Response coding Function

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

In [2]:
```
#Response table creation function
def response_table(X_df,y_df,index_val,label_col_val):
    #index_val='state'
    #label_col_val='class'
    Fin_DF= pd.concat([X_df,y_df], axis=1)
    k=Fin_DF.pivot_table(index=index_val,columns=label_col_val,aggfunc={label_
col_val:'count'}) #Convert to pivot table on values with count

    #Convert multi index to single index
    k.fillna(0,inplace=True)
    k.columns = k.columns.droplevel()
    k.columns.name = None
    k.reset_index(inplace=True)

    k['all_sum']=k[0]+k[1] #Sum of both classes total count

    #Calculate response value for class labele
    index_val1=index_val+'0'
    index_val2=index_val+'1'
    k[index_val1]=k[0]/k['all_sum']
    k[index_val2]=k[1]/k['all_sum']

    state_list=k[index_val]#List of different value for the feature

    #Change the feature values to row name
    k = k.set_index(index_val)
    k.index.names = [None]

    return k,state_list
```

In [3]:
```
#function to fit and transform the data
def fit_transform(df_main,feature_list,response_tbl,index_val):
    class_0=[]
    class_1=[]
    index_val1=index_val+'0'
    index_val2=index_val+'1'

    for i in df_main.index:
    #for i in range(len(df_main)):
        if(df_main[index_val][i] in feature_list.tolist()):
            class_0.append(response_tbl[index_val1][df_main[index_val][i]])
            class_1.append(response_tbl[index_val2][df_main[index_val][i]])
        else:
            class_0.append(0.5)
            class_1.append(0.5)

    index_val1=index_val+'_0'
    index_val2=index_val+'_1'

    df_main[index_val1]=class_0
    df_main[index_val2]=class_1
    return df_main
```

Testing the function with demo data

In [4]:
```
data=[['A',0],['B',1],['C',1],['A',0],['A',1],['B',1],['A',0],['A',1],['C',1],
['C',0]] #create data list
```

In [5]:
```
data_frame=pd.DataFrame(data,columns=['State','Class']) #create dataframe
```

In [6]:
```
data_frame #Show the data
```

Out[6]:

	State	Class
0	A	0
1	B	1
2	C	1
3	A	0
4	A	1
5	B	1
6	A	0
7	A	1
8	C	1
9	C	0

In [7]:
```
#Call response function to create the table on the data
n,s=response_table(data_frame['State'],data_frame['Class'],'State','Class')
```

In [8]:
```
n
```

Out[8]:

	0	1	all_sum	State0	State1
A	3.0	2.0	5.0	0.600000	0.400000
B	0.0	2.0	2.0	0.000000	1.000000
C	1.0	2.0	3.0	0.333333	0.666667

In [9]:
```
s
```

Out[9]:
```
0    A
1    B
2    C
Name: State, dtype: object
```

In [10]:
```
data_frame_state=pd.DataFrame(data_frame['State'], columns = ['State'])
```

In [11]:
```
#transform the data
tr_clean_categories=fit_transform(data_frame_state,s,n,'State')
```

In [12]:
```
#Final table after transform
tr_clean_categories
```

Out[12]:

	State	State_0	State_1
0	A	0.600000	0.400000
1	B	0.000000	1.000000
2	C	0.333333	0.666667
3	A	0.600000	0.400000
4	A	0.600000	0.400000
5	B	0.000000	1.000000
6	A	0.600000	0.400000
7	A	0.600000	0.400000
8	C	0.333333	0.666667
9	C	0.333333	0.666667

__End