CS 203
Assignment 9
Team 20

Chinmay Pendse (23110245)
Dakshata Bhamare (23210027)

Introduction

GitHub Link: https://github.com/chinmayp995/CS203-Assignment-10

This assignment aims to apply A/B testing using hypothesis testing on ad-click data and to detect covariate shifts in air quality data using statistical tests.

We learned how to:

- Use proportions_ztest to compare click-through rates (CTR).
- Apply the Kolmogorov–Smirnov (KS) test to identify distributional differences.

Part 1: A/B Testing using Ad Click Prediction

We used a Kaggle ad-click dataset; the first five rows of the dataset look like these. Then we Performed necessary data cleaning and preprocessing, like drop missing rows and converting numerical columns into categorical columns. Then, split the dataset into two groups:

- **Group A**: Users who saw ads at the top (ad_position = 0)
- **Group B**: Users who saw ads at the bottom (ad_position = 1)

```python
# converting to categorical
df['gender'] = df['gender'].astype('category')
df['ad_position'] = df['ad_position'].astype('category')

# map gender categories
df['gender'] = df['gender'].map({'Male': 0, 'Female': 1, 'Non-Binary': 2})

# convert to category again and add -1 to handle NaN values
df['gender'] = df['gender'].astype('category')
df['gender'] = df['gender'].cat.add_categories([-1])
df['gender'] = df['gender'].fillna(-1)

# mapping ad_position categories
df['ad_position'] = df['ad_position'].map({'Top':0,'Bottom':1,'Side':2})
```

```python
# splitting into group A (Top) and group B (bottom)
group_a = df[df['ad_position'] == 0]
group_b = df[df['ad_position'] == 1]
```

```python
group_a.head()
```

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click
0	670	User670	22.0	-1	Desktop	0	Shopping	Afternoon	1
1	3044	User3044	NaN	0	Desktop	0	NaN	NaN	1
6	7808	User7808	26.0	1	Desktop	0	NaN	NaN	1
15	7529	User7529	NaN	-1	NaN	0	Entertainment	Afternoon	0
18	2124	User2124	NaN	0	Desktop	0	NaN	Evening	1

```python
group_b.head()
```

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click
5	5942	User5942	NaN	2	NaN	1	Social Media	Evening	1
8	7993	User7993	NaN	2	Mobile	1	Social Media	NaN	1
9	4509	User4509	NaN	-1	NaN	1	Education	Afternoon	1
10	2595	User2595	NaN	-1	NaN	1	NaN	Morning	1
11	7466	User7466	47.0	-1	Mobile	1	NaN	Afternoon	1

- Used the statsmodel's proportions_ztest function to perform an independent two-sample z-test between Group A and Group B.

```
In [18]:
# CTRs :Click through rate
ctr_A = clicks_A / n_A
ctr_B = clicks_B / n_B

print(f"CTR (Top ads): {ctr_A:.4f}")
print(f"CTR (Bottom ads): {ctr_B:.4f}")

CTR (Top ads): 0.6350
CTR (Bottom ads): 0.6873
```

```
In [19]:
# performing z-test
z_score, p_value = proportions_ztest([clicks_A, clicks_B], [n_A, n_B])
print(f"Z_score: {z_score:.4f}")
print(f"P_value: {p_value:.4f}")

Z_score: -4.0642
P_value: 0.0000
```

The z-test resulted in a z-score of -4.0642 and a p-value of 0.. Since the p-value is less than 0.05, we reject the null hypothesis. This indicates a statistically significant difference in click-through rates between ads placed at the top and those at the bottom. The negative z-score also suggests that bottom-position ads had a higher click-through rate in this dataset.

Part 2: Covariate Shift Detection Using Air Quality Data

Loading all three datasets using Pandas is shown in the screenshot.

```
# loading air quality datasets
train = pd.read_csv("train.csv")
test1 = pd.read_csv("test1.csv")
test2 = pd.read_csv("test2.csv")
```

- Performing the **Kolmogorov–Smirnov test** for this, we are using **from Scipy.stats import ks_2samp**

In [26]:

```
# just preview
print("\nTrain NO2(GT):", train_no2.describe())
print("Test1 NO2(GT):", test1_no2.describe())
print("Test2 NO2(GT):", test2_no2.describe())
```

```
Train NO2(GT): count    3200.000000
mean       45.605625
std       114.663990
min      -200.000000
25%        47.750000
50%        84.000000
75%       114.000000
max       233.000000
Name: NO2(GT), dtype: float64
Test1 NO2(GT): count    800.000000
mean      42.621250
std      117.115831
min     -200.000000
25%       46.750000
50%       84.000000
75%      114.000000
max      223.000000
Name: NO2(GT), dtype: float64
Test2 NO2(GT): count    800.000000
mean     129.682500
std       61.071957
min     -200.000000
25%      100.000000
50%      133.000000
75%      163.250000
max      248.000000
Name: NO2(GT), dtype: float64
```

We removed the places where the value of the NO2(GT) column is negative

```
train_no2 = train['NO2(GT)']
test1_no2 = test1['NO2(GT)']
test2_no2 = test2['NO2(GT)']

 #removing the places where the value of NO2(GT) column is negative

train_no2_new = train_no2[(train_no2 >= 0)].dropna()
test1_no2_new = test1_no2[(test1_no2 >= 0)].dropna()
test2_no2_new = test2_no2[(test2_no2 >= 0)].dropna()
```

In [31]:
```python
# just preview
print("\nTrain NO2(GT):", train_no2_new.describe())
print("Test1 NO2(GT):", test1_no2_new.describe())
print("Test2 NO2(GT):", test2_no2_new.describe())
```

```
Train NO2(GT): count    2668.000000
mean        94.579460
std         36.584146
min          5.000000
25%         67.000000
50%         94.000000
75%        120.000000
max        233.000000
Name: NO2(GT), dtype: float64
Test1 NO2(GT): count    659.000000
mean       94.532625
std        36.639541
min         5.000000
25%        66.000000
50%        94.000000
75%       118.000000
max       223.000000
Name: NO2(GT), dtype: float64
Test2 NO2(GT): count    788.000000
mean      134.703046
std        45.870772
min        25.000000
25%       101.000000
50%       134.000000
75%       164.000000
max       248.000000
Name: NO2(GT), dtype: float64
```

In [32]:
```python
# comparing train vs test1
ks_stat1, p_val1 = ks_2samp(train_no2_new, test1_no2_new)
print(f"Test1 vs. Train: KS Stat = {ks_stat1:.4f}, P-Value = {p_val1:.4f}")


# comparing train vs test2
ks_stat2, p_val2 = ks_2samp(train_no2_new, test2_no2_new)
print(f"Test2 vs. Train: KS Stat = {ks_stat2:.4f}, P-Value = {p_val2:.4f}")
```

```
Test1 vs. Train: KS Stat = 0.0171, P-Value = 0.9971
Test2 vs. Train: KS Stat = 0.3689, P-Value = 0.0000
```

Using the Kolmogorov–Smirnov test on the NO2(GT) column, we compared test1.csv and test2.csv with train.csv. The test2 dataset showed a higher KS statistic and a lower p-value than test1, indicating a greater distributional difference. Therefore, **test2.csv exhibits a more significant covariate shift** than the training data.

In [33]:
```
alpha = 0.05
shift_test1 = p_val1 < alpha  # True if shift detected
shift_test2 = p_val2 < alpha

print(f"Covariate Shift in Test1: {shift_test1}")
print(f"Covariate Shift in Test2: {shift_test2}")
```

```
Covariate Shift in Test1: False
Covariate Shift in Test2: True
```

In [26]:
```
# Covariance shift

if ks_stat1 > ks_stat2:
    print("Test1 shows a larger covariate shift from training data.")
else:
    print("Test2 shows a larger covariate shift from training data.")
```

```
Test2 shows a larger covariate shift from training data.
```

This assignment demonstrated the practical application of statistical hypothesis testing and covariate shift detection using real-world datasets. We successfully used A/B testing to evaluate ad effectiveness. We applied the KS test to uncover distributional changes in air quality data, reinforcing key data science concepts used in AI model validation.

End of assignment. Thank you.