.0096      0.022      0.436
0.663      -0.034       0.053
income[T.Middle income]                             0.0082      0.019      0.441
```

```
0.659      -0.028       0.045
householdvehicles[T.Three plus]                    -0.0215      0.020     -1.064
0.287      -0.061       0.018
householdvehicles[T.Two vehicle]                   -0.0027      0.016     -0.173
0.863      -0.033       0.028
housing[T.Mobile based home]                       -0.0128      0.057     -0.226
0.821      -0.124       0.098
housing[T.Others]                                   0.1803      0.112      1.612
0.107      -0.039       0.399
housing[T.Single family]                            0.0361      0.021      1.690
0.091      -0.006       0.078
parkingtype[T.Shared parking]                       0.0434      0.020      2.142
0.032       0.004       0.083
-------------------------------------------------------------------------------
--------------------------------
                                  MNLogit_new=3      dy/dx    std err          z
P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------
--------------------------------
safety_concern[T.Disagree]                         -0.2812      0.024    -11.760
0.000      -0.328      -0.234
safety_concern[T.Neutral]                          -0.1158      0.032     -3.616
0.000      -0.179      -0.053
carshare_level[T.Interested in participating]       0.2140      0.025      8.491
0.000       0.165       0.263
carshare_level[T.Not participating]                 0.0940      0.018      5.343
0.000       0.060       0.129
rideshare_level[T.Interested in participating]      0.0918      0.023      4.057
0.000       0.047       0.136
rideshare_level[T.Not participating]                0.1140      0.019      5.993
0.000       0.077       0.151
pevowner[T.Yes]                                     0.1623      0.026      6.183
0.000       0.111       0.214
income[T.Low income]                               -0.0899      0.024     -3.690
0.000      -0.138      -0.042
income[T.Middle income]                            -0.0789      0.020     -3.959
0.000      -0.118      -0.040
householdvehicles[T.Three plus]                    -0.0397      0.023     -1.752
0.080      -0.084       0.005
householdvehicles[T.Two vehicle]                   -0.0164      0.018     -0.923
0.356      -0.051       0.018
housing[T.Mobile based home]                        0.0118      0.059      0.199
0.842      -0.105       0.128
housing[T.Others]                                  -0.3575      0.218     -1.637
0.102      -0.785       0.070
housing[T.Single family]                           -0.0138      0.024     -0.577
0.564      -0.061       0.033
parkingtype[T.Shared parking]                      -0.0377      0.024     -1.600
0.110      -0.084       0.008
===============================================================================
================================
```

In [18]:
```
MLE = model_MNL.llf
print("Maximum Likelihood Estimate = ", MLE)
```

```
Maximum Likelihood Estimate =  -3463.088240295422
```

In [19]:
```
p = 3
AIC_1 = - 2 * MLE + 2 * p
print("Akaike Information Criteria = ", AIC_1)
```

```
Akaike Information Criteria =  6932.176480590844
```

In [20]:
```
#c) Bayesian Information Criteria (AIC)
BIC = model_MNL.bic
```

```
print("BIC", BIC)
```

```
BIC 7188.338735667714
```

In [21]:
```
# prediction of probability
yprob = model_MNL.predict()
print(yprob)

def compute_yhat(prob):

    yhat = np.argmax(prob, axis=1)

    return yhat

ypred = compute_yhat(yprob)
ypred
```

```
[[0.47794928 0.2105058  0.31154493]
 [0.35980573 0.1877623  0.45243197]
 [0.27438853 0.46455712 0.26105434]
 ...
 [0.5383785  0.22953344 0.23208806]
 [0.66522622 0.20260384 0.13216994]
 [0.68352863 0.16554228 0.1509291 ]]
```

Out[21]:
```
array([0, 2, 1, ..., 0, 0, 0], dtype=int64)
```

In [22]:
```
from sklearn.metrics import accuracy_score
Accuracy = accuracy_score(ypred, df.MNLogit_new)
Accuracy
```

Out[22]:
```
0.06751521859435529
```

In [23]:
```
#some parameters are not significant and hence i am neglecting those
#parameters and fitting a regression for those parameters that matter
```

In [24]:
```
from statsmodels.discrete.discrete_model import MNLogit
formula = 'MNLogit_new ~ safety_concern +carshare_level+ rideshare_level+pevowner+
model_MNL = MNLogit.from_formula(formula=formula , data = df ).fit()
print(model_MNL.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.960692
         Iterations 6
                          MNLogit Regression Results
==============================================================================
Dep. Variable:            MNLogit_new   No. Observations:                 3614
Model:                        MNLogit   Df Residuals:                     3594
Method:                           MLE   Df Model:                           18
Date:                Sun, 20 Nov 2022   Pseudo R-squ.:                 0.09119
Time:                        19:51:54   Log-Likelihood:                -3471.9
converged:                       True   LL-Null:                       -3820.3
Covariance Type:            nonrobust   LLR p-value:                2.761e-136
==============================================================================
================================
                                 MNLogit_new=2       coef    std err          z
P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
--------------------------------
Intercept                                       -1.3287      0.245     -5.430
0.000      -1.808      -0.849
safety_concern[T.Disagree]                       0.2158      0.217      0.994
0.320      -0.209       0.641
safety_concern[T.Neutral]                        2.1918      0.254      8.621
0.000       1.694       2.690
carshare_level[T.Interested in participating]    0.7536      0.183      4.113
0.000       0.394       1.113
carshare_level[T.Not participating]              0.5720      0.111      5.137
0.000       0.354       0.790
rideshare_level[T.Interested in participating]   0.0429      0.138      0.311
0.756      -0.228       0.314
rideshare_level[T.Not participating]            -0.0443      0.116     -0.382
0.702      -0.271       0.183
pevowner[T.Yes]                                  0.1897      0.198      0.958
0.338      -0.198       0.578
income[T.Low income]                            -0.1129      0.149     -0.757
0.449      -0.405       0.180
income[T.Middle income]                         -0.1231      0.131     -0.940
0.347      -0.380       0.134
------------------------------------------------------------------------------
--------------------------------
                                 MNLogit_new=3       coef    std err          z
P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
--------------------------------
Intercept                                        0.3628      0.166      2.186
0.029       0.038       0.688
safety_concern[T.Disagree]                      -1.2942      0.136     -9.548
0.000      -1.560      -1.029
safety_concern[T.Neutral]                        0.2354      0.193      1.219
0.223      -0.143       0.614
carshare_level[T.Interested in participating]    1.3380      0.150      8.916
0.000       1.044       1.632
carshare_level[T.Not participating]              0.6718      0.097      6.935
0.000       0.482       0.862
rideshare_level[T.Interested in participating]   0.4641      0.123      3.779
0.000       0.223       0.705
rideshare_level[T.Not participating]             0.5528      0.103      5.363
0.000       0.351       0.755
pevowner[T.Yes]                                  0.8569      0.153      5.618
0.000       0.558       1.156
income[T.Low income]                            -0.4571      0.128     -3.562
0.000      -0.709      -0.206
income[T.Middle income]                         -0.4146      0.109     -3.788
0.000      -0.629      -0.200
```

==========

In [25]:
```
# coeff for income[T.Low income] = -0.1129 for MNL_new=2 or 'Neutral'
# odds ratio is e^-0.1129 = 0.89
# Compared to MNL=1 low income are less likely to show interest to
#use autonomous vehicles by 1-0.89 =0.11

# coef for income[T.Low income] is -0.4571 for MNL_new =3 or 'No'
# odds ratio is e^-0.4571 = 0.6333
# odds ratio decreases by 1-0.6333 = 0.366 compared to MNL=1

# This means that lower income group members who say 'No'
#show less affinity towards self autonomous by 36.6%


# coeff for Income[T.Middle income] = -0.1231
# odds ratio is e^-0.1231 = 0.8841 or 88.41%
# odds will decrease by 1-0.8841 = 0.1159 or by 11.59%
```

In [ ]:
```

```

In [ ]:
```
# coeff for income[T.Low income] = -0.1129
# odds ratio is e^-0.1129 = 0.89

# Compared to MNL=1 low income are less likely to use autonomous
# vehicles by 1-0.89 =0.11

# choice of selection of self driving autonomous vehicle(for neutral people) is 0.8
# or MNL_new = 1

# coef for income[T.Low income] is -0.4571 for MNL_new =3 or 'No'
# odds ratio is e^-0.4571 = 0.6333
# odds ratio decreases by 1-0.6333 = 0.366 compared to MNL=1
# This means that lower income group members who say 'No' show less affinity toward


# coeff for Income[T.Middle income] = -0.1231
# odds ratio is e^-0.1231 = 0.8841 or 88.41%
# odds will decrease by 1-0.8841 = 0.1159 or by 11.59%

# In general it seems like richer people are more likely to use
# self driving autonomous vehicles.

# The variables except safety_concern[T.Neutral] are all significant when
# compared with People who say 'Yes' in MNLogit_new =1 for people
# who are saying 'NO' in MNLogit_new = 2


# Among the people who are neutral with choice of autonomous vehicle the people who
# say that they are neutral with safety of auto vehicles and those
# who are interested in participating carshare_level
# and not participating are significant. This is in comparision to MNL_new=1 or 'Ye


#For Carshare_Level[T.Interested in participating],for MNL_new=2 or 'Neutral'
# the odds ratio is e^0.7536 = 2.1246 meaning that the people who are neutral with
#the adoption of self driving vehicles are 2.12 times more likely to adopt
#self driving autonomous vehicles.


#For Carshare_Level[T.Not participating] the odds ratio is
# e^0.5720 = 1.7649 hence the people who are not participating in
```

```
# car share  and have chosen neutral are 1.7649 times favarouble to use
#self driving autonomous vehicles.
```