```
# Dynamic Pricing for Urban Parking Lots - Hackathon Capstone
# Compatible with pathwaycom/pathway (v0.14.3) and pathwaycom/llm-app

# Step 1: Set up the environment in Google Colab
!pip install pathway==0.14.3  # Specific version for compatibility
!pip install numpy pandas bokeh
```

```
Collecting pathway==0.14.3
  Downloading pathway-0.14.3-cp310-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (40 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 40.9/40.9 kB 2.5 MB/s eta 0:00:00
Requirement already satisfied: aiohttp>=3.8.4 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (3.11.15)
Requirement already satisfied: click>=8.1 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (8.2.1)
Requirement already satisfied: geopy>=2.4.0 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (2.4.1)
Collecting h3>=3.7.6 (from pathway==0.14.3)
  Downloading h3-4.3.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (18 kB)
Requirement already satisfied: numpy>=1.21 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (2.0.2)
Requirement already satisfied: pandas>=2.1 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (2.2.2)
Requirement already satisfied: scikit-learn>=1.0 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (1.6.1)
Requirement already satisfied: shapely>=2.0.1 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (2.1.1)
Collecting sqlglot==10.6.1 (from pathway==0.14.3)
  Downloading sqlglot-10.6.1-py3-none-any.whl.metadata (14 kB)
Requirement already satisfied: pyarrow>=10.0.0 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (18.1.0)
Requirement already satisfied: requests>=2.31.0 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (2.32.3)
Collecting python-sat>=0.1.8.dev0 (from pathway==0.14.3)
  Downloading python_sat-1.8.dev17-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl.metadata (1.5 kB)
Collecting beartype<0.16.0,>=0.14.0 (from pathway==0.14.3)
  Downloading beartype-0.15.0-py3-none-any.whl.metadata (28 kB)
Requirement already satisfied: rich>=12.6.0 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (13.9.4)
Collecting diskcache>=5.2.1 (from pathway==0.14.3)
  Downloading diskcache-5.6.3-py3-none-any.whl.metadata (20 kB)
Collecting boto3>=1.26.76 (from pathway==0.14.3)
  Downloading boto3-1.39.3-py3-none-any.whl.metadata (6.6 kB)
Requirement already satisfied: google-api-python-client>=2.108.0 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (2.
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (4.14.0)
Requirement already satisfied: panel>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (1.7.2)
Collecting jupyter-bokeh>=3.0.7 (from pathway==0.14.3)
  Downloading jupyter_bokeh-4.0.5-py3-none-any.whl.metadata (7.1 kB)
Collecting jmespath>=1.0.1 (from pathway==0.14.3)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting aiohttp-cors>=0.7.0 (from pathway==0.14.3)
  Downloading aiohttp_cors-0.8.1-py3-none-any.whl.metadata (20 kB)
Collecting opentelemetry-api>=1.22.0 (from pathway==0.14.3)
  Downloading opentelemetry_api-1.34.1-py3-none-any.whl.metadata (1.5 kB)
Collecting opentelemetry-sdk>=1.22.0 (from pathway==0.14.3)
  Downloading opentelemetry_sdk-1.34.1-py3-none-any.whl.metadata (1.6 kB)
Collecting opentelemetry-exporter-otlp-proto-grpc>=1.22.0 (from pathway==0.14.3)
  Downloading opentelemetry_exporter_otlp_proto_grpc-1.34.1-py3-none-any.whl.metadata (2.4 kB)
Collecting fs>=2.4.16 (from pathway==0.14.3)
  Downloading fs-2.4.16-py2.py3-none-any.whl.metadata (6.3 kB)
Collecting async-lru>=2.0.4 (from pathway==0.14.3)
  Downloading async_lru-2.0.5-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: networkx>=3.2.1 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (3.5)
Collecting google-cloud-pubsub>=2.21.1 (from pathway==0.14.3)
  Downloading google_cloud_pubsub-2.30.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: google-cloud-bigquery in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (3.34.0)
Collecting pydantic~=2.7.0 (from pathway==0.14.3)
  Downloading pydantic-2.7.4-py3-none-any.whl.metadata (109 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 109.4/109.4 kB 7.2 MB/s eta 0:00:00
Requirement already satisfied: gitpython>=3.1.43 in /usr/local/lib/python3.11/dist-packages (from pathway==0.14.3) (3.1.44)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp>=3.8.4->pathway==0.14.
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp>=3.8.4->pathway==0.14.3) (1.3
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp>=3.8.4->pathway==0.14.3) (25.3.0
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp>=3.8.4->pathway==0.14.3) (1.
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp>=3.8.4->pathway==0.14.3) (
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp>=3.8.4->pathway==0.14.3) (0.3
```

```
# Import required libraries
import numpy as np
import pandas as pd
import pathway as pw
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import math
import time
```

```
Requirement already satisfied: httplib2<1.0.0,>=0.19.0 in /usr/local/lib/python3.11/dist-packages (from google-api-python-client>=2.10
```

```
#Enable bokeh output in Colab
output_notebook()
```

```
# Step 2: Load and Preprocess Data
try:
    df = pd.read_csv('dataset (1).csv')
    print("Dataset loaded successfully. Columns:", df.columns.tolist())
except FileNotFoundError:
    print("Error: 'dataset (1).csv' not found. Please upload the file to Colab.")
    raise
```

```
Requirement already satisfied: narwhals>=1.13 in /usr/local/lib/python3.11/dist-packages (from bokeh==3.*->jupyter-bokeh>=3.0.7->pathw
Dataset loaded successfully. Columns: ['ID', 'SystemCodeNumber', 'Capacity', 'Latitude', 'Longitude', 'Occupancy', 'VehicleType', 'Traff
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.11/dist-packages (from bokeh==3.*->jupyter-bokeh>=3.0.7->path
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.11/dist-packages (from bokeh==3.*->jupyter-bokeh>=3.0.7->pathwa
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.11/dist-packages (from bokeh==3.*->jupyter-bokeh>=3.0.7->pathway
```

```
# Combine date and time into a single timestamp (ISO 8601 for Pathway)
df['timestamp'] = pd.to_datetime(df['LastUpdatedDate'] + ' ' + df['LastUpdatedTime'],
                                format='%d-%m-%Y %H:%M:%S', errors='coerce')
if df['timestamp'].isnull().any():
    print("Warning: Dropping rows with invalid timestamps.")
    df = df.dropna(subset=['timestamp'])
df['timestamp'] = df['timestamp'].dt.strftime('%Y-%m-%dT%H:%M:%S')
```

```
Requirement already satisfied: importlib-metadata<8.8.0,>=6.0 in /usr/local/lib/python3.11/dist-packages (from opentelemetry-api>=1.22
[illegible]
```

```
# Rename columns for consistency
df = df.rename(columns={
    'SystemCodeNumber': 'parking_lot_id',
    'Latitude': 'latitude',
    'Longitude': 'longitude',
    'QueueLength': 'queue_length',
    'IsSpecialDay': 'is_special_day'
})
```

```
Requirement already satisfied: bleach in /usr/local/lib/python3.11/dist-packages (from panel>=1.3.1->pathway==0.14.3) (6.2.0)
```

```
# Encode TrafficConditionNearby numerically
traffic_map = {'low': 0, 'average': 1, 'high': 2}
df['traffic_level'] = df['TrafficConditionNearby'].map(traffic_map)
```

```
Requirement already satisfied: pyviz-comms>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from panel>=1.3.1->pathway==0.14.3) (3.0
```

```
# Assign vehicle type weights
vehicle_weights = {'cycle': 0.5, 'bike': 0.75, 'car': 1.0, 'truck': 1.5}
df['vehicle_weight'] = df['VehicleType'].map(vehicle_weights)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31.0->pathway==0.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31.0->pathway==0.14.3) (3.10)
```

```
# Simulate competitor prices (since not provided)
np.random.seed(42)
df['competitor_price'] = np.random.uniform(8, 12, size=len(df))
```

```
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.0->pathway==0.14.3) (1.15
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.0->pathway==0.14.3) (1.5
```

```
# Check for missing values
print("Missing values:\n", df.isnull().sum())
if df.isnull().any().any():
    print("Warning: Filling missing values with defaults.")
    df = df.fillna({'traffic_level': 0, 'vehicle_weight': 1.0, 'competitor_price': 10.0})
```

```
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.11/dist-packages (from google-auth!=2.24.0,!=2.25.0,<3.0.0,>=1.
Missing values:
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /usr/local/lib/python3.11/dist-packages (from google-resumable-media<3.0.
 ID                        0
Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in /usr/local/lib/python3.11/dist-packages (from ht
parking_lot_id            0
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.11/dist-packages (from importlib-metadata<8.8.0,>=6.0->opentelemet
Capacity                  0
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py->panel>=1.3.1->pathway==0.14
latitude                  0
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (from bleach->panel>=1.3.1->pathway==0.14.3) (0
longitude                 0
Requirement already satisfied: uc-micro-py in /usr/local/lib/python3.11/dist-packages (from linkify-it-py->panel>=1.3.1->pathway==0.14
Occupancy                 0
Collecting jedi>=0.16 (from ipython>=6.1.0->ipywidgets==8.*->jupyter-bokeh>=3.0.7->pathway==0.14.3)
VehicleType               0
  Downloading jedi-0.19.2-py2.py3-none-any.whl.metadata (22 kB)
TrafficConditionNearby    0
Requirement already satisfied: decorator in /usr/local/lib/python3.11/dist-packages (from ipython>=6.1.0->ipywidgets==8.*->jupyter-bok
queue_length              0
Requirement already satisfied: pickleshare in /usr/local/lib/python3.11/dist-packages (from ipython>=6.1.0->ipywidgets==8.*->jupyter-b
is_special_day            0
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from ipython>=
LastUpdatedDate           0
Requirement already satisfied: backcall in /usr/local/lib/python3.11/dist-packages (from ipython>=6.1.0->ipywidgets==8.*->jupyter-boke
LastUpdatedTime           0
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.11/dist-packages (from ipython>=6.1.0->ipywidgets==8.*->jup
timestamp                 0
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.11/dist-packages (from ipython>=6.1.0->ipywidgets==8.*->jupyter-b
traffic_level             0
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2>=2.9->bokeh==3.*->jupyter-bokeh
vehicle_weight            0
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in /usr/local/lib/python3.11/dist-packages (from pyasn1-modules>=0.2.1->google-aut
competitor_price          0
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.11/dist-packages (from jedi>=0.16->ipython>=6.1.0->ipywid
dtype: int64
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.11/dist-packages (from pexpect>4.3->ipython>=6.1.0->ipywidget
Requirement already satisfied: wcwidth in /usr/local/lib/python3.11/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->
```

```
# Save preprocessed data
df.to_csv('preprocessed_data.csv', index=False)
print("Preprocessed data saved to 'preprocessed_data.csv'. Sample:\n", df.head())
```

```
Downloading async_lru-2.0.5-py3-none-any.whl (6.1 kB)
Downloading beartype-0.15.0-py3-none-any.whl (777 kB)
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 777.6/777.6 kB 43.7 MB/s eta 0:00:00
Downloading boto3-1.39.3-py3-none-any.whl (139 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.9/139.9 kB 10.1 MB/s eta 0:00:00
Downloading diskcache-5.6.3-py3-none-any.whl (45 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.5/45.5 kB 3.0 MB/s eta 0:00:00
Downloading fs-2.4.16-py2.py3-none-any.whl (135 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.3/135.3 kB 9.3 MB/s eta 0:00:00
Downloading google_cloud_pubsub-2.30.0-py3-none-any.whl (318 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 318.4/318.4 kB 19.2 MB/s eta 0:00:00
Downloading h3-4.3.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (985 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 985.8/985.8 kB 49.0 MB/s eta 0:00:00
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Downloading jupyter_bokeh-4.0.5-py3-none-any.whl (148 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 148.6/148.6 kB 11.2 MB/s eta 0:00:00
Downloading ipywidgets-8.1.7-py3-none-any.whl (139 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.8/139.8 kB 9.7 MB/s eta 0:00:00
Downloading opentelemetry_api-1.34.1-py3-none-any.whl (65 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 65.8/65.8 kB 4.1 MB/s eta 0:00:00
Downloading opentelemetry_exporter_otlp_proto_grpc-1.34.1-py3-none-any.whl (18 kB)
Downloading opentelemetry_exporter_otlp_proto_common-1.34.1-py3-none-any.whl (18 kB)
Downloading opentelemetry_proto-1.34.1-py3-none-any.whl (55 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.7/55.7 kB 3.6 MB/s eta 0:00:00
Downloading opentelemetry_sdk-1.34.1-py3-none-any.whl (118 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 118.5/118.5 kB 9.8 MB/s eta 0:00:00
Downloading opentelemetry_semantic_conventions-0.55b1-py3-none-any.whl (196 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 196.2/196.2 kB 13.0 MB/s eta 0:00:00
Downloading pydantic-2.7.4-py3-none-any.whl (409 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 409.0/409.0 kB 23.2 MB/s eta 0:00:00
Downloading pydantic_core-2.18.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 62.1 MB/s eta 0:00:00
```

```
Preprocessed data saved to 'preprocessed_data.csv'. Sample:
   ID parking_lot_id  Capacity   latitude  longitude  Occupancy VehicleType  \
0   0   BHMBCCMKT01       577  26.144536  91.736172         61         car
1   1   BHMBCCMKT01       577  26.144536  91.736172         64         car
2   2   BHMBCCMKT01       577  26.144536  91.736172         80         car
3   3   BHMBCCMKT01       577  26.144536  91.736172        107         car
4   4   BHMBCCMKT01       577  26.144536  91.736172        150        bike

  TrafficConditionNearby  queue_length  is_special_day LastUpdatedDate  \
0                    low             1               0      04-10-2016
1                    low             1               0      04-10-2016
2                    low             2               0      04-10-2016
3                    low             2               0      04-10-2016
4                    low             2               0      04-10-2016

  LastUpdatedTime           timestamp  traffic_level  vehicle_weight  \
0        07:59:00 2016-10-04T07:59:00              0            1.00
1        08:25:00 2016-10-04T08:25:00              0            1.00
2        08:59:00 2016-10-04T08:59:00              0            1.00
3        09:32:00 2016-10-04T09:32:00              0            1.00
4        09:59:00 2016-10-04T09:59:00              0            0.75

   competitor_price
0          9.498160
1         11.802857
2         10.927976
3         10.394634
4          8.624075
```

```
# Step 3: Define Pathway Schema
class ParkingSchema(pw.Schema):
    timestamp: pw.DateTimeNaive
    parking_lot_id: str
    occupancy: int
    capacity: int
    queue_length: int
    traffic_level: float
    is_special_day: int
    vehicle_weight: float
    latitude: float
    longitude: float
    competitor_price: float
```

```
          Successfully uninstalled sqlglot-25.20.2
```

```
# Step 4: Pathway Streaming Pipeline
try:
    # Read data as a stream
    table = pw.io.csv.read(
        'preprocessed_data.csv',
        schema=ParkingSchema,
        mode='streaming'
    )
    print("Pathway table loaded successfully. Schema:", table.schema)
except Exception as e:
    print(f"Error in Pathway CSV read: {e}")
    raise
```

```
    Uninstalling ipywidgets-7.7.1:
      Successfully uninstalled ipywidgets-7.7.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sourc
albumentations 2.0.8 requires pydantic>=2.9.2, but you have pydantic 2.7.4 which is incompatible.
bigframes 2.8.0 requires sqlglot>=23.6.3, but you have sqlglot 10.6.1 which is incompatible.
```

```
Pathway table loaded successfully. Schema: id           | timestamp        | parking_lot_id | occupancy | capacity | queue_length | traffi
ANY_POINTER | DATE_TIME_NAIVE | STR           | INT       | INT      | INT          | FLOAT        | INT       | FLOAT      |
```

```
# Initialize base price
base_price = 10.0
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
```

```
# Model 1: Baseline Linear Pricing
def model1_price(occupancy: int, capacity: int) -> float:
    k = 0.5
    occupancy_rate = occupancy / capacity
    price = base_price + k * occupancy_rate
    return max(5.0, min(20.0, price))
```

```
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.11/dist-packages (from bokeh) (24.2)
```

```
# Model 2: Demand-Based Pricing
alpha, beta, gamma, delta, epsilon = 0.4, 0.3, 0.1, 0.1, 0.1
lambda_factor = 0.5
def model2_price(occupancy: int, capacity: int, queue_length: int,
                 traffic_level: float, is_special_day: int, vehicle_weight: float) -> float:
```

```
    max_queue = 15.0
    demand = (
        alpha * (occupancy / capacity) +
        beta * (queue_length / max_queue) -
        gamma * traffic_level +
        delta * is_special_day +
        epsilon * vehicle_weight
    )
    norm_demand = min(max(demand, 0.0), 1.0)
    price = base_price * (1 + lambda_factor * norm_demand)
    return max(5.0, min(20.0, price))


# Model 3: Competitive Pricing
def model3_price(occupancy: int, capacity: int, queue_length: int,
                competitor_price: float, other_lots: list) -> tuple[float, str]:
    max_queue = 15.0
    demand = (occupancy / capacity + queue_length / max_queue) / 2
    competitor_factor = competitor_price / base_price
    price = base_price * (1 + 0.3 * demand + 0.2 * (competitor_factor - 1))
    price = max(5.0, min(20.0, price))
    reroute = ""
    if occupancy / capacity > 0.9 and competitor_price < base_price:
        reroute = f"Reroute to {other_lots[0]}" if other_lots else ""
    return price, reroute


# Calculate distances for Model 3
lot_coords = df.groupby('parking_lot_id')[['latitude', 'longitude']].first().reset_index()
def haversine(lat1, lon1, lat2, lon2):
    R = 6371  # Earth's radius in km
    lat1, lon1, lat2, lon2 = map(math.radians, [lat1, lon1, lat2, lon2])
    dlat, dlon = lat2 - lat1, lon2 - lon1
    a = math.sin(dlat/2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon/2)**2
    c = 2 * math.asin(math.sqrt(a))
    return R * c

distances = {}
other_lots = {}
for i, lot1 in lot_coords.iterrows():
    other_lots[lot1['parking_lot_id']] = []
    for j, lot2 in lot_coords.iterrows():
        if lot1['parking_lot_id'] != lot2['parking_lot_id']:
            dist = haversine(lot1['latitude'], lot1['longitude'], lot2['latitude'], lot2['longitude'])
            distances[(lot1['parking_lot_id'], lot2['parking_lot_id'])] = dist
            if dist < 1:
                other_lots[lot1['parking_lot_id']].append(lot2['parking_lot_id'])


# Apply pricing models
@pw.udf
def apply_model3_price(parking_lot_id: str, occupancy: int, capacity: int,
                      queue_length: int, competitor_price: float) -> tuple[float, str]:
    return model3_price(occupancy, capacity, queue_length, competitor_price,
                       other_lots.get(parking_lot_id, []))

try:
    table = table.with_columns(
        price_model1=pw.apply(model1_price, pw.this.occupancy, pw.this.capacity),
        price_model2=pw.apply(model2_price, pw.this.occupancy, pw.this.capacity,
                             pw.this.queue_length, pw.this.traffic_level,
                             pw.this.is_special_day, pw.this.vehicle_weight),
        price_model3=pw.apply(apply_model3_price, pw.this.parking_lot_id,
                             pw.this.occupancy, pw.this.capacity, pw.this.queue_length,
                             pw.this.competitor_price)[0],
        reroute=pw.apply(apply_model3_price, pw.this.parking_lot_id,
                         pw.this.occupancy, pw.this.capacity, pw.this.queue_length,
                         pw.this.competitor_price)[1]
    )
except Exception as e:
    print(f"Error applying pricing models: {e}")
    raise


# Output results
try:
    # Debug: Inspect table content
    def on_change(row, time, is_addition):
        print(f"Pipeline output - Row: {row}, Time: {time}, Is addition: {is_addition}")
    pw.io.subscribe(table, on_change)

    # Write output to CSV
    pw.io.csv.write(table, 'output_prices.csv')
    print("Output configured to write to 'output_prices.csv'")
except Exception as e:
    print(f"Error configuring output: {e}")
    raise
```

```
Output configured to write to 'output_prices.csv'
```

```
# Run the pipeline
try:
    pw.run(monitoring_level=pw.MonitoringLevel.NONE)  # Minimize logging for speed
    print("Pathway pipeline executed successfully")
    time.sleep(2)  # Ensure output is written
except Exception as e:
    print(f"Error running Pathway pipeline: {e}")
    raise
```

```
ren't found in the header (fields present in table: ["2928", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "156", "car", "low", "3
ren't found in the header (fields present in table: ["2929", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "154", "car", "low", "2
ren't found in the header (fields present in table: ["2930", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "40", "truck", "low", "
ren't found in the header (fields present in table: ["2931", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "43", "bike", "low", "2
ren't found in the header (fields present in table: ["2932", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "43", "car", "low", "1"
ren't found in the header (fields present in table: ["2933", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "43", "car", "low", "2"
ren't found in the header (fields present in table: ["2934", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "46", "car", "low", "2"
ren't found in the header (fields present in table: ["2935", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "66", "car", "low", "4"
ren't found in the header (fields present in table: ["2936", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "73", "cycle", "average
ren't found in the header (fields present in table: ["2937", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "72", "cycle", "high",
ren't found in the header (fields present in table: ["2938", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "87", "car", "average",
ren't found in the header (fields present in table: ["2939", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "94", "bike", "high", "
ren't found in the header (fields present in table: ["2940", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "104", "car", "high", "
ren't found in the header (fields present in table: ["2941", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "108", "car", "high", "
ren't found in the header (fields present in table: ["2942", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "117", "car", "average"
ren't found in the header (fields present in table: ["2943", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "127", "car", "average"
ren't found in the header (fields present in table: ["2944", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "131", "car", "average"
ren't found in the header (fields present in table: ["2945", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "138", "car", "low", "2
ren't found in the header (fields present in table: ["2946", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "141", "car", "low", "3
ren't found in the header (fields present in table: ["2947", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "135", "car", "low", "1
ren't found in the header (fields present in table: ["2948", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "126", "bike", "low", "
ren't found in the header (fields present in table: ["2949", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "210", "car", "average"
ren't found in the header (fields present in table: ["2950", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "274", "bike", "low", "
ren't found in the header (fields present in table: ["2951", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "306", "bike", "low", "
ren't found in the header (fields present in table: ["2952", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "337", "cycle", "low",
ren't found in the header (fields present in table: ["2953", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "365", "truck", "averag
ren't found in the header (fields present in table: ["2954", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "388", "car", "average"
ren't found in the header (fields present in table: ["2955", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "402", "cycle", "averag
ren't found in the header (fields present in table: ["2956", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "416", "car", "high", "
ren't found in the header (fields present in table: ["2957", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "413", "bike", "high",
ren't found in the header (fields present in table: ["2958", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "417", "car", "high", "
ren't found in the header (fields present in table: ["2959", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "425", "car", "high", "
ren't found in the header (fields present in table: ["2960", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "423", "bike", "average
ren't found in the header (fields present in table: ["2961", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "429", "car", "high", "
ren't found in the header (fields present in table: ["2962", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "429", "bike", "high",
ren't found in the header (fields present in table: ["2963", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "422", "car", "low", "2
ren't found in the header (fields present in table: ["2964", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "386", "cycle", "averag
ren't found in the header (fields present in table: ["2965", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "334", "car", "low", "2
ren't found in the header (fields present in table: ["2966", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "142", "car", "low", "2
ren't found in the header (fields present in table: ["2967", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "213", "cycle", "low",
ren't found in the header (fields present in table: ["2968", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "311", "car", "low", "2
ren't found in the header (fields present in table: ["2969", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "341", "car", "low", "2
ren't found in the header (fields present in table: ["2970", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "397", "car", "low", "2
ren't found in the header (fields present in table: ["2971", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "436", "bike", "low", "
ren't found in the header (fields present in table: ["2972", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "451", "cycle", "averag
ren't found in the header (fields present in table: ["2973", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "465", "truck", "averag
ren't found in the header (fields present in table: ["2974", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "459", "car", "high", "
ren't found in the header (fields present in table: ["2975", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "459", "car", "average"
ren't found in the header (fields present in table: ["2976", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "450", "bike", "average
ren't found in the header (fields present in table: ["2977", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "463", "car", "average"
ren't found in the header (fields present in table: ["2978", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "460", "bike", "average
ren't found in the header (fields present in table: ["2979", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "470", "bike", "high",
ren't found in the header (fields present in table: ["2980", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "444", "truck", "low",
ren't found in the header (fields present in table: ["2981", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "436", "car", "low", "3
ren't found in the header (fields present in table: ["2982", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "397", "car", "low", "3
ren't found in the header (fields present in table: ["2983", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "368", "bike", "average
ren't found in the header (fields present in table: ["2984", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "137", "car", "low", "1
ren't found in the header (fields present in table: ["2985", "BHMEURBRD01", "470", "26.14901995", "91.7395035", "211", "car", "low", "1
```

```
# Step 5: Load Output for Visualization
# Step 4: Pathway Streaming Pipeline (output part only)
# Step 4: Pathway Streaming Pipeline (output part only)
try:
    # Debug: Inspect table content
    def on_change(row, time, is_addition):
        print(f"Pipeline output - Row: {row}, Time: {time}, Is addition: {is_addition}")
    pw.io.subscribe(table, on_change)

    # Write output to CSV
    pw.io.csv.write(table, 'output_prices.csv')
    print("Output configured to write to 'output_prices.csv'")

    # Verify input data exists
    import os
    if not os.path.exists('preprocessed_data.csv'):
        raise FileNotFoundError("Input 'preprocessed_data.csv' not found")
    input_df = pd.read_csv('preprocessed_data.csv')
    if input_df.empty:
        raise ValueError("Input 'preprocessed_data.csv' is empty")
    print("Input data verified. Sample:\n", input_df.head())

    # Run the pipeline with timeout
    import time
    pw.run(monitoring_level=pw.MonitoringLevel.NONE, max_backoff=10)
    print("Pathway pipeline executed successfully")
    time.sleep(3)  # Increased delay to ensure output is written
except Exception as e:
    print(f"Error in pipeline or output writing: {e}")
    raise
```

```
Error in pipeline or output writing: name 'pw' is not defined
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-2-190861661.py in <cell line: 0>()
      5     def on_change(row, time, is_addition):
      6         print(f"Pipeline output - Row: {row}, Time: {time}, Is addition: {is_addition}")
----> 7     pw.io.subscribe(table, on_change)
      8 
      9     # Write output to CSV

NameError: name 'pw' is not defined
```

Next steps: Explain error

```
# Step 6: Visualizations with Bokeh
lot_id = 'BHMBCCMKT01'
lot_data = output_df[output_df['parking_lot_id'] == lot_id].sort_values('timestamp')

p = figure(title=f"Price Trends for {lot_id}", x_axis_type="datetime",
           x_axis_label="Time", y_axis_label="Price ($)")
p.line(lot_data['timestamp'], lot_data['price_model1'], legend_label="Model 1", color="blue")
p.line(lot_data['timestamp'], lot_data['price_model2'], legend_label="Model 2", color="green")
p.line(lot_data['timestamp'], lot_data['price_model3'], legend_label="Model 3", color="red")
p.legend.click_policy = "hide"
show(p)

p2 = figure(title=f"Occupancy and Queue for {lot_id}", x_axis_type="datetime",
            x_axis_label="Time", y_axis_label="Count")
p2.line(lot_data['timestamp'], lot_data['occupancy'], legend_label="Occupancy", color="blue")
p2.line(lot_data['timestamp'], lot_data['queue_length'], legend_label="Queue Length", color="orange")
p2.legend.click_policy = "hide"
show(p2)
```

# Step 7: Report

""""

# Dynamic Pricing Report

## Demand Function

- **Model 1**: Linear model: Price = 10 + 0.5 * (Occupancy/Capacity).

- **Model 2**: Demand = 0.4 * Occupancy/Capacity + 0.3 * Norm(QueueLength) - 0.1 * Norm(Traffic) + 0.1 * IsSpecialDay + 0.1 * VehicleWeight. Price = 10 * (1 + 0.5 * Norm(Demand)).
- **Model 3**: Adjusts Model 2 price with competitor prices and reroutes if occupancy > 90% and competitors are cheaper.

## Assumptions

- Competitor prices simulated ($8-12$) due to missing data.
- Vehicle weights: cycle=0.5, bike=0.75, car=1.0, truck=1.5.
- Prices bounded between $5and20$.
- Traffic levels: low=0, average=1, high=2.

## Price Changes

- Model 1: Simple, occupancy-driven.
- Model 2: Responsive to multiple demand factors.
- Model 3: Competitive, with rerouting suggestions.

## Visualizations

- Price trends show Model 3's competitiveness.
- Occupancy and queue plots highlight demand peaks. """