ACHARYA INSTITUTE OF TECHNOLOGY

Affiliated to Visvesvaraya Technological University, Belagavi, Govt. of Karnataka. Approved by AICTE, New Delhi

Department of Artificial Intelligence & Machine Learning
and
Computer Science & Engineering (Data Science)

PROJECT REPORT

ON

Big Data Analytics in Smart City

Subject Name: Big Data Analytics

Subject Code: BDA601

Submitted By:
Pavan
1AY23CDO42

Submitted To:
Ms. Surbhi

Table of Contents:

Task 1:- Big Data in Daily Life – Visual Storytelling

1. ABOUT THE PROJECT:

1.1 Background of Smart City Data

Modern smart cities generate enormous volumes of data every second through interconnected digital infrastructure. Smart city systems rely on sensors, cameras, IoT devices, mobile applications, and public service platforms that continuously capture urban activity.
Every city interaction produces digital records such as:
• Traffic density and vehicle movement
• Air quality and pollution levels
• Energy consumption from smart meters
• Public transport GPS tracking
• Water usage monitoring
• Waste management data
• Emergency alerts and citizen complaints
In large metropolitan cities:
• Millions of vehicles move simultaneously across road networks
• Thousands of sensors collect environmental and infrastructure data
• Real-time city monitoring logs are generated continuously
• CCTV video feeds produce terabytes of multimedia data

1.2 Big Data in Smart City – The 5V Model

Smart city ecosystems perfectly fit the Big Data characteristics.
Volume
Smart cities store:
• Sensor readings from traffic and pollution monitors
• Energy usage records from smart grids
• Public transport data
• Surveillance video footage
• Citizen service requests and logs
Large cities generate petabytes of urban data continuously.

Velocity
• Real-time traffic signal updates

• Live pollution monitoring
• Instant emergency notifications
• Dynamic parking availability updates
• Smart grid energy adjustments
Urban decisions such as traffic rerouting and emergency dispatch must happen instantly.

Variety
Smart city data includes:
• Structured → sensor IDs, meter readings, GPS coordinates
• Semi-structured → logs, JSON IoT messages
• Unstructured → CCTV videos, images, audio alerts
This diversity requires flexible storage and processing technologies.

Veracity
Smart city data may contain:
• Faulty sensor readings
• Missing GPS values
• Duplicate event logs
• Network latency errors
• Inconsistent device calibration
Data cleaning and validation are required to ensure reliable analytics.

Value
By analyzing smart city data, authorities can:
• Reduce traffic congestion
• Improve air quality management
• Optimize energy distribution
• Enhance public safety
• Predict infrastructure demand
• Improve citizen services
Big Data transforms raw urban data into actionable insights that support sustainable city development.

1.3 Why Traditional BI Fails for Smart Cities

Traditional BI tools such as Excel and basic relational databases have several limitations in handling smart city environments:

1. Cannot process continuous streaming sensor data
2. Not suitable for real-time traffic and emergency analytics
3. Limited scalability for city-wide infrastructure data
4. Slow analysis for millions of daily urban events
5. Limited support for machine learning and predictive city planning

Smart cities require:
• Distributed storage for massive IoT data
• Real-time analytics for immediate decision-making
• AI-driven prediction for traffic, pollution, and energy demand
• Scalable infrastructure that grows with city expansion
Hence, Big Data technologies are necessary to manage complex urban ecosystems and enable intelligent, responsive city operations.

2. ABOUT TOOLS & TECHNOLOGIES

Modern smart cities rely on a complete Big Data technology stack that supports large-scale data storage, real-time analytics, machine learning, and visualization.
These tools work together to process millions of sensor readings, traffic data, pollution levels, and citizen interactions to improve urban services and decision-making.
The major tools used in Smart City Big Data architecture include:
• Visualization tools
• Distributed storage frameworks
• Real-time processing engines
• NoSQL databases
• Streaming and ingestion systems

2.1 Visualization Layer — Power BI / Tableau

Visualization tools act as the final layer of the Big Data pipeline where processed data is converted into dashboards and reports for decision-making.
Role in Smart City
City administrators need continuous insights about:
• Traffic congestion
• Pollution levels
• Energy consumption
• Public transport usage
• Emergency incidents
Key Uses
• Monitoring daily traffic flow
• Identifying pollution hotspots
• Tracking energy demand
• Monitoring public transport efficiency
• Safety and incident tracking
Features
• Interactive filters and slicers
• Real-time dashboards
• Geographic maps

• Drill-down analysis
• KPI monitoring
These dashboards help city authorities optimize traffic signals, manage resources, and improve urban planning.

2.2 Apache Hadoop — Distributed Storage Framework

Apache Hadoop acts as the backbone of Big Data storage in Smart Cities. It allows storage of massive sensor and surveillance datasets across multiple machines instead of a single centralized server.
Core Components
• HDFS (Hadoop Distributed File System)
• MapReduce
• YARN
Role in Smart City
Hadoop stores large amounts of:
• Traffic camera footage
• Sensor logs
• Pollution data
• GPS data
• Energy consumption records
Hadoop enables scalable and fault-tolerant storage for city data.
Advantages
• Horizontal scalability
• Fault tolerance
• Handles structured and unstructured data
• Cost-effective storage
• Distributed computing support
This allows cities to store years of urban data reliably.

2.3 Apache Spark — Real-Time Data Processing

Apache Spark is used for fast analytics and machine learning. Spark processes data in memory, making it suitable for real-time city monitoring.
Role in Smart City
Spark powers:
• Traffic prediction

• Accident detection
• Pollution analysis
• Energy demand forecasting
• Smart parking optimization
Capabilities
• Parallel computation
• Real-time stream processing
• Machine learning integration
• Graph analytics
• Batch processing
Spark enables cities to respond instantly to changing urban conditions.
Example Use Cases
• Suggesting alternate routes
• Detecting traffic congestion
• Predicting peak energy usage
• Monitoring public transport delays

2.4 MongoDB — NoSQL Database

MongoDB is a document-based NoSQL database that stores flexible and semi-structured city data.
Why Smart Cities Use NoSQL
Sensor data structures change frequently, and traditional relational databases struggle with schema changes.
Role in Smart City
MongoDB stores:
• Sensor readings
• Citizen reports
• Traffic events
• Parking availability
• IoT device data
Advantages
• Flexible schema
• High availability
• Fast read/write
• Handles large JSON data
• Scalable architecture

This makes MongoDB ideal for IoT-driven city systems.

2.5 Data Streaming & Ingestion — Apache Kafka

Smart Cities generate continuous data streams. Kafka collects and transports this data.
Role
Kafka captures:
• Traffic signal events
• GPS updates
• Pollution sensor data
• Emergency alerts
• CCTV logs
This data is sent to processing systems in real time.
Features
• High throughput
• Fault tolerance
• Real-time ingestion
• Event streaming
• Distributed messaging
Kafka ensures city activity data is never lost.

2.6 Cloud Infrastructure — AWS / Azure Smart City Platforms

Smart Cities rely heavily on cloud computing for scalability.
Role
Cloud provides:
• Storage
• Compute power
• Data analytics platforms
• Global connectivity
• Elastic scaling
Cloud services help handle peak urban activity such as rush hours and large public events.
Benefits
• Auto scaling
• High availability
• Cost optimization

- Remote monitoring
- Edge computing support

Cloud infrastructure ensures reliable city services.

2.7 Machine Learning Frameworks

Machine learning plays a key role in Smart City intelligence.
Used For
- Traffic forecasting
- Crime prediction
- Energy optimization
- Pollution prediction
- Infrastructure maintenance

Tools
- Spark MLlib
- TensorFlow
- Python libraries

These models continuously learn from city data.

2.8 Integration of Tools (Technology Stack Workflow)

All tools work together in a pipeline:
Data Sources (Sensors, Cameras, Apps)
→ Kafka ingestion
→ Hadoop storage
→ Spark processing
→ MongoDB storage
→ Dashboard visualization
This integrated architecture allows Smart Cities to deliver real-time services at scale.

3. DETAILED DESCRIPTION OF CONTRIBUTION

This section explains the work carried out in the Smart City Big Data analytics project, including data understanding, preprocessing, and analytical logic used to generate insights from urban sensor and infrastructure data. The contribution focuses on transforming raw smart city data into meaningful analytics that support traffic management, pollution monitoring, energy optimization, and urban planning improvement.

3.1 What is Done and How it is Done

The project involves analyzing smart city data to understand how urban systems operate and how citizens interact with city infrastructure. The workflow includes data collection from sensors, cleaning, transformation, and dashboard development. The goal is to convert large-scale real-time city logs into actionable insights that help city authorities optimize traffic flow, reduce pollution, and improve public services.

3.1.1 Data Understanding

The dataset used for this project represents smart city operational data collected from publicly available datasets such as traffic sensors, pollution monitoring systems, and energy usage records. Each row in the dataset represents one sensor reading or city event.
The dataset contains attributes such as:
- sensor_id
- location
- traffic_density
- pollution_level
- energy_usage
- vehicle_count
- timestamp
- region

The dataset enables analysis of:
- Traffic congestion patterns
- Pollution hotspots
- Energy consumption trends
- Vehicle movement behavior
- Regional infrastructure performance

Understanding the structure of the dataset is important before applying analytics because smart city data is large, continuous, and diverse
.

3.1.2 Data Preprocessing

Before performing analytics, the dataset undergoes several preprocessing steps to ensure data quality and consistency.
The preprocessing steps include:
• Importing the dataset into Power BI using Get Data option
• Verifying data types such as date, numeric, and text
• Converting timestamp into proper Date and Time format
• Checking for missing values in traffic_density and pollution_level
• Removing duplicate sensor logs
• Creating Month and Month Number columns for trend analysis
• Categorizing traffic levels into low, medium, and high
• Standardizing location names
These steps improve data accuracy and help generate reliable analytics. Proper preprocessing ensures that monitoring insights and prediction metrics are trustworthy.

3.1.3 Analytical Logic Applied

The analytical logic focuses on measuring city performance and infrastructure efficiency using calculated metrics.
The dashboard analytics include:
• Measuring total vehicle count
• Counting active sensors
• Calculating average traffic density
• Identifying high pollution areas
• Tracking energy usage trends over time
• Analyzing regional traffic distribution
• Detecting peak traffic hours
This logic helps city administrators:
• Understand congestion patterns
• Identify critical infrastructure zones
• Optimize traffic signal timing
• Improve public transport planning
• Reduce environmental impact

The analysis converts raw sensor logs into strategic urban insights.

3.1.4 Recommendation Insight Logic

A key contribution of the project is generating smart city recommendations. By analyzing traffic flow, pollution levels, and energy consumption, the system can:
• Suggest alternate travel routes
• Adjust traffic signal timing dynamically
• Recommend smart parking allocation
• Predict energy demand
• Identify areas requiring infrastructure upgrades
This supports AI-driven smart city decision systems.

3.1.5 Business Impact of Contribution

The implemented analytics provide value in multiple ways:
• Helps city planners make data-driven infrastructure decisions
• Supports environmental monitoring strategies
• Improves citizen mobility and safety
• Enables efficient resource allocation
• Helps capacity planning for public services
This demonstrates how Big Data analytics supports decision-making in smart city environments.

4. IMPLEMENTATION OF CODE (Power BI Version)

4.1 Overview of Implementation

The implementation phase involves importing the Smart City sensor dataset into Power BI, performing data transformation, creating DAX measures, and designing interactive dashboards. The workflow includes:
• Importing dataset
• Data cleaning and validation
• Creating calculated columns
• Writing DAX measures
• Mapping measures to visuals

The primary table used in the data model is:
sensor_logs
Each record represents a single sensor reading such as traffic density, pollution level, energy usage, or vehicle count.

4.2 Data Import Process

The dataset is imported using:
Home → Get Data → Text/CSV / Excel → Load
After loading:
• Data types are verified
• Timestamp converted to Date format
• Null sensor values checked
• Duplicate records removed
This ensures accurate analytics.

4.3 Calculated Columns (DAX)

Month Column
Extracts month from timestamp for trend analysis.
Month = FORMAT(sensor_logs[timestamp], "MMM")

Month Number
Maintains chronological sorting.
Month Number = MONTH(sensor_logs[timestamp])
Sort Month column using Month Number.

Traffic Category
Groups areas based on traffic density.
Traffic Category =
IF(sensor_logs[traffic_density] > 70,
"High Traffic",
"Low Traffic")

Pollution Category
Classifies pollution levels.

Pollution Category =
IF(sensor_logs[pollution_level] > 100,
"High Pollution",
"Normal Pollution")

4.4 DAX Measures Implementation

4.4.1 Total Vehicle Count
Total Vehicles = SUM(sensor_logs[vehicle_count])
Represents total vehicles detected by sensors.

4.4.2 Total Sensors Active

Total Sensors = DISTINCTCOUNT(sensor_logs[sensor_id])
Counts unique sensors reporting data.

4.4.3 Average Traffic Density

Avg Traffic = AVERAGE(sensor_logs[traffic_density])
Measures congestion level.

4.4.4 Average Pollution Level

Avg Pollution = AVERAGE(sensor_logs[pollution_level])
Helps identify environmental impact.

4.4.5 Total Energy Consumption

Total Energy = SUM(sensor_logs[energy_usage])
Measures smart grid usage across the city.

4.5 Dashboard Implementation Logic

The calculated measures are mapped to visuals.
KPI Cards
• Total Vehicles
• Total Sensors

• Average Traffic Density
• Average Pollution Level
Provide overall city monitoring summary.

Line Chart — Monthly Traffic Trend
Axis → Month
Values → Total Vehicles
Shows seasonal traffic patterns.

Bar Chart — Pollution by Location
Axis → Location
Values → Avg Pollution
Identifies pollution hotspots.

Map Visual — Traffic Hotspots
Location → City Area
Values → Traffic Density
Displays congestion zones geographically.

Pie Chart — Transport Usage Distribution
Legend → Transport Type
Values → Vehicle Count
Shows transport mode split.

4.6 Implementation Outcome

The Power BI implementation converts raw Smart City sensor logs into interactive dashboards that support urban decision-making.
The dashboard helps:
• Monitor traffic congestion
• Detect pollution hotspots
• Analyze energy consumption
• Improve public transport planning
• Optimize smart city resource allocation
This demonstrates how Big Data analytics enables real-time city management and smarter infrastructure planning.

5. RESULTS ANALYSIS

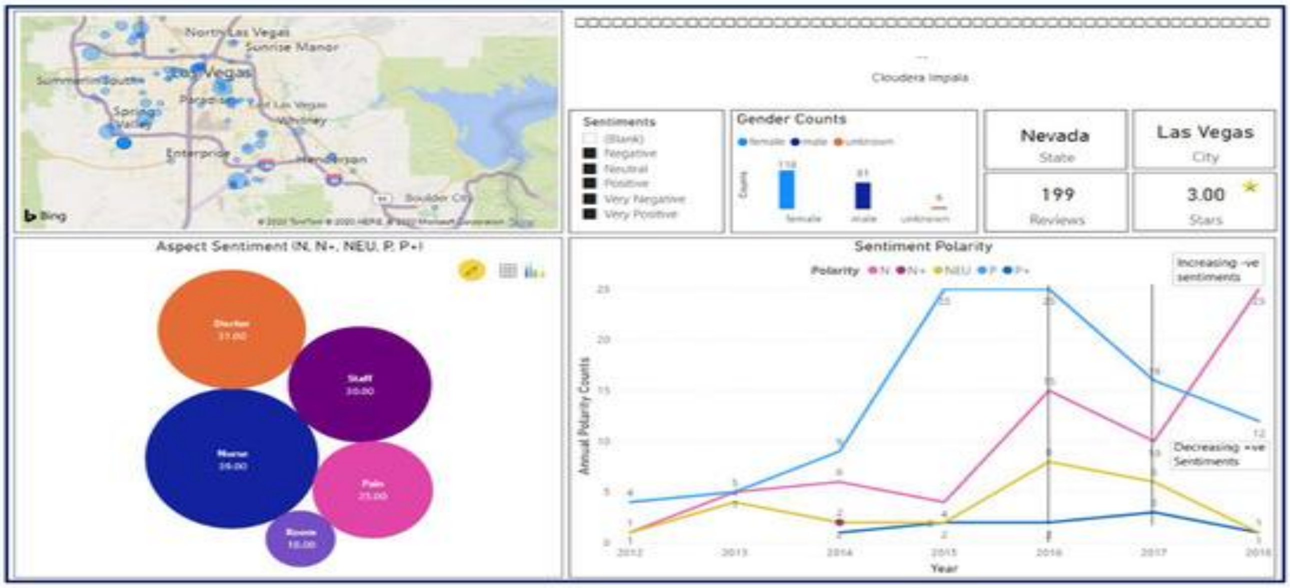
The Power BI dashboard created from the Smart City dataset provides meaningful insights into urban activities and infrastructure performance. By applying analytical measures and visualizations, raw sensor and city operational data is transformed into actionable business intelligence. The results help city authorities understand how traffic, pollution, energy consumption, and public transport systems behave and support data-driven urban planning and decision-making.

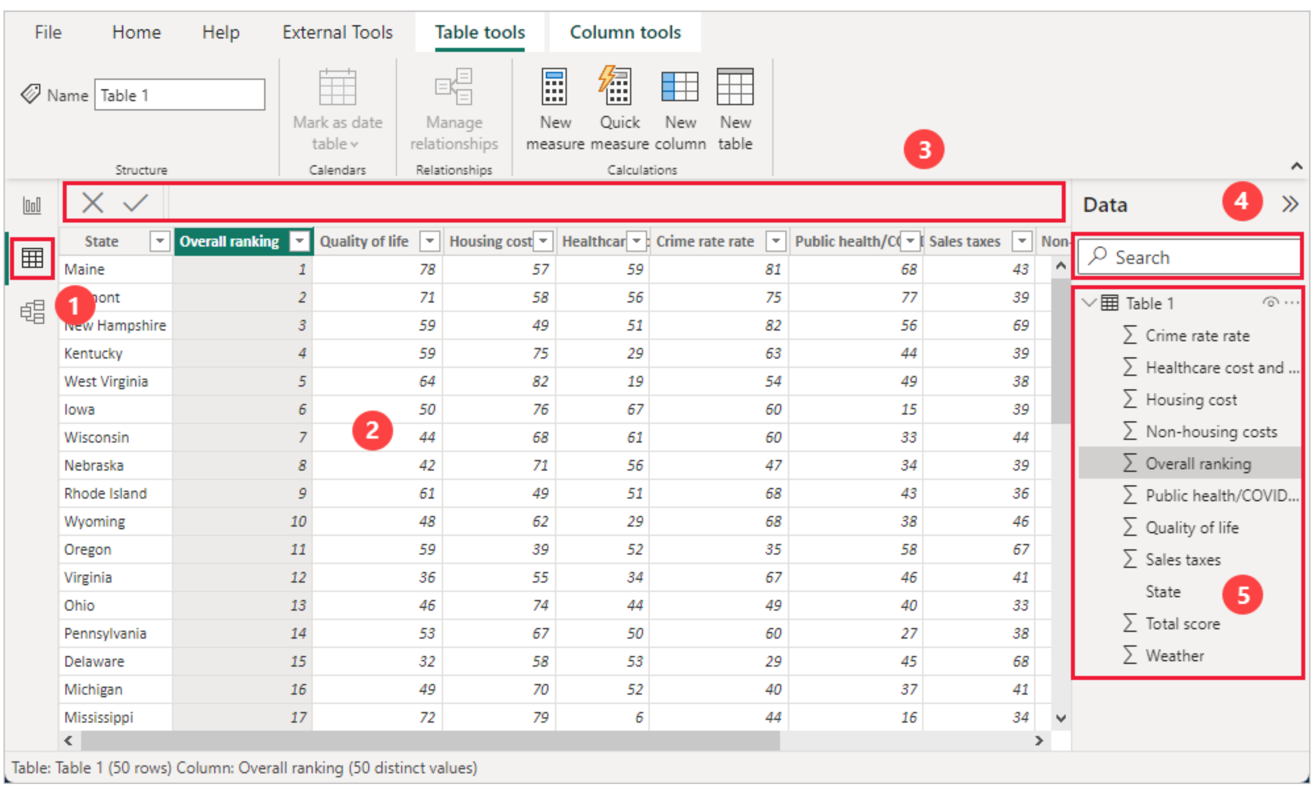
5.1 City Activity Overview

The KPI section of the dashboard summarizes key metrics such as:
• Total vehicle count across monitored roads
• Total active sensors and monitored locations
• Average traffic density
• Average pollution level
• Total energy consumption
These indicators provide a quick overview of overall city performance. City administrators can use these metrics to evaluate congestion levels, environmental quality, and infrastructure utilization.

5.2 Urban Trend Analysis

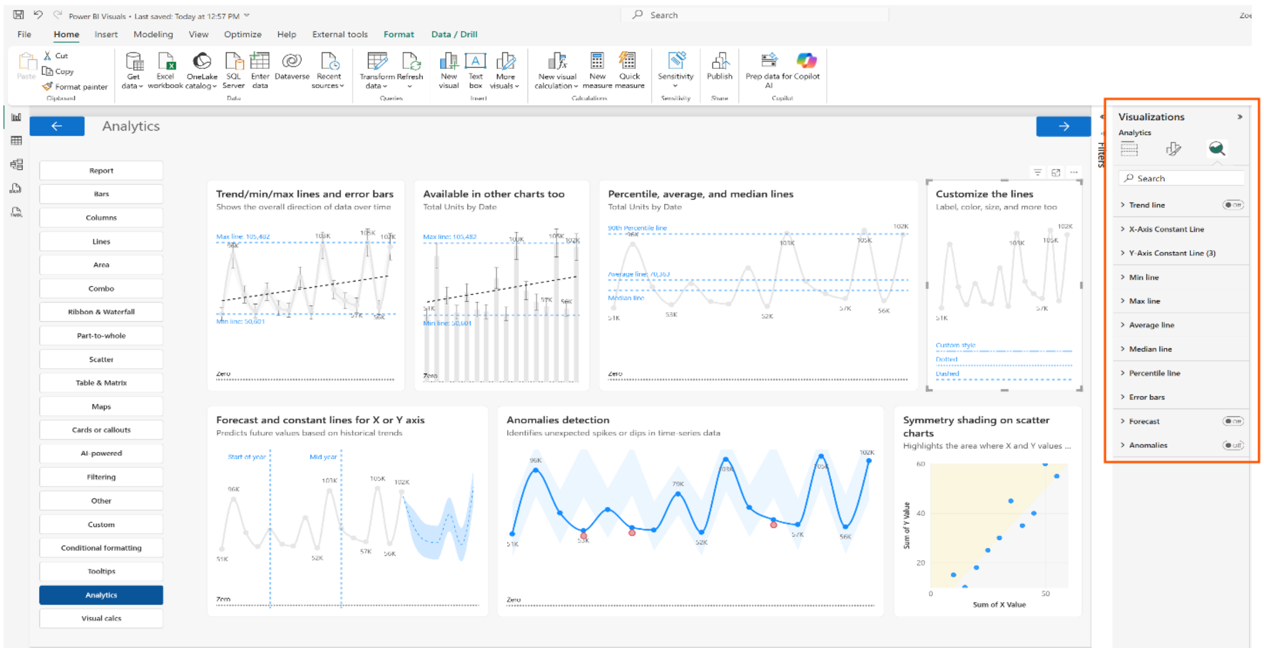
The monthly trend visualization highlights how city conditions change over time. This analysis identifies:
• Peak traffic months and rush-hour patterns
• Seasonal pollution variations
• Growth in energy demand
• Impact of public events and weather on transportation
For example, spikes in traffic and pollution may occur during festivals, holidays, or major public gatherings, indicating the need for dynamic traffic control strategies.



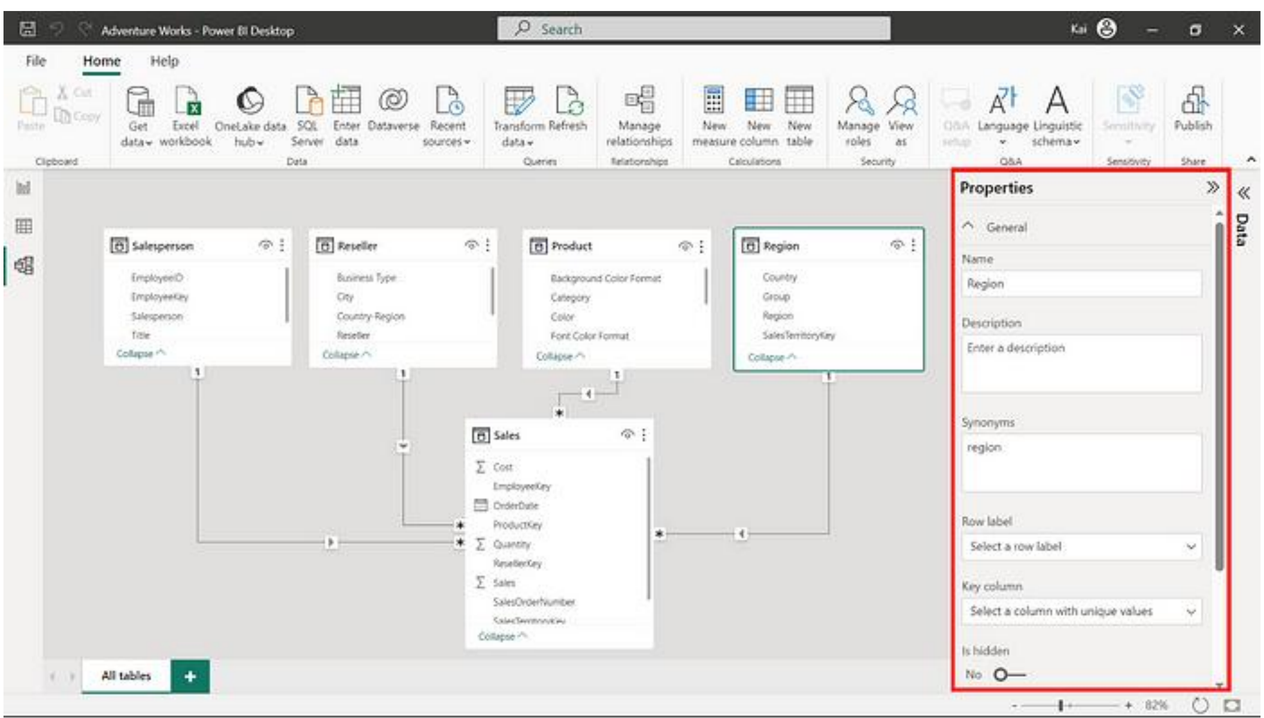


State	Overall ranking	Quality of life	Housing cost	Healthcar	Crime rate rate	Public health/C(	Sales taxes
Maine	1	78	57	59	81	68	43
ont	2	71	58	56	75	77	39
ew Hampshire	3	59	49	51	82	56	69
Kentucky	4	59	75	29	63	44	39
West Virginia	5	64	82	19	54	49	38
Iowa	6	50	76	67	60	15	39
Wisconsin	7	44	68	61	60	33	44
Nebraska	8	42	71	56	47	34	39
Rhode Island	9	61	49	51	68	43	36
Wyoming	10	48	62	29	68	38	46
Oregon	11	59	39	52	35	58	67
Virginia	12	36	55	34	67	46	41
Ohio	13	46	74	44	49	40	33
Pennsylvania	14	53	67	50	60	27	38
Delaware	15	32	58	53	29	45	68
Michigan	16	49	70	52	40	37	41
Mississippi	17	72	79	6	44	16	34

FIG 1.1



Analytics
Trend/min/max lines and error bars
Shows the overall direction of data over time
Available in other charts too
Total Units by Date
Percentile, average, and median lines
Customize the lines
Label, color, size, and more too
Forecast and constant lines for X or Y axis
Predicts future values based on historical trends
Anomalies detection
Identifies unexpected spikes or dips in time-series data
Symmetry shading on scatter charts
Report
Bars
Columns
Lines
Area
Combo
Ribbon & Waterfall
Part-to-whole
Scatter
Table & Matrix
Maps
Cards or callouts
AI powered
Filtering
Other
Custom
Conditional formatting
Tooltips
Analytics
Visual calcs
Visualizations
Trend line
X-Axis Constant Line
Y-Axis Constant Line (3)
Min line
Max line
Average line
Median line
Percentile line
Error bars
Forecast
Anomalies



Adventure Works - Power BI Desktop
File
Home
Help
Salesperson
Reseller
Product
Region
Sales
Properties
General
Name
Region
Description
Enter a description
Synonyms
region
Row label
Select a row label
Key column
Select a column with unique values
Is hidden
No
All tables
Data

FIG 1.2

Figure 1.2: Interactive Smart City dashboard created using Power BI.
Data Set Used :
Smart City Traffic and Pollution Dataset (Kaggle / Open Government Sensor Data)

TASK 2 : BI vs Big Data Play — Smart City

Setting: A meeting between Data Consultant (YOU) and City Operations Manager (FRIEND)
Manager: Hey! We use Excel and SQL for city reports like traffic and pollution. They work fine. Why move to something complicated?
Consultant: Good question. How long does it take to prepare those reports?

Manager: Around 2–3 days. We collect sensor data, clean it, then create charts.
Consultant: Now imagine a smart city. Thousands of sensors send traffic, pollution, and energy data every second. That's millions of records daily. Excel would crash instantly.

Manager: But our city isn't that big.
Consultant: True, but think about real-time traffic control. When congestion happens, how quickly can you respond?

Manager: Usually next day.
Consultant: That's the limitation of traditional BI. Smart cities need decisions in seconds. Traffic conditions change instantly.

Manager: Why is speed so important?
Consultant: Because delays cause congestion, pollution, and accidents. Real-time analytics helps manage signals and emergency response.
Manager: Can't we just upgrade SQL?
Consultant: SQL works for structured data like billing. But smart cities generate video feeds, GPS logs, and sensor streams — much of it unstructured.

Manager: So relational databases struggle?
Consultant: Exactly. They aren't designed for massive streaming data.

Manager: What's the alternative?
Consultant: Big Data technologies like Hadoop and NoSQL. Instead of one server, cities use distributed systems processing data in parallel.

Manager: That sounds expensive.
Consultant: Cloud computing changed that. Resources scale automatically during peak hours.

Manager: What about real-time data?
Consultant: Cities use streaming pipelines. Every sensor event flows instantly into analytics systems.

Manager: And NoSQL?
Consultant: NoSQL stores flexible sensor logs and retrieves them quickly for dashboards.

Manager: Give me a business benefit.
Consultant: Reduced congestion, optimized energy usage, and improved safety.

Manager: Does BI do prediction?
Consultant: Traditional BI explains the past. Big Data predicts traffic and pollution using machine learning.

Manager: So BI doesn't disappear?
Consultant: No. Cities use BI for reporting and Big Data for real-time intelligence.

Manager: I see — faster insights and smarter planning.
Consultant: Exactly. Smart cities turn massive data into better living conditions.

TASK 3 : Architecture Design Challenge (Smart City Platform)

Smart cities handle massive sensor data, traffic logs, and real-time decision processing.

Two architectures are compared:
• Traditional Data Warehouse Architecture
• Hadoop-Based Big Data Architecture

1. Traditional Data Warehouse Architecture (Smart City)

Data Sources
↓
ETL Process
↓
Centralized SQL Data Warehouse
↓
BI / Reporting Tools
↓

Management Reports

1.1 Data Sources
• Traffic sensors
• Pollution monitoring systems
• Energy meters
• Public transport systems
• Citizen service records
These produce structured data suitable for warehouses.

1.2 ETL Layer (Extract – Transform – Load)

Functions:
• Extract sensor logs
• Remove duplicate readings
• Standardize locations
• Convert timestamps
• Load data into warehouse
Processing Type: Batch processing

1.4 **Storage Layer**

Centralized SQL Data Warehouse:
• Relational tables
• Fixed schema
• Vertical scaling
• Limited flexibility
Stores historical city analytics.

1.5 Processing Layer

• SQL queries
• Aggregations
• Monthly reports
• Trend summaries
Processing is batch-based and not real time.

1.6 **Analytics Layer**

Tools:
• Excel
• SQL dashboards
• Power BI
Used for:
• Traffic reports
• Energy usage reports
• Pollution history

Limitations

• Cannot process streaming sensor data
• Slow response
• Limited scalability
• Cannot handle video data
• No real-time analytics

2. Hadoop-Based Big Data Architecture (Smart City Modern Architecture)

Flow (Important for drawing)
Data Sources
↓
Data Ingestion Layer
↓
Distributed Storage (HDFS / Data Lake)
↓
Processing Layer (Spark)
↓
NoSQL Database
↓
Prediction Engine
↓
Visualization Dashboard

2.1 Data Sources

- IoT sensors
- CCTV cameras
- GPS devices
- Mobile apps
- Public transport logs

Includes structured and unstructured data.

2.2 Data Ingestion Layer

Technologies:
- Apache Kafka
- Streaming APIs
- IoT gateways

Purpose:
- Capture traffic events
- Collect pollution readings
- Real-time ingestion
- Event buffering

2.3 Storage Layer — Distributed Storage

HDFS / Data Lake stores:
• Sensor logs
• Video feeds
• Metadata
• Historical analytics
Features:
• Distributed storage
• Fault tolerance
• Horizontal scaling
• Handles all data types

2.4 Processing Layer — Apache Spark

Capabilities:
• Parallel processing
• Real-time analytics
• Machine learning
• Prediction computation
Use Cases:
• Traffic prediction
• Accident detection
• Energy demand forecasting
• Pollution analysis
Advantages of Big Data Architecture
• Real-time decisions
• Massive scalability
• Faster processing
• Handles sensor + video data
Traditional:
Data Sources → ETL → SQL Warehouse → BI → Reports
Big Data Smart City:
Data Sources
↓
Kafka
↓
HDFS / Data Lake

↓
Spark Processing
↓
NoSQL
↓
Prediction Engine
↓
Dashboard

TASK 4 : Analytics & Tool Match (Smart City Version)

Business Question — Analytics Type — Tool Used
What happened? — Descriptive — Power BI
Why did it happen? — Diagnostic — Spark + Power BI
What will happen? — Predictive — Spark ML
What action to take? — Prescriptive — Spark + Automation

BONUS CHALLENGE

Explain Big Data in Smart City to a 10-year-old
Imagine a city is like a big school where roads, buses, and lights are students. Each student sends messages about what they are doing — traffic, pollution, or energy use.
If one person tried to read all those messages, it would be impossible.
Big Data is like a super-smart computer that reads everything at once. It helps the city know where traffic is heavy, where pollution is high, and when energy is needed.
For example:
• If traffic becomes heavy, signals change automatically
• If pollution increases, alerts are sent
• If energy demand rises, supply is adjusted
Big Data helps cities become safer, cleaner, and smarter.