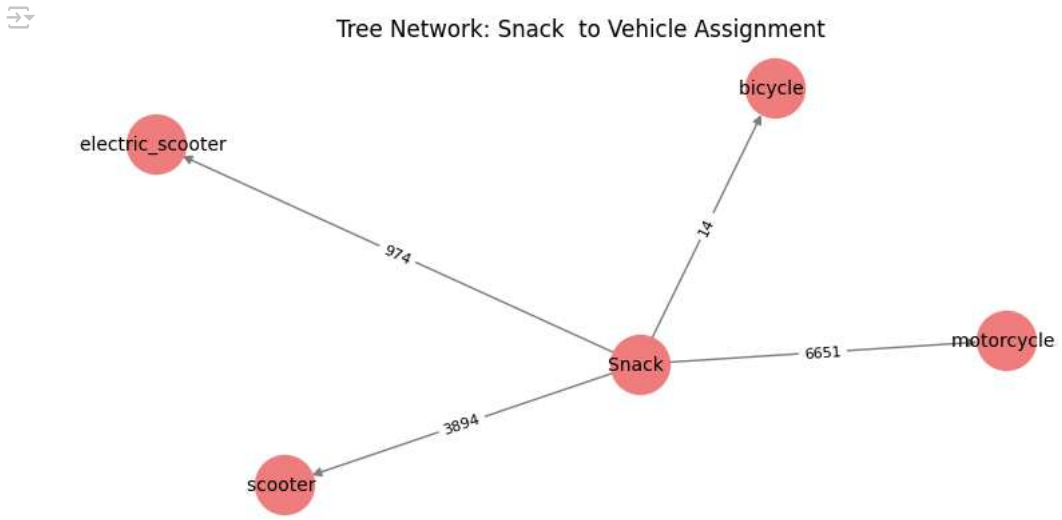
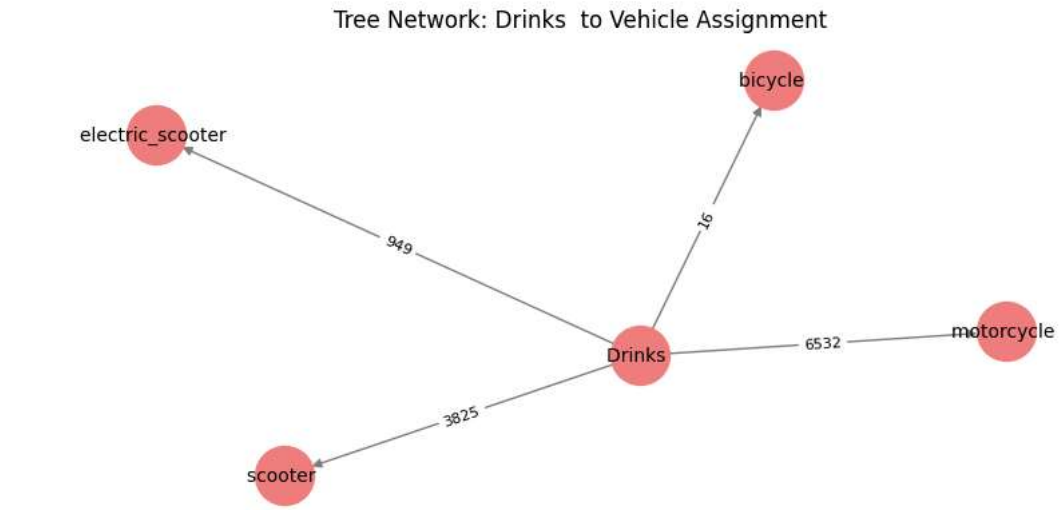
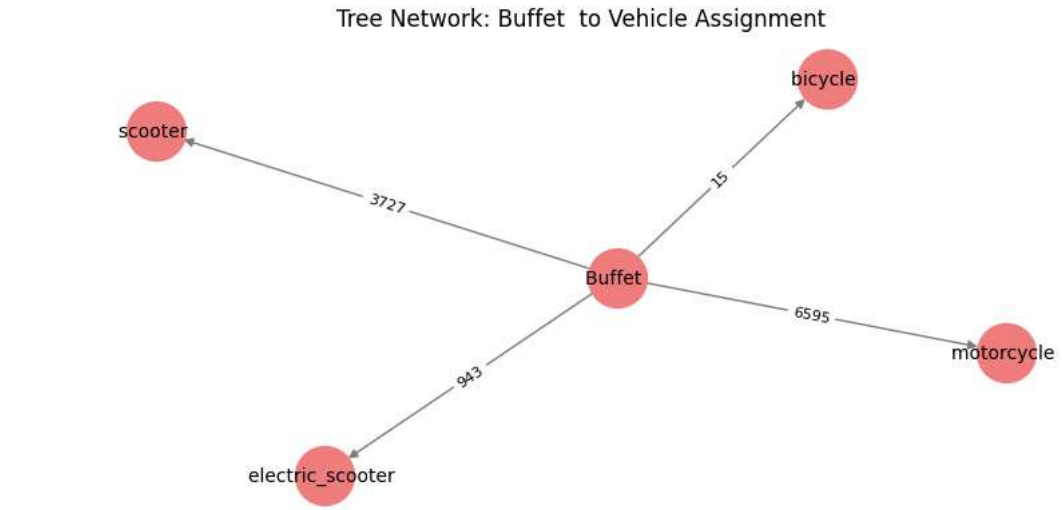
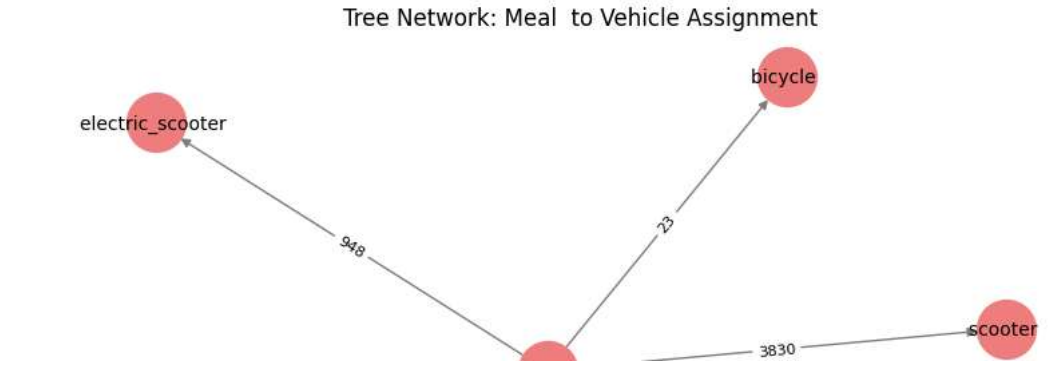
```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Load dataset

import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/train.csv')
# Drop missing values
df = df.dropna(subset=["Type_of_order", "Type_of_vehicle"])

# Get unique food types
food_types = df["Type_of_order"].unique()

# Plot separate trees for each food type
fig, axes = plt.subplots(nrows=len(food_types), figsize=(8, len(food_types) * 4))

for idx, food in enumerate(food_types):
    ax = axes[idx] if len(food_types) > 1 else axes  # Handle single plot case
    G = nx.DiGraph()

    # Filter data for each food type
    food_df = df[df["Type_of_order"] == food]

    # Add edges between food type and vehicle type
    for _, row in food_df.iterrows():
        vehicle_type = row["Type_of_vehicle"]

        if G.has_edge(food, vehicle_type):
            G[food][vehicle_type]["weight"] += 1
        else:
            G.add_edge(food, vehicle_type, weight=1)

    # Draw the graph
    pos = nx.spring_layout(G, seed=42)
    edges = G.edges(data=True)

    nx.draw(G, pos, with_labels=True, node_color="lightcoral", edge_color="gray", node_size=1000, font_size=10, ax=ax)
    edge_labels = {(u, v): d["weight"] for u, v, d in edges}
    nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=8, ax=ax)

    ax.set_title(f"Tree Network: {food} to Vehicle Assignment")

plt.tight_layout()
plt.show()
```


Tree Network: Snack to Vehicle Assignment
bicycle
electric_scooter
974
14
Snack
6651
motorcycle
3894
scooter


Tree Network: Drinks to Vehicle Assignment
bicycle
electric_scooter
949
16
Drinks
6532
motorcycle
3825
scooter


Tree Network: Buffet to Vehicle Assignment
bicycle
scooter
3727
15
Buffet
6595
motorcycle
943
electric_scooter


Tree Network: Meal to Vehicle Assignment
bicycle
electric_scooter
948
23
3830
scooter