Team number: 27
Team members:
Siju Niyati Samji 23110312
Makkena Lakshmi Manasa 23110193

Google colab link: STT_A10.ipynb

Screenshots of the code:
Graphs are added for better understanding.

Part 1: A/B Testing using Ad Click Prediction

[4]

```python
#importing all the needed libraries
import pandas as pd
import numpy as np
from scipy.stats import ks_2samp
from statsmodels.stats.proportion import proportions_ztest
import matplotlib.pyplot as plt
from tabulate import tabulate
```

1.Load the dataset into a pandas DataFrame.

```python
# Load Ad Click Prediction Dataset
ad_df = pd.read_csv('/content/ad_click_dataset.csv')

print(ad_df)
```

```
        id full_name   age      gender device_type ad_position  \
0      670   User670  22.0         NaN     Desktop         Top
1     3044  User3044   NaN        Male     Desktop         Top
2     5912  User5912  41.0  Non-Binary         NaN        Side
3     5418  User5418  34.0        Male         NaN         NaN
4     9452  User9452  39.0  Non-Binary         NaN         NaN
...    ...       ...   ...         ...         ...         ...
9995  8510  User8510   NaN         NaN      Mobile         Top
9996  7843  User7843   NaN      Female     Desktop      Bottom
9997  3914  User3914   NaN        Male      Mobile        Side
9998  7924  User7924   NaN         NaN     Desktop         NaN
9999  3056  User3056  44.0        Male      Tablet         Top

     browsing_history time_of_day  click
0            Shopping   Afternoon      1
1                 NaN         NaN      1
2           Education       Night      1
3       Entertainment     Evening      1
4        Social Media     Morning      0
...               ...         ...    ...
9995        Education         NaN      0
9996    Entertainment         NaN      0
9997              NaN     Morning      0
9998         Shopping     Morning      1
9999     Social Media     Morning      0

[10000 rows x 9 columns]
```

2. Perform necessary data cleaning and preprocessing: Handle missing values Convert categorical columns (e.g., gender, ad_position)

```python
# --- Preprocessing ---
# Handle missing values
ad_df=ad_df.dropna()

#convert into catagorical columns
ad_df['gender']=ad_df['gender'].astype('category').cat.codes
ad_df['ad_position']=ad_df['ad_position'].astype('category').cat.codes

print(ad_df)
```

```
        id full_name   age  gender device_type  ad_position browsing_history  \
17     188   User188  56.0       0      Tablet            0             News
25    4890  User4890  43.0       1      Tablet            0        Education
33    4985  User4985  37.0       1      Mobile            2             News
52    9888  User9888  49.0       1      Mobile            2             News
102   8201  User8201  59.0       0     Desktop            0     Social Media
...    ...       ...   ...     ...         ...          ...              ...
9951  7268  User7268  28.0       0     Desktop            0             News
9952  5912  User5912  41.0       2      Mobile            1        Education
9960  9638  User9638  64.0       2     Desktop            2    Entertainment
9986  5574  User5574  52.0       0     Desktop            0         Shopping
9999  3056  User3056  44.0       1      Tablet            2     Social Media

     time_of_day  click
17       Morning      1
25     Afternoon      1
33       Evening      0
52       Morning      1
102      Morning      0
...          ...    ...
9951     Evening      1
9952       Night      1
9960     Morning      0
9986   Afternoon      1
9999     Morning      0
```

0s completed at 4:37 PM

3.Split the dataset into two groups: Group A: Users with ad_position = 0 (Top) Group B: Users with ad_position = 1 (Bottom)

```python
# Split into two groups
group_A = ad_df[ad_df['ad_position'] == 0]  # Top position
group_B = ad_df[ad_df['ad_position'] == 1]  # Bottom position

print('group A:\n',group_A)
print('group B:\n',group_B)
```

```
group A:
        id full_name   age  gender device_type  ad_position browsing_history  \
17     188   User188  56.0       0      Tablet            0             News
25    4890  User4890  43.0       1      Tablet            0        Education
102   8201  User8201  59.0       0     Desktop            0     Social Media
154    118   User118  43.0       0      Tablet            0     Social Media
170   3062  User3062  34.0       1     Desktop            0    Entertainment
...    ...       ...   ...     ...         ...          ...              ...
9866  6989  User6989  28.0       1      Mobile            0         Shopping
9904  7267  User7267  20.0       1     Desktop            0         Shopping
9925  5574  User5574  52.0       0     Desktop            0         Shopping
9951  7268  User7268  28.0       0     Desktop            0             News
9986  5574  User5574  52.0       0     Desktop            0         Shopping

     time_of_day  click
17       Morning      1
25     Afternoon      1
102      Morning      0
154        Night      0
170      Evening      1
...          ...    ...
9866   Afternoon      1
9904       Night      0
9925   Afternoon      1
9951     Evening      1
9986   Afternoon      1

[283 rows x 9 columns]
```

```
group B:
        id full_name   age  gender device_type  ad_position browsing_history  \
122   5484  User5484  64.0       0      Mobile            1        Education
135   5703  User5703  55.0       0      Tablet            1         Shopping
140    820   User820  39.0       2      Mobile            1     Social Media
146   4630  User4630  19.0       2      Tablet            1         Shopping
244   6167  User6167  20.0       1      Mobile            1         Shopping
...    ...       ...   ...     ...         ...          ...              ...
9843  2812  User2812  25.0       1      Tablet            1         Shopping
9845  2602  User2602  61.0       0     Desktop            1         Shopping
9917  6075  User6075  32.0       2     Desktop            1     Social Media
9926  4620  User4620  43.0       2      Mobile            1    Entertainment
9952  5912  User5912  41.0       2      Mobile            1        Education

     time_of_day  click
122      Morning      0
135        Night      0
140    Afternoon      0
146      Evening      0
244        Night      1
...          ...    ...
9843     Morning      0
9845   Afternoon      1
9917       Night      1
9926       Night      1
9952       Night      1

[258 rows x 9 columns]
```

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

[8]
```
# Get the total number of click
click_A = group_A['click'].sum()
n_A = len(group_A) #total entries with top position

click_B = group_B['click'].sum()
n_B = len(group_B) #total entries with bottom position

# Perform two-sample z-test
count = np.array([click_A, click_B])
nobs = np.array([n_A, n_B])
z_stat, p_value = proportions_ztest(count, nobs)
```

5. Print the following: The z-score The p-value

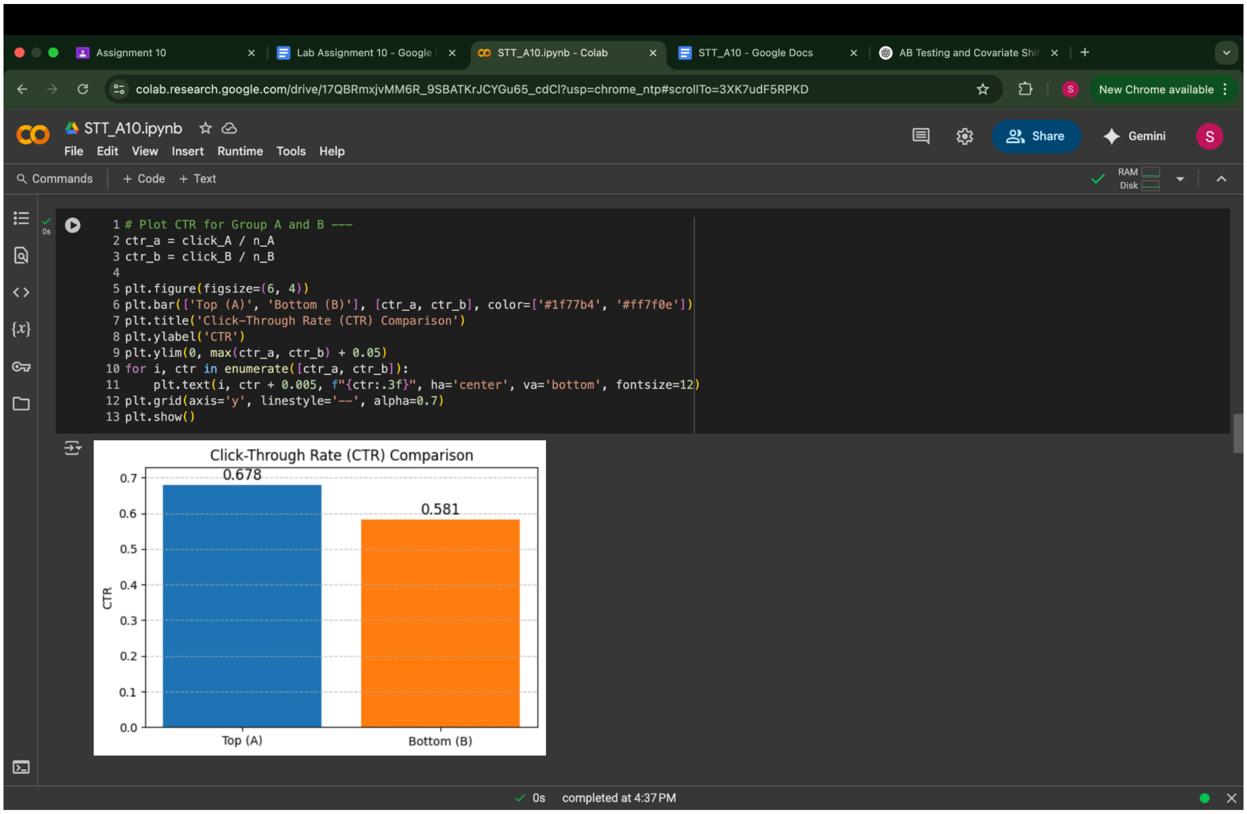
```
#outputs
print(f"Clicks (Top): {click_A} / {n_A}")
print(f"Clicks (Bottom): {click_B} / {n_B}")
print(f"Z-score: {z_stat}")
print(f"P-value: {p_value}")
```

```
Clicks (Top): 192 / 283
Clicks (Bottom): 150 / 258
Z-score: 2.3380678956239453
P-value: 0.019383726320686367
```

6. Interpret the result: Is there a statistically significant difference in click-through rates between the two groups? Justify your answer.

Yes, there is a statistically significant difference in click-through rates between the two groups.

Based on the outputs of the two-sample z-test, we can observe that the p-value is less than the commonly used threshold which is 0.05. This shows that we can reject the null hypothesis, which assumed that there is no difference in the click-through rates between ads placed at the top (Group A) and those placed at the bottom (Group B). In other words, the data provides statistically significant evidence that ad position does influence user behavior. Specifically, users are more likely to click on ads displayed at the top of the page compared to those shown at the bottom. This insight can be valuable for marketing teams and advertisers, as it suggests that prioritizing top-position ad placements could lead to higher engagement and better campaign performance.



Part 2: Covariate Shift Detection Using Air Quality Data

1. You are provided with 3 datasets via this Google Drive link:

train.csv

test1.csv

test2.csv

Assignment 10 | Lab Assignment 10 - Google | STT_A10.ipynb - Colab | STT_A10 - Google Docs | AB Testing and Covariate Shi

colab.research.google.com/drive/17QBRmxjvMM6R_9SBATKrJCYGu65_cdCl?usp=chrome_ntp#scrollTo=3XK7udF5RPKD

New Chrome available

STT_A10.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text

2. Load all three datasets using pandas.

```python
# Load datasets
test1= pd.read_csv('/content/test1.csv')
test2= pd.read_csv('/content/test2.csv')
train_df= pd.read_csv('/content/train.csv')

print('train:\n', train_df)
print('test1:\n',test1)
print('test2: \n', test2)
```

```
train:
       Unnamed: 0        Date      Time CO(GT)  PT08.S1(CO)  NMHC(GT) C6H6(GT)  \
0           1849  26/05/2004  19.00.00   -200       1130.0    -200.0     22,7
1           2533  24/06/2004  07.00.00    1,2       1030.0    -200.0      6,9
2           3047  15/07/2004  17.00.00    3,2       1164.0    -200.0     20,3
3            805  13/04/2004  07.00.00    3,9       1496.0     524.0     19,1
4           2962  12/07/2004  04.00.00   -200        780.0    -200.0      1,8
...          ...         ...       ...    ...          ...       ...      ...
3195        3830  17/08/2004  08.00.00    2,5       1155.0    -200.0     16,0
3196        1897  28/05/2004  19.00.00    1,8       1100.0    -200.0     11,4
3197        2538  24/06/2004  12.00.00    1,2        959.0    -200.0      5,0
3198        1146  27/04/2004  12.00.00    2,4       1179.0     255.0     14,6
3199        3489  03/08/2004  03.00.00   -200        838.0    -200.0      1,7

      PT08.S2(NMHC)  NOx(GT)  PT08.S3(NOx)  NO2(GT)  PT08.S4(NO2)  \
0            1368.0   -200.0         933.0   -200.0        1709.0
1             851.0    102.0         824.0     68.0        1700.0
2            1306.0    259.0         648.0    198.0        1886.0
3            1272.0    328.0         667.0    130.0        2011.0
4             568.0     24.0        1200.0     34.0        1331.0
...             ...      ...           ...      ...           ...
3195         1181.0    167.0         580.0     82.0        1850.0
3196         1028.0    112.0         900.0    103.0        1710.0
3197          763.0     56.0         976.0     58.0        1561.0
3198         1137.0    132.0         829.0     87.0        1651.0
3199          559.0   -200.0        1186.0   -200.0        1428.0
```

✓ 0s completed at 4:37 PM

Assignment 10 | Lab Assignment 10 - Google | STT_A10.ipynb - Colab | STT_A10 - Google Docs | AB Testing and Covariate Shi

colab.research.google.com/drive/17QBRmxjvMM6R_9SBATKrJCYGu65_cdCl?usp=chrome_ntp#scrollTo=3XK7udF5RPKD

New Chrome available

STT_A10.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text

```
      PT08.S5(O3)     T    RH      AH  Unnamed: 15  Unnamed: 16
0          1269.0  26,7  19,5  0,6754          NaN          NaN
1           983.0  21,9  57,0  1,4742          NaN          NaN
2          1218.0  35,5  19,1  1,0888          NaN          NaN
3          1399.0  11,0  64,2  0,8398          NaN          NaN
4           501.0  19,9  51,3  1,1803          NaN          NaN
...           ...   ...   ...     ...          ...          ...
3195       1302.0  24,3  47,8  1,4321          NaN          NaN
3196        994.0  22,1  44,1  1,1573          NaN          NaN
3197        549.0  31,9  30,3  1,4125          NaN          NaN
3198        949.0  27,2  22,0  0,7785          NaN          NaN
3199        559.0  28,0  45,2  1,6832          NaN          NaN

[3200 rows x 18 columns]
test1:
     Unnamed: 0        Date      Time CO(GT)  PT08.S1(CO)  NMHC(GT) C6H6(GT)  \
0          3123  18/07/2004  21.00.00    1,2       1067.0    -200.0      9,0
1           877  16/04/2004  07.00.00    4,5       1657.0     523.0     23,2
2          3457  01/08/2004  19.00.00    1,4       1037.0    -200.0      8,0
3          1494  12/05/2004  00.00.00    1,7       1122.0    -200.0      8,7
4           713  09/04/2004  11.00.00    2,6       -200.0     262.0   -200,0
..          ...         ...       ...    ...          ...       ...      ...
795        1345  05/05/2004  19.00.00    1,6       1000.0    -200.0      8,5
796        1269  02/05/2004  15.00.00    1,5       1047.0    -200.0      7,0
797          49  12/03/2004  19.00.00    3,7       1525.0     242.0     18,2
798        3202  22/07/2004  04.00.00      1       1015.0    -200.0      5,4
799         908  17/04/2004  14.00.00   -200        926.0    -200.0      3,9

    PT08.S2(NMHC)  NOx(GT)  PT08.S3(NOx)  NO2(GT)  PT08.S4(NO2)  PT08.S5(O3)  \
0           938.0    102.0         825.0     99.0        1520.0        912.0
1          1384.0    352.0         579.0    109.0        2176.0       1600.0
2           900.0     75.0         817.0     95.0        1584.0        619.0
3           926.0    105.0         805.0     88.0        1619.0       1174.0
4          -200.0    219.0        -200.0    121.0        -200.0       -200.0
..            ...      ...           ...      ...           ...          ...
795         918.0     68.0         881.0     67.0        1661.0        802.0
796         859.0     92.0         948.0     84.0        1538.0        677.0
797        1246.0    202.0         821.0    145.0        1847.0       1448.0
798         781.0     72.0         796.0     70.0        1595.0       1181.0
799         707.0   -200.0        1130.0   -200.0        1429.0        532.0
```

✓ 0s completed at 4:37 PM

STT_A10.ipynb

File Edit View Insert Runtime Tools Help

```
        T    RH      AH  Unnamed: 15  Unnamed: 16
0    29,7  24,8  1,0160          NaN          NaN
1    12,8  71,0  1,0428          NaN          NaN
2    33,1  32,7  1,6200          NaN          NaN
3    16,9  58,8  1,1250          NaN          NaN
4    -200  -200    -200          NaN          NaN
..    ...   ...     ...          ...          ...
795  17,5  63,9  1,2667          NaN          NaN
796  23,9  34,3  1,0020          NaN          NaN
797  14,4  43,4  0,7084          NaN          NaN
798  23,8  55,4  1,6073          NaN          NaN
799  19,3  46,6  1,0298          NaN          NaN

[800 rows x 18 columns]
test2:
     Unnamed: 0        Date      Time CO(GT)  PT08.S1(CO)  NMHC(GT) C6H6(GT)  \
0          8500  27/02/2005  22.00.00    1,0        875.0    -200.0      2,1
1          8501  27/02/2005  23.00.00    1,3        943.0    -200.0      3,9
2          8502  28/02/2005  00.00.00    1,6        947.0    -200.0      3,8
3          8503  28/02/2005  01.00.00    1,0        865.0    -200.0      1,8
4          8504  28/02/2005  02.00.00    0,6        823.0    -200.0      1,0
..          ...         ...       ...    ...          ...       ...      ...
795        9295  02/04/2005  01.00.00    1,1        838.0    -200.0      1,9
796        9296  02/04/2005  02.00.00    0,6        835.0    -200.0      1,5
797        9297  02/04/2005  03.00.00    0,5        820.0    -200.0      1,4
798        9298  02/04/2005  04.00.00    0,4        815.0    -200.0      0,9
799        9299  02/04/2005  05.00.00    0,4        842.0    -200.0      1,4

    PT08.S2(NMHC)  NOx(GT)  PT08.S3(NOx)  NO2(GT)  PT08.S4(NO2)  PT08.S5(O3)  \
0           594.0    128.0        1079.0    105.0         793.0        451.0
1           703.0    169.0         950.0    119.0         870.0        581.0
2           697.0    215.0         913.0    150.0         878.0        698.0
3           566.0    111.0        1119.0     94.0         797.0        423.0
4           503.0     60.0        1268.0     56.0         755.0        332.0
..            ...      ...           ...      ...           ...          ...
795         580.0    167.0        1055.0    127.0         791.0        516.0
796         546.0    102.0        1111.0     81.0         768.0        391.0
797         536.0     92.0        1137.0     70.0         777.0        374.0
798         488.0     63.0        1223.0     49.0         742.0        326.0
799         535.0     59.0        1140.0     44.0         805.0        352.0
```

0s completed at 4:37 PM

```
        T    RH      AH  Unnamed: 15  Unnamed: 16
0     4,5  48,0  0,4085          NaN          NaN
1     4,3  48,6  0,4069          NaN          NaN
2     4,0  50,0  0,4115          NaN          NaN
3     4,0  52,9  0,4338          NaN          NaN
4     4,0  51,0  0,4200          NaN          NaN
..    ...   ...     ...          ...          ...
795  12,7  33,6  0,4902          NaN          NaN
796  12,2  35,6  0,5038          NaN          NaN
797  11,9  35,6  0,4945          NaN          NaN
798  11,2  38,3  0,5099          NaN          NaN
799  11,0  40,5  0,5316          NaN          NaN

[800 rows x 18 columns]
```

3. 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
# Perform KS test on 'NO2(GT)'
ks_stat_1, p_val_1 = ks_2samp(
    train_df['NO2(GT)'][(train_df['NO2(GT)'] >= 0)].dropna(),
    test1['NO2(GT)'][(test1['NO2(GT)'] >= 0)].dropna()
)

ks_stat_2, p_val_2 = ks_2samp(
    train_df['NO2(GT)'][(train_df['NO2(GT)'] >= 0)].dropna(),
    test2['NO2(GT)'][(test2['NO2(GT)'] >= 0)].dropna()
)


```

4. Report the KS statistic and p-value for each feature.

```python
print("Test1 vs Train:")
print(f"KS Statistic: {ks_stat_1}, P-value: {p_val_1}")

print("\nTest2 vs Train:")
print(f"KS Statistic: {ks_stat_2}, P-value: {p_val_2}")
```

```
Test1 vs Train:
KS Statistic: 0.017062220028073977, P-value: 0.9971378232852736

Test2 vs Train:
KS Statistic: 0.3688536442438679, P-value: 2.53172387531317e-74
```

5. 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.

[13]

```
# Determine covariate shift according to Ks test on only NO2(GT)
if p_val_1 < 0.05:
    print("\nConclusion: Test1 shows covariate shift w.r.t training set.")
else:
    print("\nConclusion: Test1 does NOT show significant covariate shift w.r.t training set.")

if p_val_2 < 0.05:
    print("Conclusion: Test2 shows covariate shift w.r.t training set.")
else:
    print("Conclusion: Test2 does NOT show significant covariate shift w.r.t training set.")
```

```
Conclusion: Test1 does NOT show significant covariate shift w.r.t training set.
Conclusion: Test2 shows covariate shift w.r.t training set.
```

0s completed at 4:37 PM

The null hypothesis states that the two samples being compared are drawn from the same distribution.

If the p value is less than 0.05 that means we can reject the null hypothesis, which means those samples are not taken form the same distribution. that indicates covariate shift.

To determine which test dataset exhibits a covariate shift relative to the training dataset, we conducted the Kolmogorov–Smirnov (KS) test on the NO2(GT) feature. The results for Test1 vs Train showed a KS Statistic of 0.017 and a p-value of 0.99, which indicates a high similarity between their distributions. Since the p-value is much greater than 0.05, we fail to reject the null hypothesis, suggesting that there is no significant distributional difference between Test1 and the training set for this feature.

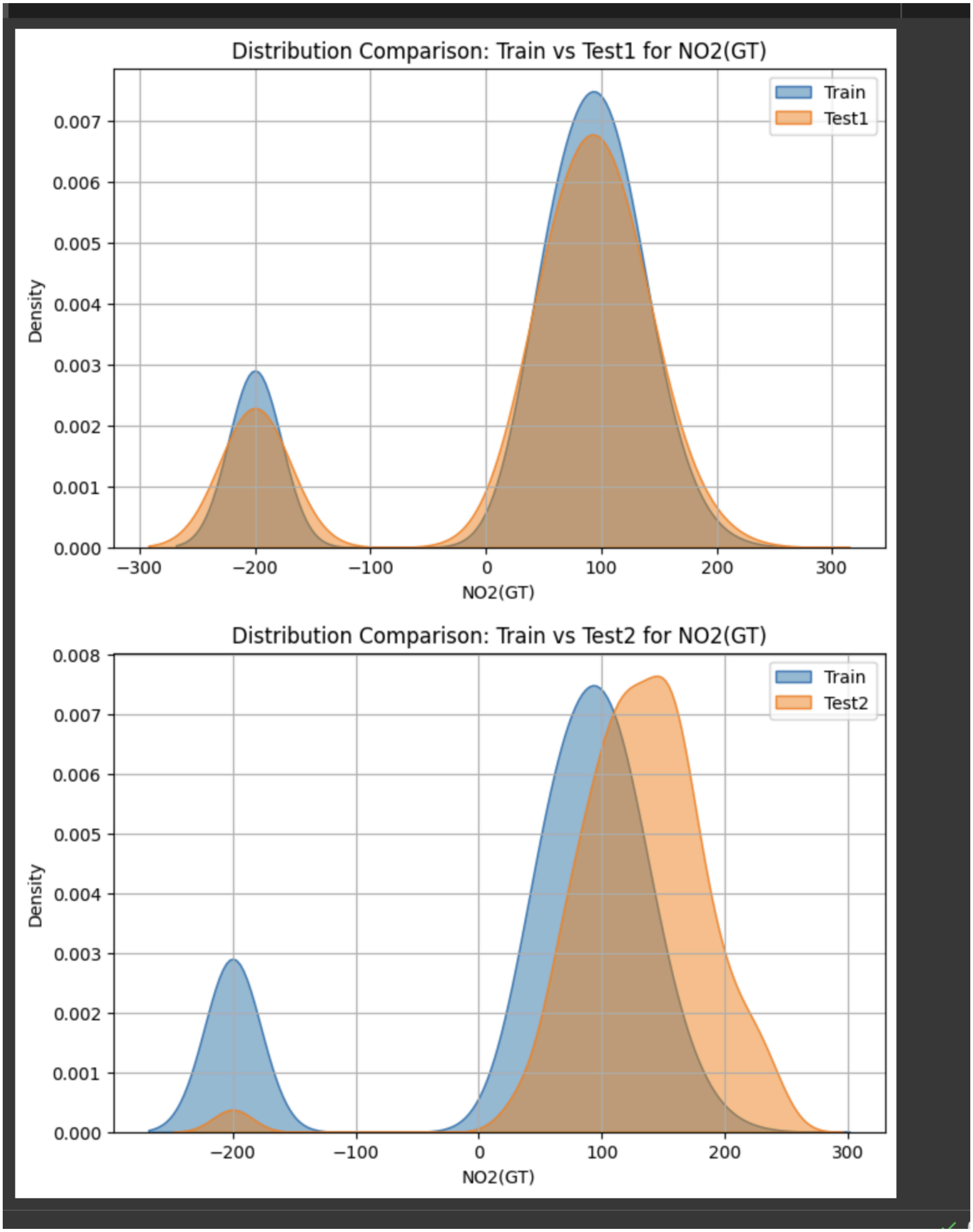
on the other hand, the results for Test2 vs Train showed a KS Statistic of 0.36 and an extremely small p-value, which is far less than the significance threshold which is 0.05. This provides strong evidence to reject the null hypothesis, indicating that Test2's NO2(GT) distribution is significantly different from that of the training data.

Therefore, based on the KS test results, we conclude that Test2 exhibits a covariate shift relative to the training dataset, while Test1 does not.

```
import seaborn as sns

def plot_distribution_comparison(train_col, test_col, feature_name, test_name):
    plt.figure(figsize=(8, 5))
    sns.kdeplot(train_col.dropna(), label='Train', fill=True, alpha=0.5)
    sns.kdeplot(test_col.dropna(), label=test_name, fill=True, alpha=0.5)
    plt.title(f'Distribution Comparison: Train vs {test_name} for {feature_name}')
    plt.xlabel(feature_name)
    plt.ylabel('Density')
    plt.legend()
    plt.grid(True)
    plt.show()

# Example:
plot_distribution_comparison(train_df['NO2(GT)'], test1['NO2(GT)'], 'NO2(GT)', 'Test1')
plot_distribution_comparison(train_df['NO2(GT)'], test2['NO2(GT)'], 'NO2(GT)', 'Test2')
```


Distribution Comparison: Train vs Test1 for NO2(GT)
Train
Test1
Density
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
−300
−200
−100
0
100
200
300
NO2(GT)
Distribution Comparison: Train vs Test2 for NO2(GT)
Train
Test2
Density
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
−200
−100
0
100
200
300
NO2(GT)