**GROUP 28**

LAKSH JAIN – 23110185

TANISH YELGOE – 23110328

GUTHUB LINK: [Jain-Laksh/CS203-Lab-10]

# CS203: LAB 10

## Part 1: A/B Testing using Ad Click Prediction

### 1] Load the dataset into a pandas DataFrame

```python
ad_data = pd.read_csv("ad_click_dataset.csv")
display(ad_data)
```

Python

| | # id | full_name | # age | gender | device_type | ad_position | browsing_history |
|---|---|---|---|---|---|---|---|
| 0 | 670 | User670 | 22.0 | Missing value | Desktop | Top | Shopping |
| 1 | 3044 | User3044 | Missing value | Male | Desktop | Top | Missing value |
| 2 | 5912 | User5912 | 41.0 | Non-Binary | Missing value | Side | Education |
| 3 | 5418 | User5418 | 34.0 | Male | Missing value | Missing value | Entertainment |
| 4 | 9452 | User9452 | 39.0 | Non-Binary | Missing value | Missing value | Social Media |
| 5 | 5942 | User5942 | Missing value | Non-Binary | Missing value | Bottom | Social Media |
| 6 | 7808 | User7808 | 26.0 | Female | Desktop | Top | Missing value |
| 7 | 5065 | User5065 | 40.0 | Male | Mobile | Side | Missing value |
| 8 | 7993 | User7993 | Missing value | Non-Binary | Mobile | Bottom | Social Media |
| 9 | 4509 | User4509 | Missing value | Missing value | Missing value | Bottom | Education |

10,000 rows x 9 cols 10 per page Page 1 of 1000

### 2] Perform necessary data cleaning and preprocessing: **[10 points]**

### A] Handle missing values

```python
# Handle missing values in ad position by removing them
ad_data_clean = ad_data.dropna(subset=['ad_position'])
print(ad_data_clean.isnull().sum())
```

✓ 0.0s

```
id                  0
full_name           0
age              3814
gender           3779
device_type      1567
ad_position         0
browsing_history 3773
time_of_day      1592
click               0
dtype: int64
```

### B] Convert categorical columns (e.g., gender, ad_position)

```python
# Convert categorical columns to numerical values
ad_data_clean["gender"] = ad_data_clean["gender"].astype('category').cat.codes
ad_data_clean["device_type"] = ad_data_clean["device_type"].astype('category').cat.codes
ad_data_clean["browsing_history"] = ad_data_clean["browsing_history"].astype('category').cat.codes
ad_data_clean["time_of_day"] = ad_data_clean["time_of_day"].astype('category').cat.codes
ad_data_clean["click"] = ad_data_clean["click"].astype(int)

ad_data_clean = ad_data_clean[ad_data_clean['ad_position'].isin(['Top', 'Bottom'])].copy()
ad_data_clean['ad_position'] = ad_data_clean['ad_position'].map({'Top': 0, 'Bottom': 1})

ad_data_clean
```

| gender | device_type | ad_position | browsing_history | time_of_day | click |
|---|---|---|---|---|---|
| -1 | 0 | 0 | 3 | 0 | |
| 1 | 0 | 0 | -1 | -1 | |
| 2 | -1 | 1 | 4 | 1 | |
| 0 | 0 | 0 | -1 | -1 | |
| 2 | 1 | 1 | 4 | -1 | |
| -1 | -1 | 1 | 0 | 0 | |
| -1 | -1 | 1 | -1 | 2 | |
| -1 | 1 | 1 | -1 | 0 | |
| -1 | -1 | 0 | 1 | 0 | |
| 1 | 2 | 1 | -1 | -1 | |

5,414 rows x 9 cols 10 per page Page 1 of 542

3] Split the dataset into two groups: **[10 points]**

```
# Divide into Group A and Group B
group_a = ad_data_clean[ad_data_clean['ad_position'] == 0].copy() # Users with ad_position = 0 (Top)
group_b = ad_data_clean[ad_data_clean['ad_position'] == 1].copy() # Users with ad_position = 1 (Bottom)
```

```
GROUP A
Number of samples:  2597
```

| | id | full_name | age | gender | device_type | ad_position | browsing_history |
|---|---|---|---|---|---|---|---|
| 0 | 670 | User670 | 22.0 | -1 | 0 | 0 | |
| 1 | 3044 | User3044 | *Missing value* | 1 | 0 | 0 | |
| 6 | 7808 | User7808 | 26.0 | 0 | 0 | 0 | |
| 15 | 7529 | User7529 | *Missing value* | -1 | -1 | 0 | |
| 18 | 2124 | User2124 | *Missing value* | 1 | 0 | 0 | |

5 rows x 9 cols 10 per page Page 1 of 1

```
GROUP B
Number of samples:  2817
```

| | id | full_name | age | gender | device_type | ad_position | browsing_history |
|---|---|---|---|---|---|---|---|
| 5 | 5942 | User5942 | *Missing value* | 2 | -1 | 1 | |
| 8 | 7993 | User7993 | *Missing value* | 2 | 1 | 1 | |
| 9 | 4509 | User4509 | *Missing value* | -1 | -1 | 1 | |
| 10 | 2595 | User2595 | *Missing value* | -1 | -1 | 1 | |
| 11 | 7466 | User7466 | 47.0 | -1 | 1 | 1 | |

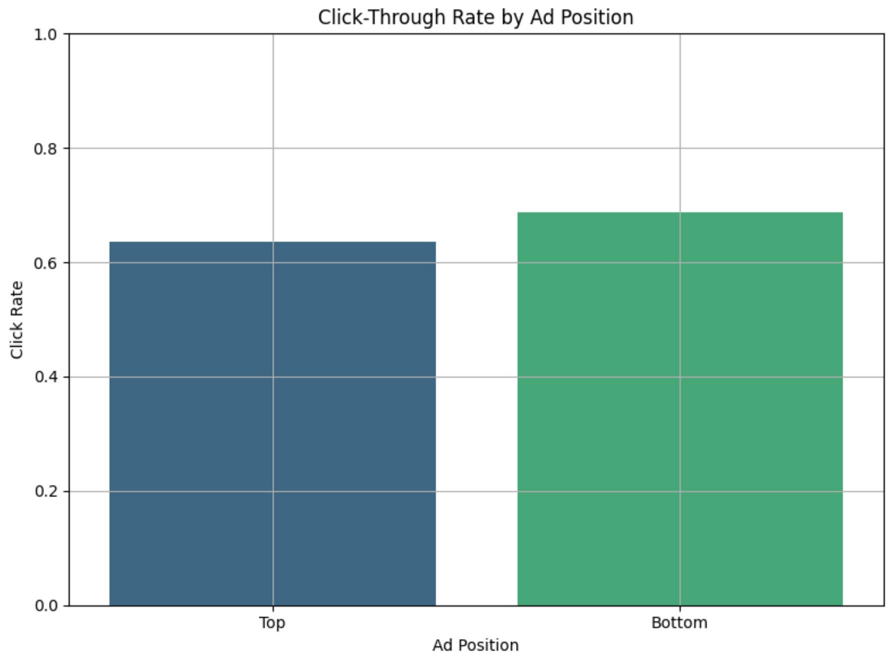
5 rows x 9 cols 10 per page Page 1 of 1

4] Use the scipy's stats.proportions_ztest function to perform an independent two-sample z-test between Group A and Group B.

```
from statsmodels.stats.proportion import proportions_ztest as ztest

clicks = [group_a['click'].sum(), group_b['click'].sum()]
n_samples = [group_a.shape[0], group_b.shape[0]]
z_stat, p_value = ztest(clicks, n_samples)
print(f"Z-score: {z_stat:.4f}\nP-value: {p_value}")
```

5] Print the z-score and the p-value

```
Z-score: -4.0642
P-value: 4.819430188759425e-05
```




6] Interpretations

**DEFINITIONS**

*z-score*: A z-score tells you how many standard deviations away your observed difference is from zero (i.e., no difference between the two groups). For a difference to be considered statistically significant, we usually look for a z-score beyond ±1.96 (for 95% confidence).

*p-value*: Represents the probability that the observed difference in click-through rates happened by random chance, assuming there's no real difference between the groups. To reject the null hypothesis, the value must be less than 0.05.

**INTERPRETATION**:

Based on the z-test, the z-score is -4.0642 and the p-value is 0.00004. Since the p-value is lesser than the standard significance level of 0.05, we can reject the null hypothesis. This means there is statistically significant difference in click-through rates between users who saw the ad at the top and those who saw it at the bottom. The observed click-through rate of Group A (top ad) is lower than Group B (bottom ad), by about 4 standard deviations. Also, the z-score is beyond ±1.96 (>95% confidence).

**Part 2: Covariate Shift Detection Using Air Quality Data**

1] Load all three datasets using pandas. **[10 points]**

```
train = pd.read_csv("Air_Quality_Dataset/train.csv")
test1 = pd.read_csv("Air_Quality_Dataset/test1.csv")
test2 = pd.read_csv("Air_Quality_Dataset/test2.csv")

display(train)
```

```
# Data Pre-processing

train = train.drop(['Unnamed: 15','Unnamed: 16'], axis=1)
test1 = test1.drop(['Unnamed: 15','Unnamed: 16'], axis=1)
test2 = test2.drop(['Unnamed: 15','Unnamed: 16'], axis=1)

train
```

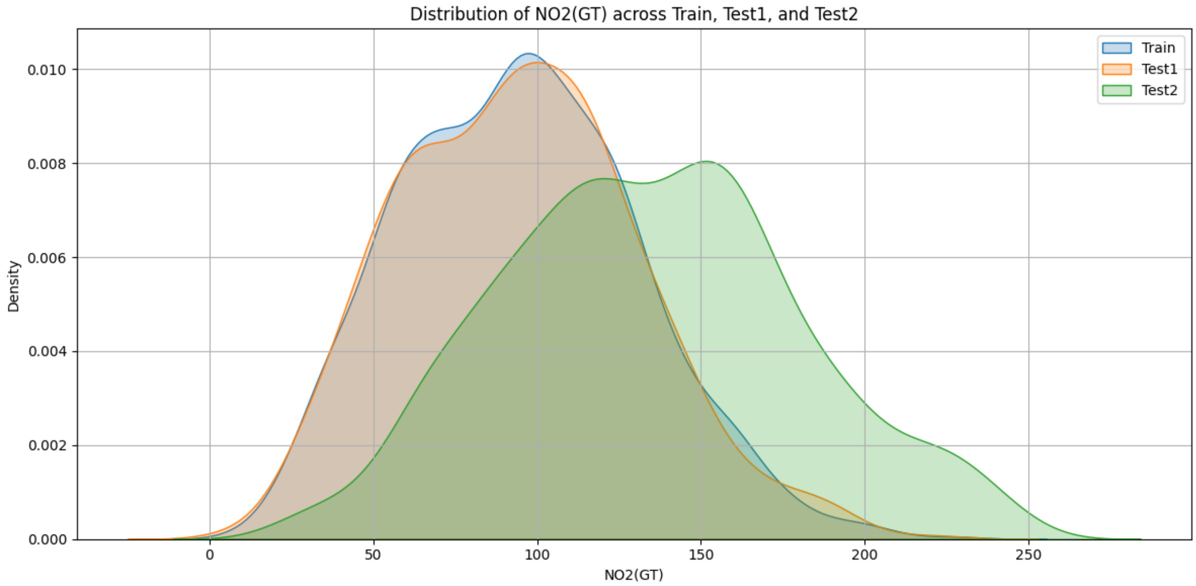
2] For each test dataset (test1.csv and test2.csv), compare it with train.csv using the **Kolmogorov–Smirnov test** (scipy.stats.ks_2samp). Perform the KS test on the **NO2(GT)** column to identify whether there are any distributional differences. **[20 points]**

```
from scipy.stats import ks_2samp

# Remove rows with missing values in the 'NO2(GT)' column
train_no2 = train['NO2(GT)'].dropna()
test1_no2 = test1['NO2(GT)'].dropna()
test2_no2 = test2['NO2(GT)'].dropna()

# Remove negative values from the 'NO2(GT)' column
train_no2 = train_no2[train_no2 >= 0]
test1_no2 = test1_no2[test1_no2 >= 0]
test2_no2 = test2_no2[test2_no2 >= 0]

# KS test between train and test1
ks_stat_test1, p_value_test1 = ks_2samp(train_no2, test1_no2)

# KS test between train and test2
ks_stat_test2, p_value_test2 = ks_2samp(train_no2, test2_no2)
```

3] Report the KS statistic and p-value for each feature. **[10 points]**

```
KS Test: Train vs Test1
KS Statistic: 0.0171
P-value: 0.9971

KS Test: Train vs Test2
KS Statistic: 0.3689
P-value: 2.53172387531317e-74
```

4] Determine which of the two test datasets (test1.csv or test2.csv) exhibits a covariate shift relative to the training dataset (train.csv). Use the results of the Kolmogorov–Smirnov test to support your answer. **[10 points]**




**DEFINITIONS**

1] KS (Kolmogorov-Smirnov) statistic: Measures the maximum difference between the cumulative distributions of two datasets. The higher the KS score, the greater the difference between the two distributions. A low KS score means the distributions are very similar.

2] P-value: Measures the probability of observing the data, assuming the null hypothesis (that the two distributions are the same) is true. A low p-value (< 0.05) suggests that the null hypothesis can be rejected, meaning the two distributions are likely different. A high p-value (> 0.05) indicates that we fail to reject the null hypothesis, suggesting no significant difference between the distributions.

**INTERPRETATION**

1] Train vs Test1

The numbers indicate that the distributions of NO2(GT) for the training set and test1 are nearly identical. The very high p-value suggests that we fail to reject the null hypothesis, meaning there is no statistically significant difference between these two distributions.

2] Train vs Test2

The values show a significant difference between the distributions. The high KS statistic and extremely low p-value lead us to reject the null hypothesis, indicating that the distribution of NO2(GT) in test2 is notably different from that in the training set.

**CONCLUSION**

Test2 exhibits a covariate shift relative to the training dataset, as its distribution for NO2(GT) is statistically significantly different from the training set (p-value = 0.0000). In contrast, Test1 does not exhibit a covariate shift (p-value = 0.9971).