STTAI - Lab Assignment 10

Name	Roll Number
Romit Mohane	23110279
Rudra Pratap Singh	23110281

Using real-world data, this assignment will introduces us to key concepts in A/B testing and Covariate Shift Detection. We performed hypothesis testing using the scipy library and identified distributional shifts in datasets using classification-based techniques.

Part 1: A/B Testing using Ad Click Prediction

1. **Load the Dataset into a pandas df**

```
df = pd.read_csv('ad_click_dataset.csv')
df
```

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click
0	670	User670	22.0	NaN	Desktop	Top	Shopping	Afternoon	1
1	3044	User3044	NaN	Male	Desktop	Top	NaN	NaN	1
2	5912	User5912	41.0	Non-Binary	NaN	Side	Education	Night	1
3	5418	User5418	34.0	Male	NaN	NaN	Entertainment	Evening	1
4	9452	User9452	39.0	Non-Binary	NaN	NaN	Social Media	Morning	0
...	...	...	...	...	...	...	...	...	...
9995	8510	User8510	NaN	NaN	Mobile	Top	Education	NaN	0
9996	7843	User7843	NaN	Female	Desktop	Bottom	Entertainment	NaN	0
9997	3914	User3914	NaN	Male	Mobile	Side	NaN	Morning	0
9998	7924	User7924	NaN	NaN	Desktop	NaN	Shopping	Morning	1
9999	3056	User3056	44.0	Male	Tablet	Top	Social Media	Morning	0

10000 rows × 9 columns

2. **Perform necessary data cleaning and preprocessing:**

```
df_nullhandled_dropped
```

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click
17	188	User188	56.0	Female	Tablet	Bottom	News	Morning	1
25	4890	User4890	43.0	Male	Tablet	Bottom	Education	Afternoon	1
33	4985	User4985	37.0	Male	Mobile	Top	News	Evening	0
52	9888	User9888	49.0	Male	Mobile	Top	News	Morning	1
102	8201	User8201	59.0	Female	Desktop	Bottom	Social Media	Morning	0
...	...	...	...	...	...	...	...	...	...
9951	7268	User7268	28.0	Female	Desktop	Bottom	News	Evening	1
9952	5912	User5912	41.0	Non-Binary	Mobile	Side	Education	Night	1
9960	9638	User9638	64.0	Non-Binary	Desktop	Top	Entertainment	Morning	0
9986	5574	User5574	52.0	Female	Desktop	Bottom	Shopping	Afternoon	1
9999	3056	User3056	44.0	Male	Tablet	Top	Social Media	Morning	0

816 rows × 9 columns

```
# Work on the dropped values DF
df_current = df_nullhandled_dropped.copy()
# Keep only the two positions of interest
df_current = df_current[df_current['ad_position'].isin(['Top','Bottom'])]

# Map to numbers
df_current['ad_position_flag'] = df_current['ad_position'].map({'Top': 0, 'Bottom': 1})
df_current['gender'] = df_current['gender'].map({'Male': 0, 'Female': 1, 'Non-Binary': 2})
df_current['time_of_day'] = df_current['time_of_day'].map({'Morning': 0, 'Afternoon': 1, 'Evening': 2, 'Night': 3})
df_current['device_type'] = df_current['device_type'].map({'Desktop': 0, 'Mobile': 1, 'Tablet': 2})
df_current['browsing_history'] = df_current['browsing_history'].map({'Shopping': 0, 'Social Media': 1, 'Education': 2, 'News': 3, 'Entertainment': 4})
```

```
df_current # df with dropped rows where null values were present
```

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click	ad_position_flag
17	188	User188	56.0	1	2	Bottom	3	0	1	1
25	4890	User4890	43.0	0	2	Bottom	2	1	1	1
33	4985	User4985	37.0	0	1	Top	3	2	0	0
52	9888	User9888	49.0	0	1	Top	3	0	1	0
102	8201	User8201	59.0	1	0	Bottom	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...
9928	7790	User7790	43.0	2	1	Top	1	0	0	0
9951	7268	User7268	28.0	1	0	Bottom	3	2	1	1
9960	9638	User9638	64.0	2	0	Top	4	0	0	0
9986	5574	User5574	52.0	1	0	Bottom	0	1	1	1
9999	3056	User3056	44.0	0	2	Top	1	0	0	0

558 rows × 10 columns

3. **Split the dataset**

```
groupA = df_current[df_current['ad_position_flag']==0]

# Group B: Bottom (flag=1)
groupB = df_current[df_current['ad_position_flag']==1]
```

```
groupA
```

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click	ad_position_flag
33	4985	User4985	37.0	Male	Mobile	Top	News	Evening	0	0
52	9888	User9888	49.0	Male	Mobile	Top	News	Morning	1	0
158	3007	User3007	42.0	Male	Desktop	Top	Shopping	Night	0	0
204	8530	User8530	52.0	Female	Mobile	Top	Social Media	Afternoon	1	0
231	4625	User4625	33.0	Non-Binary	Mobile	Top	News	Morning	0	0
...	...	...	...	...	...	...	...	...	...	...
9888	5055	User5055	45.0	Male	Desktop	Top	Education	Morning	1	0
9915	3335	User3335	24.0	Male	Mobile	Top	Entertainment	Night	1	0
9928	7790	User7790	43.0	Non-Binary	Mobile	Top	Social Media	Morning	0	0
9960	9638	User9638	64.0	Non-Binary	Desktop	Top	Entertainment	Morning	0	0
9999	3056	User3056	44.0	Male	Tablet	Top	Social Media	Morning	0	0

275 rows × 10 columns

```
groupB
```

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click	ad_position_flag
17	188	User188	56.0	Female	Tablet	Bottom	News	Morning	1	1
25	4890	User4890	43.0	Male	Tablet	Bottom	Education	Afternoon	1	1
102	8201	User8201	59.0	Female	Desktop	Bottom	Social Media	Morning	0	1

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

```python
    clicks = [ groupA['click'].sum(),
               groupB['click'].sum() ]

    # number of users in each group
    nobs   = [ len(groupA),
               len(groupB) ]
    print(clicks[0]/nobs[0], clicks[1]/nobs[1], "\n")


    z_score, p_value = proportions_ztest(count=clicks, nobs=nobs)
    print("z-score:", z_score)
    print("p-value:", p_value)

    if p_value < 0.05:
        print("Reject null hypothesis: the two CTRs are not equal.")
    else:
        print("Fail to reject null hypothesis: the two CTRs are equal.")
```

```
0.6327272727272727 0.6784452296819788

z-score: -1.1365075404030447
p-value: 0.2557442115851094
Fail to reject null hypothesis: the two CTRs are equal.
```

4. **Interpret the result: Is there a statistically significant difference in click-through rates between the two groups? Justify your answer.** In our A/B test on 10,000 users (with missing values dropped), we compared click-through rates (CTRs) for ads shown at the Top vs. Bottom positions. Using a two-sample z-test, we obtained `z = -1.137` and `p = 0.256` (> 0.05), so we accept H_0 that the two CTRs are equal. The negative z-score indicates Bottom-positioned ads achieved a lower CTR than Top-positioned ads.

   But, this difference is statistically insignificant.

Part 2: Covariate Shift Detection Using Air Quality Data

1. **Load the datasets**

```python
import pandas as pd

air_df_train = pd.read_csv('train.csv')
air_df_test1 = pd.read_csv('test1.csv')
air_df_test2 = pd.read_csv('test2.csv')

air_df_train = air_df_train[air_df_train['NO2(GT)'] >= 0]
air_df_test1 = air_df_test1[air_df_test1['NO2(GT)'] >= 0]
air_df_test2 = air_df_test2[air_df_test2['NO2(GT)'] >= 0]

air_df_train.head()
```

	Unnamed: 0	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH	Unnamed: 15	Unnamed: 16
1	2533	24/06/2004	07.00.00	1,2	1030.0	-200.0	6,9	851.0	102.0	824.0	68.0	1700.0	983.0	21,9	57,0	1,4742	NaN	NaN
2	3047	15/07/2004	17.00.00	3,2	1164.0	-200.0	20,3	1306.0	259.0	648.0	198.0	1886.0	1218.0	35,5	19,1	1,0888	NaN	NaN
3	805	13/04/2004	07.00.00	3,9	1496.0	524.0	19,1	1272.0	328.0	667.0	130.0	2011.0	1399.0	11,0	64,2	0,8398	NaN	NaN
4	2962	12/07/2004	04.00.00	-200	780.0	-200.0	1,8	568.0	24.0	1200.0	34.0	1331.0	501.0	19,9	51,3	1,1803	NaN	NaN
7	1910	29/05/2004	08.00.00	0,6	843.0	-200.0	4,0	712.0	77.0	1303.0	77.0	1343.0	548.0	18,9	42,1	0,9081	NaN	NaN

```python
air_df_train.columns
```

```
Index(['Unnamed: 0', 'Date', 'Time', 'CO(GT)', 'PT08.S1(CO)', 'NMHC(GT)',
       'C6H6(GT)', 'PT08.S2(NMHC)', 'NOx(GT)', 'PT08.S3(NOx)', 'NO2(GT)',
       'PT08.S4(NO2)', 'PT08.S5(O3)', 'T', 'RH', 'AH', 'Unnamed: 15',
       'Unnamed: 16'],
      dtype='object')
```

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**

```python
ks_statistic_test1, p_value_test1 = ks_2samp(air_df_train['NO2(GT)'], air_df_test1['NO2(GT)'])
print("\nMean for test set 1:", air_df_test1['NO2(GT)'].mean())
print("KS Test for test1.csv:")
print("KS Statistic: {ks_statistic_test1}")
print("P-value: {p_value_test1}")

# Perform KS test for test2.csv
ks_statistic_test2, p_value_test2 = ks_2samp(air_df_train['NO2(GT)'], air_df_test2['NO2(GT)'])
print("\nMean for test set 2:", air_df_test2['NO2(GT)'].mean())
print("KS Test for test2.csv:")
print("KS Statistic: {ks_statistic_test2}")
print("P-value: {p_value_test2}")

if p_value_test1 < 0.05:
    print("\nReject the null hypothesis for test1.csv")
else:
    print("\nFail to reject the null hypothesis for test1.csv")

if p_value_test2 < 0.05:
    print("Reject the null hypothesis for test2.csv")
else:
    print("Fail to reject the null hypothesis for test2.csv")
```

```
Mean for train set: 94.57946026986507

Mean for test set 1: 94.53262518968134
KS Test for test1.csv:
KS Statistic: 0.017062220028073977
P-value: 0.9971378232852736

Mean for test set 2: 134.7030456852792
KS Test for test2.csv:
KS Statistic: 0.3688536442438679
P-value: 2.53172387531317e-74

Fail to reject the null hypothesis for test1.csv
Reject the null hypothesis for test2.csv
```

Therefore, there is a distributional difference in the values of `NO2(GT)` between the test sets.

3. **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.** The 2nd test set exhibits a covariate shift relative to the training set, since:

- The p-value for test set 1 and train set is `0.99714`
- The p-value for test set 2 and train set is around `0`

This rejects the Null Hypothesis for Test2 and shows strong covariate shift in `test2` dataset with respect to the `train` set.