In [1]:

```
import numpy as np
import re
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler, OneHotEncoder
from sklearn.metrics import confusion_matrix, accuracy_score, mean_squared_error, r2_scor
e
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
import plotly.express as px
import scipy.stats as stats
import scipy.cluster.hierarchy as ch
```

In [3]:

```
data = pd.DataFrame({
    'Version': ['A'] * 1000 + ['B'] * 1000,
    'Conversion': [1] * 120 + [0] * 880 + [1] * 150 + [0] * 850,
    'Click-through': [1] * 400 + [0] * 600 + [1] * 300 + [0] * 700,
    'Bounce': [1] * 800 + [0] * 200 + [1] * 700 + [0] * 300,
    'OrderValue': [50] * 120 + [0] * 880 + [60] * 150 + [0] * 850
})
```

In [4]:

```
data
```

Out[4]:

	Version	Conversion	Click-through	Bounce	OrderValue
0	A	1	1	1	50
1	A	1	1	1	50
2	A	1	1	1	50
3	A	1	1	1	50
4	A	1	1	1	50
...	...	...	...	...	...
1995	B	0	0	0	0
1996	B	0	0	0	0
1997	B	0	0	0	0
1998	B	0	0	0	0
1999	B	0	0	0	0

2000 rows × 5 columns

Null Hypothesis: There is no significance difference between the conversion rate of version A and B

Alternative Hypothesis: There is significance difference between the conversion rate of version A and B

In [11]:

```
# conversion rate refers to the percentage of users who have completed a desired action.
# click-through refers to the percentage of people visiting a web page from an advertisem
```

```
ent or organic search result.
# Bounce rate measures the number of users who enter a website and exit without visiting
any other page on the website.
```

In [6]:

```
data.groupby('Version')['Conversion'].mean()*100
```

Out[6]:

```
Version
A    12.0
B    15.0
Name: Conversion, dtype: float64
```

In [12]:

```
data.groupby('Version')['Click-through'].mean()*100
```

Out[12]:

```
Version
A    40.0
B    30.0
Name: Click-through, dtype: float64
```

In [13]:

```
data.groupby('Version')['Bounce'].mean()*100
```

Out[13]:

```
Version
A    80.0
B    70.0
Name: Bounce, dtype: float64
```

In [8]:

```
data.groupby('Version')['OrderValue'].sum()
```

Out[8]:

```
Version
A    6000
B    9000
Name: OrderValue, dtype: int64
```

In [10]:

```
data[data['Conversion'] == 1].groupby('Version')['OrderValue'].mean()
```

Out[10]:

```
Version
A    50.0
B    60.0
Name: OrderValue, dtype: float64
```

Statistical Test of Conversion Rates

In [31]:

```
version_A = data[data['Version'] == 'A']['Conversion']
version_B = data[data['Version'] == 'B']['Conversion']

t_stats, p_value = stats.ttest_ind(version_A, version_B)
print("T-Statistic:", t_stats)
print("P-Value:", p_value)

if p_value > 0.05:
    print("Null Hypothesis accepted")
```

```
else:
    print("Null Hypothesis rejected")
    print("Hence, we can conclude that there is a significance difference in conversion r
ates between the 2 homepage versions")
```

```
T-Statistic: -1.9639610121239313
P-Value: 0.04967307061620513
Null Hypothesis rejected
Hence, we can conclude that there is a significance difference in conversion rates betwee
n the 2 homepage versions
```

In [17]:

```
def confidence_interval(data):
    mean = np.mean(data)
    std = np.std(data)
    num = len(data)
    std_error = std/np.sqrt(num)
    interval = stats.t.interval(0.95, num-1, loc = mean, scale = std_error)
    return interval
```

In [18]:

```
interval_A = confidence_interval(version_A)
print("Confidence interval for conversion rates of Version A", interval_A)
```

```
Confidence interval for conversion rates of Version A (0.09983461402638706, 0.14016538597
361294)
```

In [20]:

```
interval_B = confidence_interval(version_B)
print("Confidence interval for conversion rates of Version B", interval_B)
```

```
Confidence interval for conversion rates of Version B (0.12784204519172387, 0.17215795480
827611)
```

Statistical Test of Click-through

In [37]:

```
ct_version_A = data[data['Version'] == 'A']['Click-through']
ct_version_B = data[data['Version'] == 'B']['Click-through']

t_stats, p_value = stats.ttest_ind(ct_version_A, ct_version_B)
print("T-Statistic:", t_stats)
print("P-Value:", p_value)

if p_value > 0.05:
    print("Null Hypothesis accepted")
else:
    print("Null Hypothesis rejected")
    print("Hence, we can conclude that there is a significance difference in click-throug
h rates between the 2 homepage versions")
```

```
T-Statistic: 4.7116875957559
P-Value: 2.6260518162186423e-06
Null Hypothesis rejected
Hence, we can conclude that there is a significance difference in click-through rates bet
ween the 2 homepage versions
```

In [28]:

```
interval_A = confidence_interval(ct_version_A)
print("Confidence interval for click-through of Version A", interval_A)
```

```
Confidence interval for click-through of Version A (0.3695995368058314, 0.430400463194168
64)
```

In [30]:

```
ct_interval_B = confidence_interval(ct_version_B)
print("Confidence interval for click-through of Version B", ct_interval_B)
```

```
Confidence interval for click-through of Version B (0.27156297058204715, 0.32843702941795
283)
```

Statistical Test of Bounce

In [36]:

```
b_version_A = data[data['Version'] == 'A']['Bounce']
b_version_B = data[data['Version'] == 'B']['Bounce']

t_stats, p_value = stats.ttest_ind(b_version_A, b_version_B)
print("T-Statistic:", t_stats)
print("P-Value:", p_value)

if p_value > 0.05:
    print("Null Hypothesis accepted")
else:
    print("Null Hypothesis rejected")
    print("Hence, we can conclude that there is a significance difference in bounce rates
between the 2 homepage versions")
```

```
T-Statistic: 5.196152422706636
P-Value: 2.2418354097918818e-07
Null Hypothesis rejected
Hence, we can conclude that there is a significance difference in bounce rates between th
e 2 homepage versions
```

In [34]:

```
interval_A = confidence_interval(b_version_A)
print("Confidence interval for bounce of Version A", interval_A)
```

```
Confidence interval for bounce of Version A (0.7751781257433422, 0.8248218742566579)
```

In [35]:

```
interval_B = confidence_interval(b_version_B)
print("Confidence interval bounce of Version B", interval_B)
```

```
Confidence interval bounce of Version B (0.6715629705820472, 0.7284370294179527)
```

From the above tests based on relevant metrices we can conclude that there is a significance difference in all rates between the 2 homepage version

From the above observations we conclude that conversion rate of version B is higher than version A, hence version B is recommended to the company