In [1]:

```
#Load Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:

```
#Load Dataset
dataset = pd.read_csv('./marketing_AB.csv')
dataset.head()
```

Out[2]:

	test group	total ads	most ads day	most ads hour	converted
0	ad	130	Monday	20	False
1	ad	93	Tuesday	22	False
2	ad	21	Tuesday	18	False
3	ad	355	Tuesday	10	False
4	ad	276	Friday	14	False

In [3]:

```
#Create copy of Dataset
dataset2=dataset

#Convert Object columns to numeric
dataset2 = pd.get_dummies(dataset2)
dataset2.head()
```

Out[3]:

	total ads	most ads hour	converted	test group_ad	test group_psa	most ads day_Friday	most ads day_Monday	most ads day_Saturday	most ads day_Sunday	most ads day_Thursday	most ads day_Tuesday	most ads day_Wednesday
0	130	20	False	1	0	0	1	0	0	0	0	0
1	93	22	False	1	0	0	0	0	0	0	1	0
2	21	18	False	1	0	0	0	0	0	0	1	0
3	355	10	False	1	0	0	0	0	0	0	1	0
4	276	14	False	1	0	1	0	0	0	0	0	0

In [4]:

```
#Create x and y variables
x = dataset2.drop('converted', axis=1).to_numpy()
y = dataset2['converted'].to_numpy()

#Create x and y datasets
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x, y, test_size = 0.20, stratify=y, random_state = 100)

#Scale Data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train2 = sc.fit_transform(x_train)
x_test2 = sc.transform(x_test)
```

In [5]:

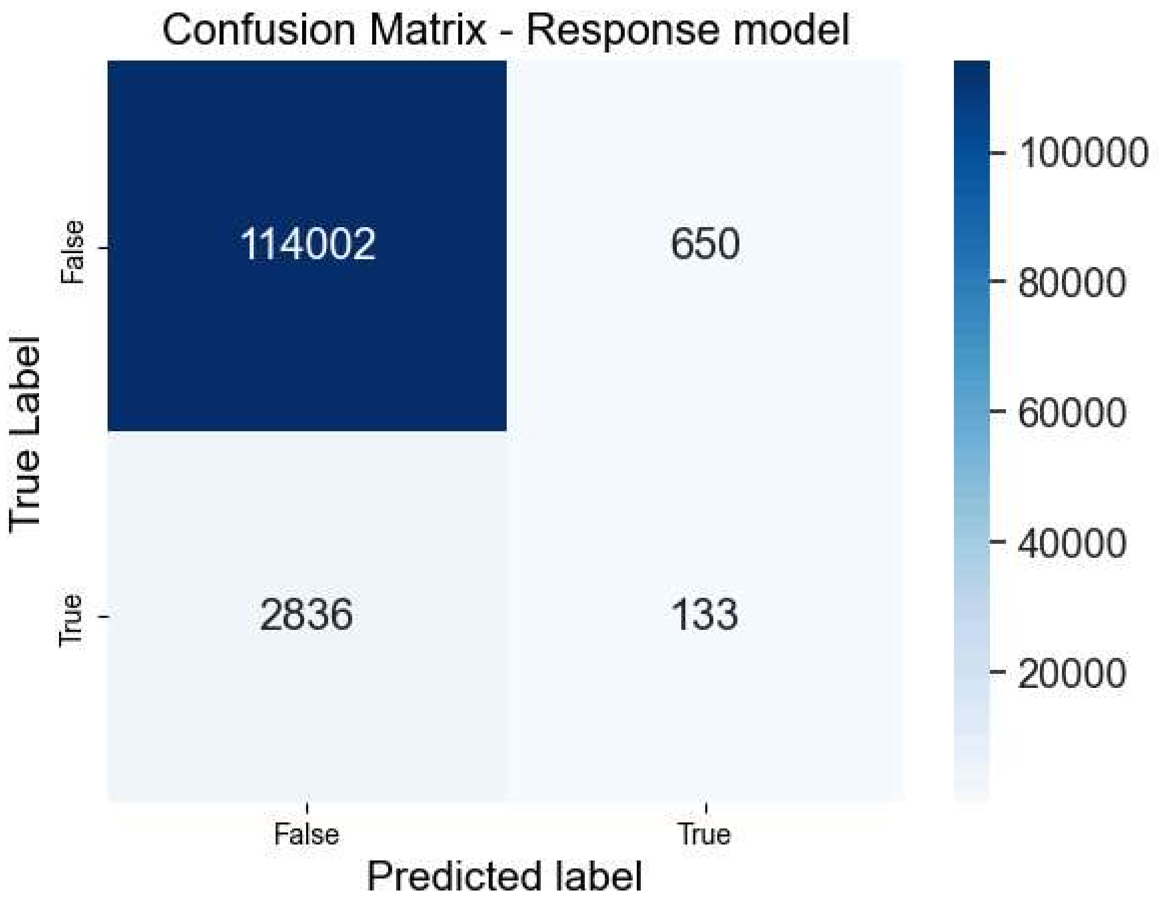
```
#Create Response Model
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.tree import DecisionTreeClassifier

for name,method in [('DT', DecisionTreeClassifier(random_state=100))]:
    method.fit(x_train2,y_train)
    predict = method.predict(x_test2)
    print('\nEstimator: {}'.format(name))
    matrix_df = confusion_matrix(y_test,predict)
    print(confusion_matrix(y_test,predict))
    print(classification_report(y_test,predict))
```

```
Estimator: DT
[[114002    650]
 [  2836    133]]
              precision    recall  f1-score   support

       False       0.98      0.99      0.98    114652
        True       0.17      0.04      0.07      2969

    accuracy                           0.97    117621
   macro avg       0.57      0.52      0.53    117621
weighted avg       0.96      0.97      0.96    117621
```

In [6]:

```
#confusion matrix
Y = dataset2['converted']
labels = Y.unique()
ax = plt.axes()
sns.set(font_scale=1.3)
plt.figure(figsize=(10,7))
sns.heatmap(matrix_df, annot=True, fmt="g", ax=ax, cmap="Blues")
#set axis titles
```

```
ax.set_title('Confusion Matrix - Response model')
ax.set_xlabel("Predicted label", fontsize =15)
ax.xaxis.set_ticklabels(['False', 'True'])
ax.set_ylabel("True Label", fontsize=15)
ax.yaxis.set_ticklabels(['False', 'True'])
plt.show()
```



```
<Figure size 1000x700 with 0 Axes>
```

```
In [ ]:
```