

## Upper Confidence Bound (UCB)

### Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import math
```

### Importing Dataset

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

### Implementing UCB

```
N = 10000
d = 10
ads_selected = []
numbers_of_selections = [0] * d
sums_of_rewards = [0] * d
total_reward = 0
for n in range(0, N):
    ad = 0
    max_upper_bound = 0
    for i in range(0, d):
        if (numbers_of_selections[i] > 0):
            average_reward = sums_of_rewards[i] / numbers_of_selections[i]
            delta_i = math.sqrt(3/2 * math.log(n + 1) / numbers_of_selections[i])
            upper_bound = average_reward + delta_i
        else:
            upper_bound = 1e400
    # Move this check outside the if/else block so it is always executed
    if (upper_bound > max_upper_bound):
        max_upper_bound = upper_bound
        ad = i

ads_selected.append(ad)
numbers_of_selections[ad] += 1
reward = df.values[n,ad]
# Initialize average_reward to 0 if it hasn't been calculated yet
if 'average_reward' not in locals():
    average_reward = 0
sums_of_rewards[ad] = sums_of_rewards[ad] + reward # Update with the actual reward, not the average
total_reward = total_reward + reward
```

### Visualizing the results

```

import plotly.graph_objects as go

# Creating a histogram
fig = go.Figure()

fig.add_trace(go.Histogram(
    x=ads_selected,
    nbinsx=10, # Number of bins
    marker_color='blue', # Set color to blue
    opacity=0.75
))

# Updating layout for better visual appeal
fig.update_layout(
    title='Interactive Histogram of Ads Selections',
    xaxis=dict(
        title='Ads',
        tickmode='linear'
    ),
    yaxis=dict(
        title='Number of Times Each Ad Was Selected'
    ),
    bargap=0.2,
    bargroupgap=0.1
)

# Adding hover info
fig.update_traces(
    hoverinfo='x+y',
    hoverlabel=dict(
        bgcolor='white',
        font_size=16,
        font_family='Rockwell'
    )
)

# Show the plot
fig.show()

```



Interactive Histogram of Ads Selections

