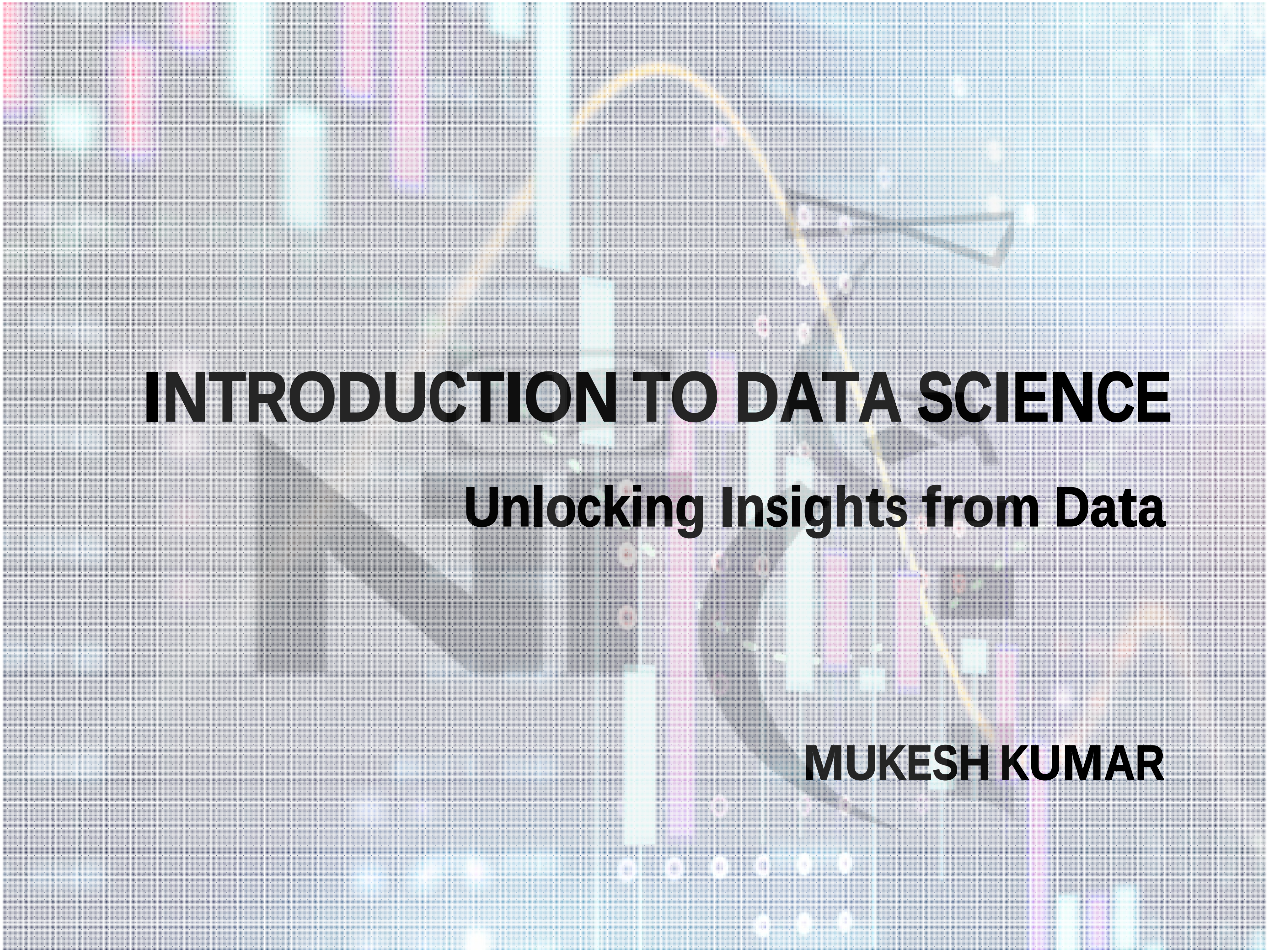


INTRODUCTION TO DATA SCIENCE

Unlocking Insights from Data

MUKESH KUMAR

AGENDA

- What is Data Science?
- Importance of Data Science
- Key Components of Data Science
- Data Science Workflow
- End to End Data Science project Demo
- Applications
- Q&A

What is Data Science?

- Data Science is the art and science of turning raw data into actionable insights to solve real-world problems

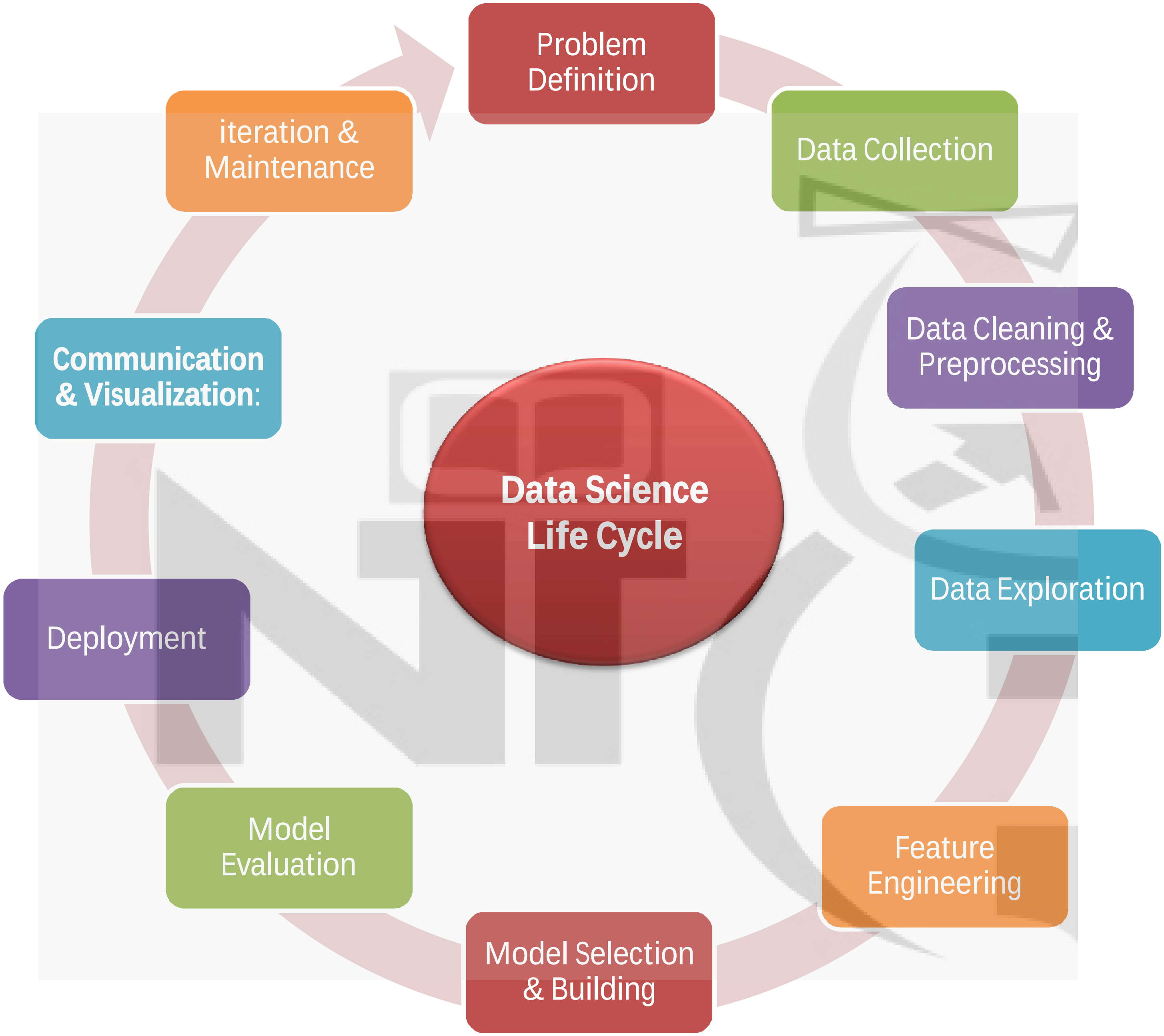
- Data science is an interdisciplinary field focused on extracting knowledge from typically large data sets and applying the knowledge and insights from that data to solve problems in a wide range of application domains.

Purpose of Data Science

- To uncover hidden patterns and trends in data.
- To support better decision-making across industries.
- To solve complex problems in fields like healthcare, finance, marketing, and technology.

Data Science Workflow

- Problem Definition
- Data Collection
- Data Cleaning
- Data Exploration
- Model Selection & Building
- Model Evaluation
- Deployment
- Iteration & Maintenance


Data Science Life Cycle
Problem Definition
Data Collection
Data Cleaning & Preprocessing
Data Exploration
Feature Engineering
Model Selection & Building
Model Evaluation
Deployment
Communication & Visualization:
iteration & Maintenance

Problem Definition

- **Objective**: Understand the business problem or research question.
- **Key Actions**:
  - Define goals and deliverables.
  - Identify the problem domain and expected outcomes.
  - Understand constraints (time, budget, data availability).

Data Collection

- **Objective**: Gather data required to solve the problem.
- **Key Actions**:
  - Identify data sources (databases, APIs, sensors, web scraping, etc.).
  - Collect relevant raw data from multiple sources.

Data Cleaning and Preprocessing

- **Objective**: Prepare the raw data for analysis and modeling.

- **Key Actions**:
  - Handle missing or inconsistent data.
  - Remove duplicates or irrelevant data points.
  - Normalize or standardize numerical values.
  - Encode categorical variables.

- **Tools**: Pandas, NumPy.

Exploratory Data Analysis (EDA)

- **Objective**: Understand the data and identify patterns, trends, and anomalies.

- **Key Actions**:
  - Generate visualizations (e.g., histograms, scatter plots, box plots).
  - Compute summary statistics (mean, median, mode, correlation).
  - Detect relationships and outliers.

- **Tools**: Matplotlib, Seaborn, Plotly.

Feature Engineering

- **Objective**: Create meaningful input features for the model.
- **Key Actions**:
  - Select important variables (feature selection).
  - Transform data (e.g., log transformations, scaling).
  - Create new features (e.g., time-based features, ratios).

Feature Engineering Example

- Sales Data for a product

Transaction Date	Sales
1/1/2025	120
1/2/2025	150
1/3/2025	200
1/4/2025	250
1/5/2025	100

Feature Engineering Example

Extract new features like:

- **Day of the week** (e.g., Monday, Tuesday) to capture weekly patterns.
- **Month** to identify seasonal trends.
- **Is Holiday** (binary: 1 for holidays, 0 otherwise) to account for holiday effects.

- **Purpose:** These features help models understand temporal trends affecting sales.

Feature Engineering Example

- Sales Data after Feature Engg

Transaction Date	Sales	Day of Week	Month	Is Holiday
1/1/2025	120	Wednesday	January	1
1/2/2025	150	Thursday	January	0
1/3/2025	200	Friday	January	0
1/4/2025	250	Saturday	January	1
1/5/2025	100	Sunday	January	0

Model Selection and Building

- **Objective**: Develop predictive or analytical models.

- **Key Actions**:
  - Choose appropriate algorithms (linear regression, decision trees, neural networks, etc.).
  - Split data into training, validation, and test sets.
  - Train models on the training set and tune hyperparameters.

- **Tools**: Scikit-learn, TensorFlow, PyTorch.

Model Evaluation

- **Objective**: Assess model performance and refine as needed.
- **Key Actions**:
  - Evaluate metrics (e.g., accuracy, precision, recall, F1-score, AUC-ROC).
  - Perform cross-validation to ensure model robustness.
  - Compare multiple models and select the best one.

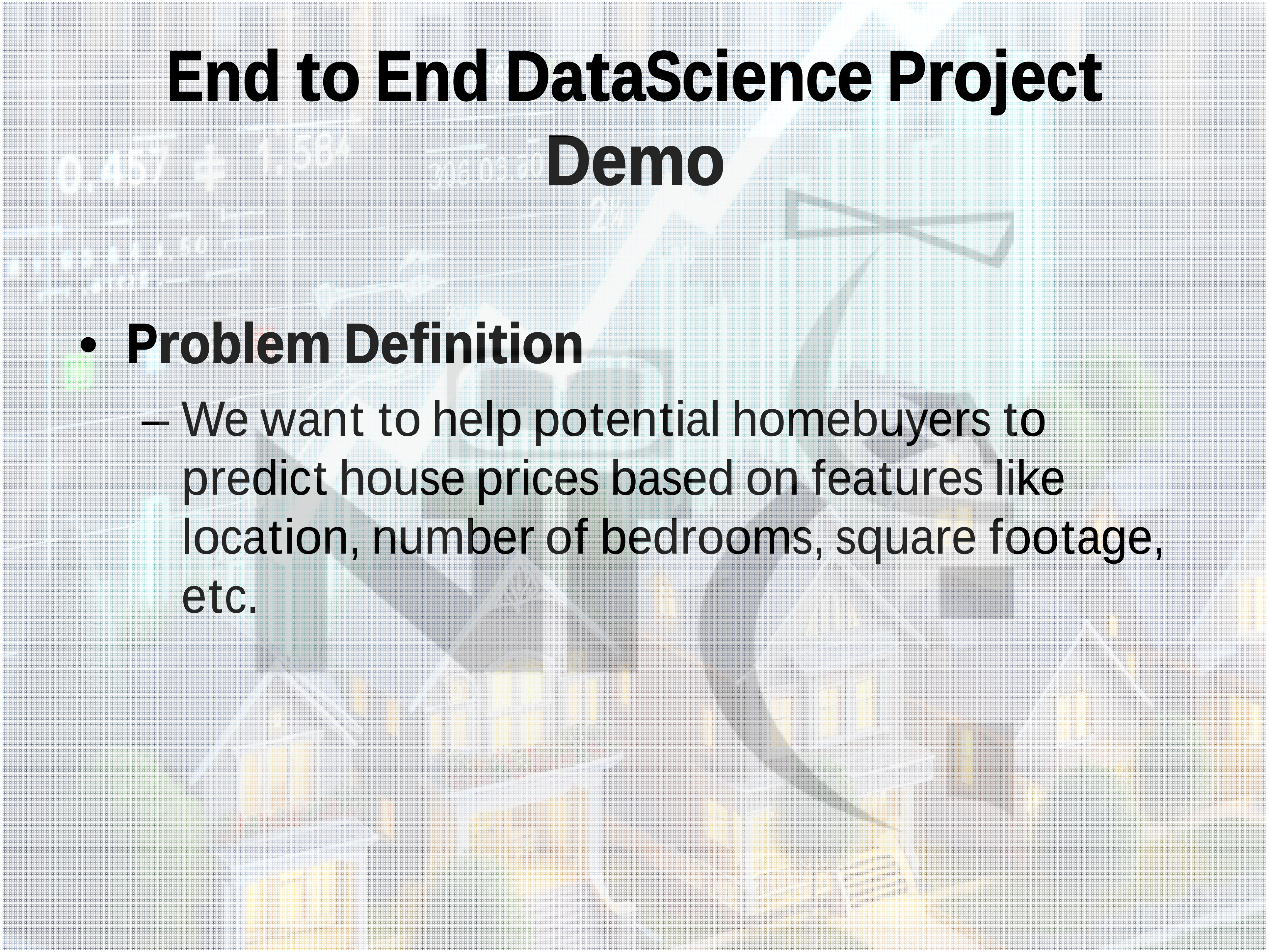
Deployment

- **Objective**: Integrate the final model into production.
- **Key Actions**:
  - Build APIs, dashboards, or applications for end-users.
  - Monitor real-time model performance.
  - Continuously update the model as new data becomes available.

Iteration and Maintenance

- **Objective**: Continuously improve and adapt the solution.
- **Key Actions**:
  - Gather feedback from users and stakeholders.
  - Update the model with new data or insights.
  - Monitor for data drift or performance degradation.

End to End DataScience Project Demo



End to End DataScience Project Demo

- **Problem Definition**
  - We want to help potential homebuyers to predict house prices based on features like location, number of bedrooms, square footage, etc.

Data Collection

- Include features like:
  - Location
  - Number of bedrooms
  - Square footage
  - Age of the house
  - Price (as the target variable).

House Dataset

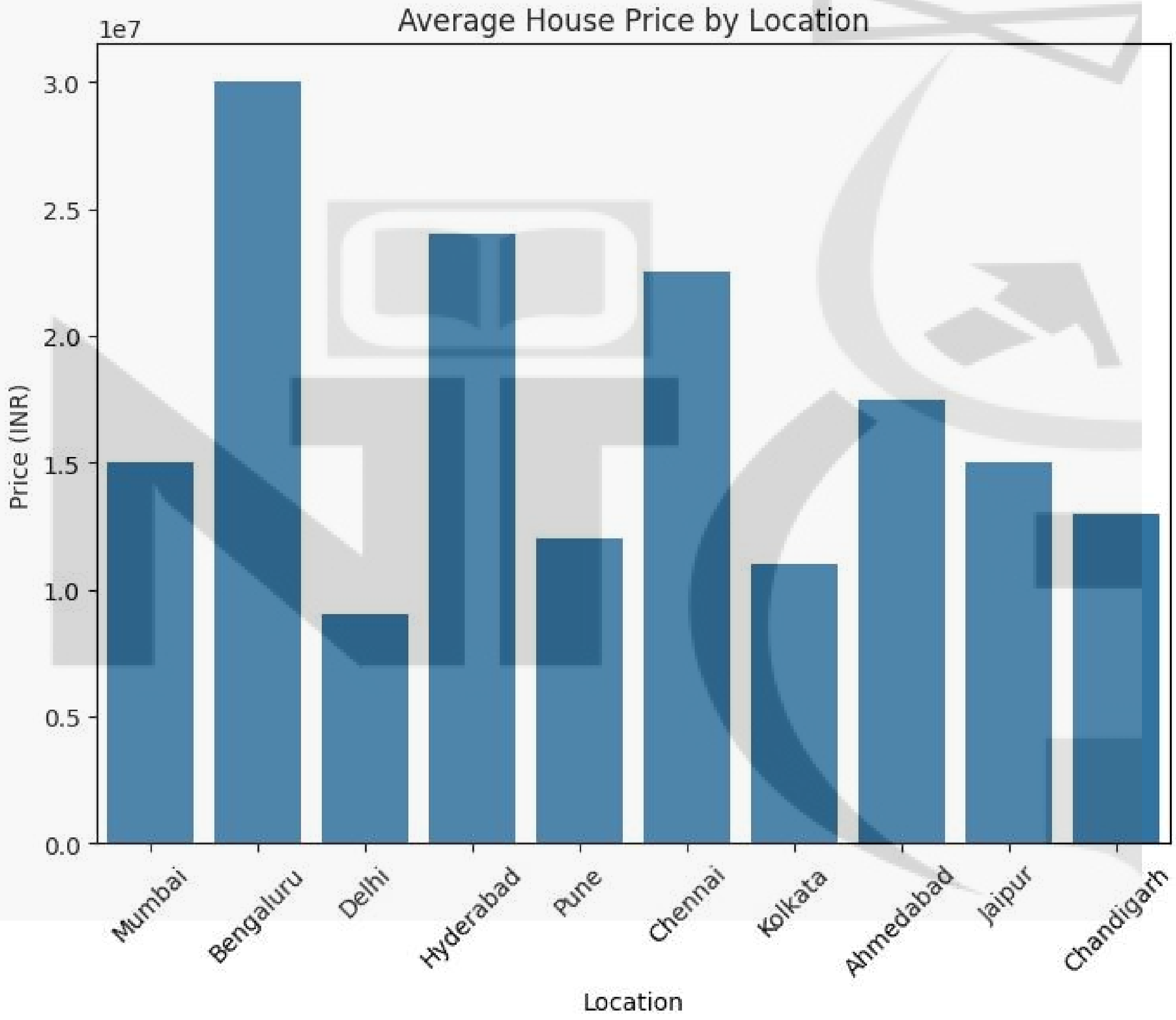
ID	Location	Bedrooms	Bathrooms	Square Footage	Age (Years)	Price (₹)
1	Mumbai	3	2	1200	10	1,50,00,000
2	Bengaluru	4	3	2500	5	3,00,00,000
3	Delhi	2	1	800	20	90,00,000
4	Hyderabad	5	4	3000	8	2,40,00,000
5	Pune	3	2	1500	12	1,20,00,000
6	Chennai	4	3	1800	7	2,25,00,000
7	Kolkata	2	1	1000	15	1,10,00,000
8	Ahmedabad	3	2	2000	6	1,75,00,000
9	Jaipur	4	3	2200	9	1,50,00,000
10	Chandigarh	3	2	1400	11	1,30,00,000

Data Cleaning

- Examples of handling missing data, removing duplicates, and dealing with outliers.

Exploratory Data Analysis (EDA)

- Create a few graphs or charts:
  - A bar chart showing the average price in different cities.
  - A scatterplot showing the relationship between square footage and price.


Average House Price by Location
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Price (INR)
Mumbai
Bengaluru
Delhi
Hyderabad
Pune
Chennai
Kolkata
Ahmedabad
Jaipur
Chandigarh
Location

Insights from the Graph

- **Highest Prices**:
  - **Bengaluru** has the highest average house prices, indicating a highly valued real estate market in this city.
- **Moderately High Prices**:
  - **Hyderabad** and **Chennai** follow Bengaluru, with moderately high average house prices, possibly due to their growing infrastructure and real estate demand.
- **Lower Prices**:
  - **Delhi** shows one of the lowest average house prices, contrary to expectations for a metropolitan area.
  - Other cities like **Kolkata**, **Ahmedabad**, **Jaipur**, and **Chandigarh** also have relatively lower average prices, making them more affordable.

Feature Engineering

- Creating a new feature: **Price per Square Foot**

Price_Per_SqFt :

- This feature will give us an idea about which city is costliest and which is affordable.

Feature Engineering

- Updated Dataset

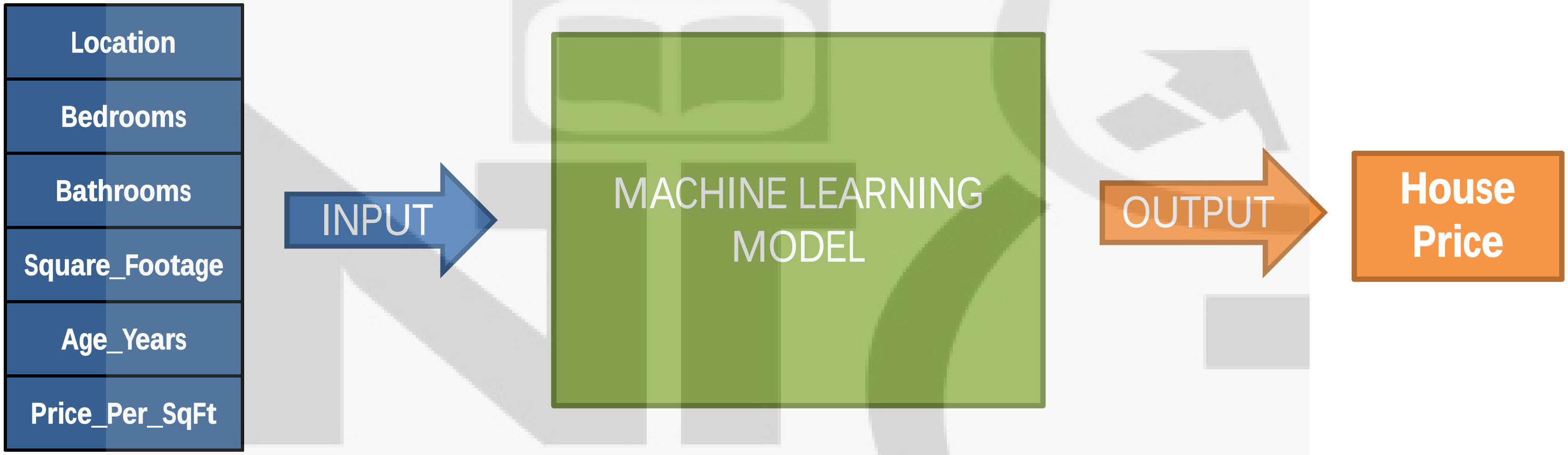
ID	Location	Bedrooms	Bathrooms	Square_Footage	Age_Years	Price_INR	Price_Per_SqFt
1	Mumbai	3	2	1200	10	15000000	12500
2	Bengaluru	4	3	2500	5	30000000	12000
3	Delhi	2	1	800	20	9000000	11250
4	Hyderabad	5	4	3000	8	24000000	8000
5	Pune	3	2	1500	12	12000000	8000
6	Chennai	4	3	1800	7	22500000	12500
7	Kolkata	2	1	1000	15	11000000	11000
8	Ahmedabad	3	2	2000	6	17500000	8750
9	Jaipur	4	3	2200	9	15000000	6818.181818
10	Chandigarh	3	2	1400	11	13000000	9285.714286

Insights from the new feature

- **Mumbai** ,**Delhi** and **Banglore** have the highest price per square foot, making them less affordable for large families.

- **Jaipur** offer more cost-efficient options for homebuyers.

Model Building



Model Deployment

- Once the mode is Evaluated and is perfroming as per requirement its deployed in pipeline and made available to users via app

Communication & Visualization

- **Communication:**
  In data science, communication involves effectively conveying findings, insights, and recommendations to stakeholders using clear language and context.

- **Visualization:**
  Visualization is the process of creating graphical representations of data and results to make complex patterns and insights easier to understand and interpret.

Maintenance

- Maintenance ensures the model stays accurate and reliable by updating it with new data, monitoring performance, and addressing any drift or inconsistencies over time.

Real-Life Applications

- Healthcare (predicting diseases)
- E-commerce (recommendation systems)
- Social Media (sentiment analysis)
- Finance (fraud detection)
- Transportation

Netflix Example

Netflix Example

Problem Statement:

- Netflix has millions of users watching movies and shows every day.
- Each user has different tastes.
- Netflix wants to recommend shows or movies so that each user will likely enjoy, keeping them engaged.

How Data Science Helps Netflix

- **Collecting Data:** Netflix gathers data about what each user watches, rates, searches, and how long they watch.

UserID	Name	Age	Gender	Movie/TV Show Watched	Rating	Search Queries	Watch Time (minutes)
1	Alice	28	Female	Inception, Breaking Bad	4.5	Sci-fi, Thriller, Action	120
2	Bob	35	Male	The Witcher, Stranger Things	5	Fantasy, Horror	180
3	Charlie	22	Male	The Crown, The Office	3	Drama, Comedy	90
4	Diana	40	Female	Friends, The Queen's Gambit	4.7	Comedy, Drama	150

How Data Science Helps Netflix

Processing Data:

- Data scientists organize this huge volume of data and identify patterns, like which types of shows are popular with specific groups of users.

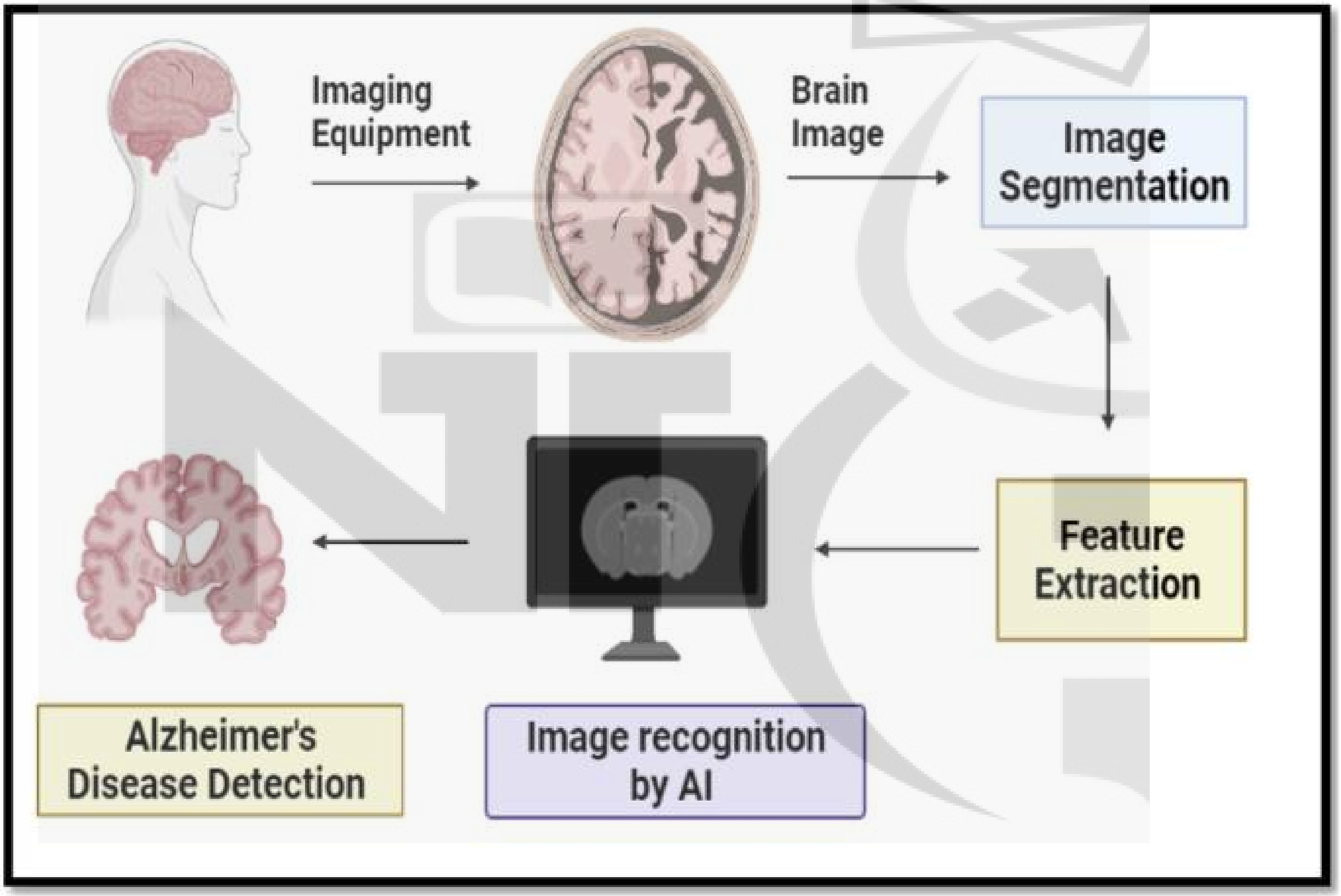
How Data Science Helps Netflix

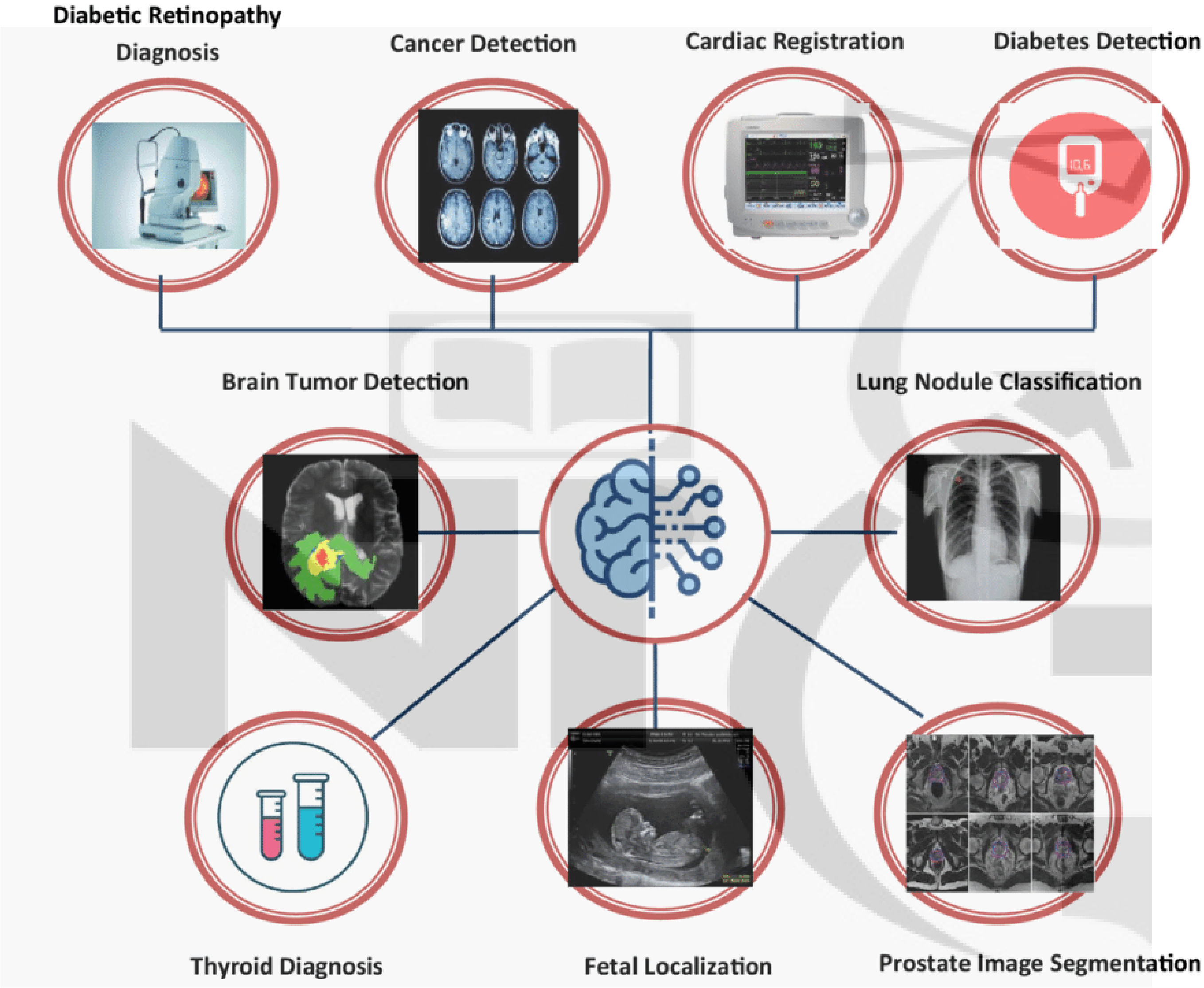
- **Building a Model:** Using machine learning, they create algorithms that predict what you might like based on your viewing history and similar users' preferences.

- **Delivering Insights:** The system suggests "Top Picks for You" on your Netflix homepage, making your experience personalized.

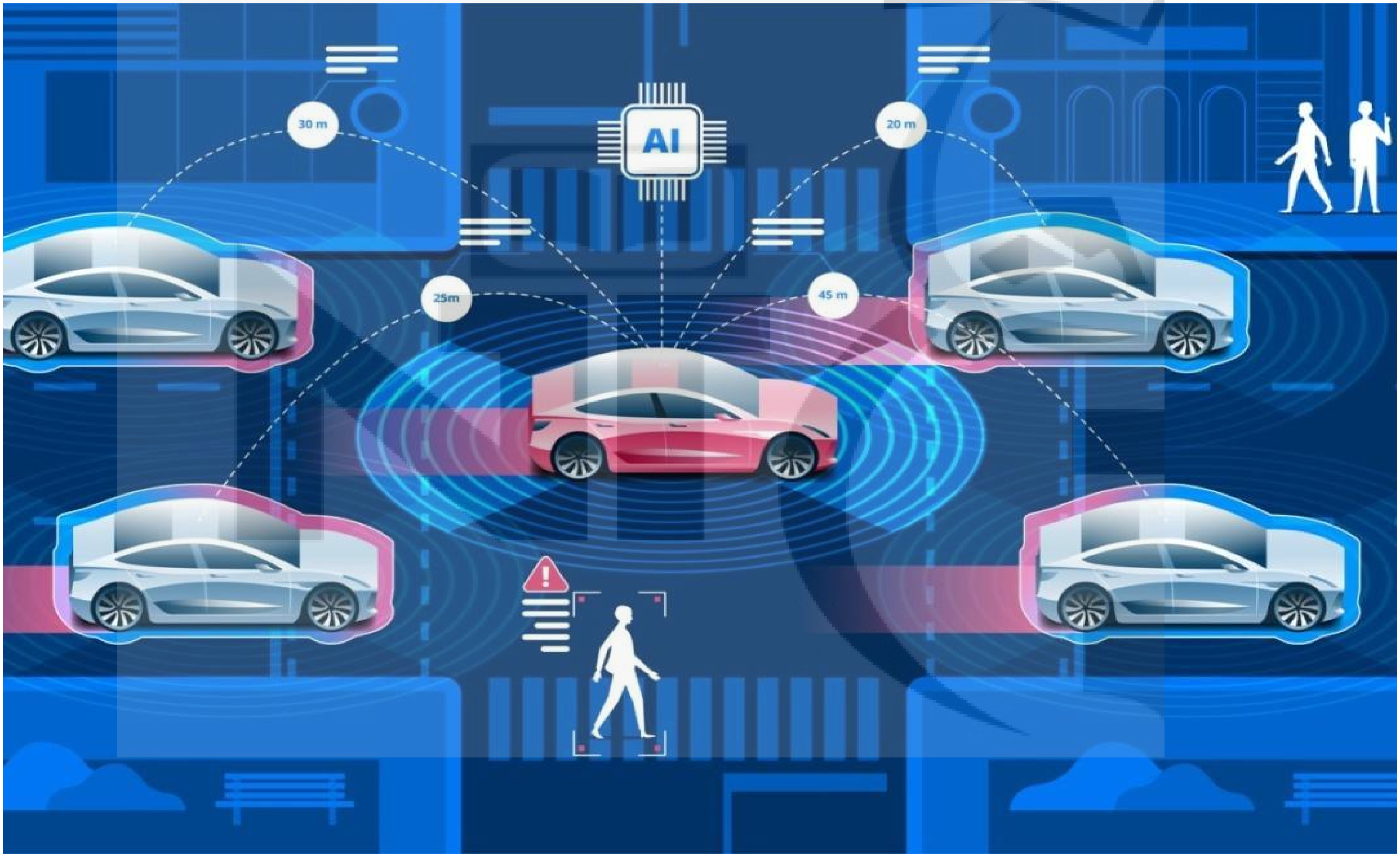
HealthCare

Alzheimer Disease Predictor

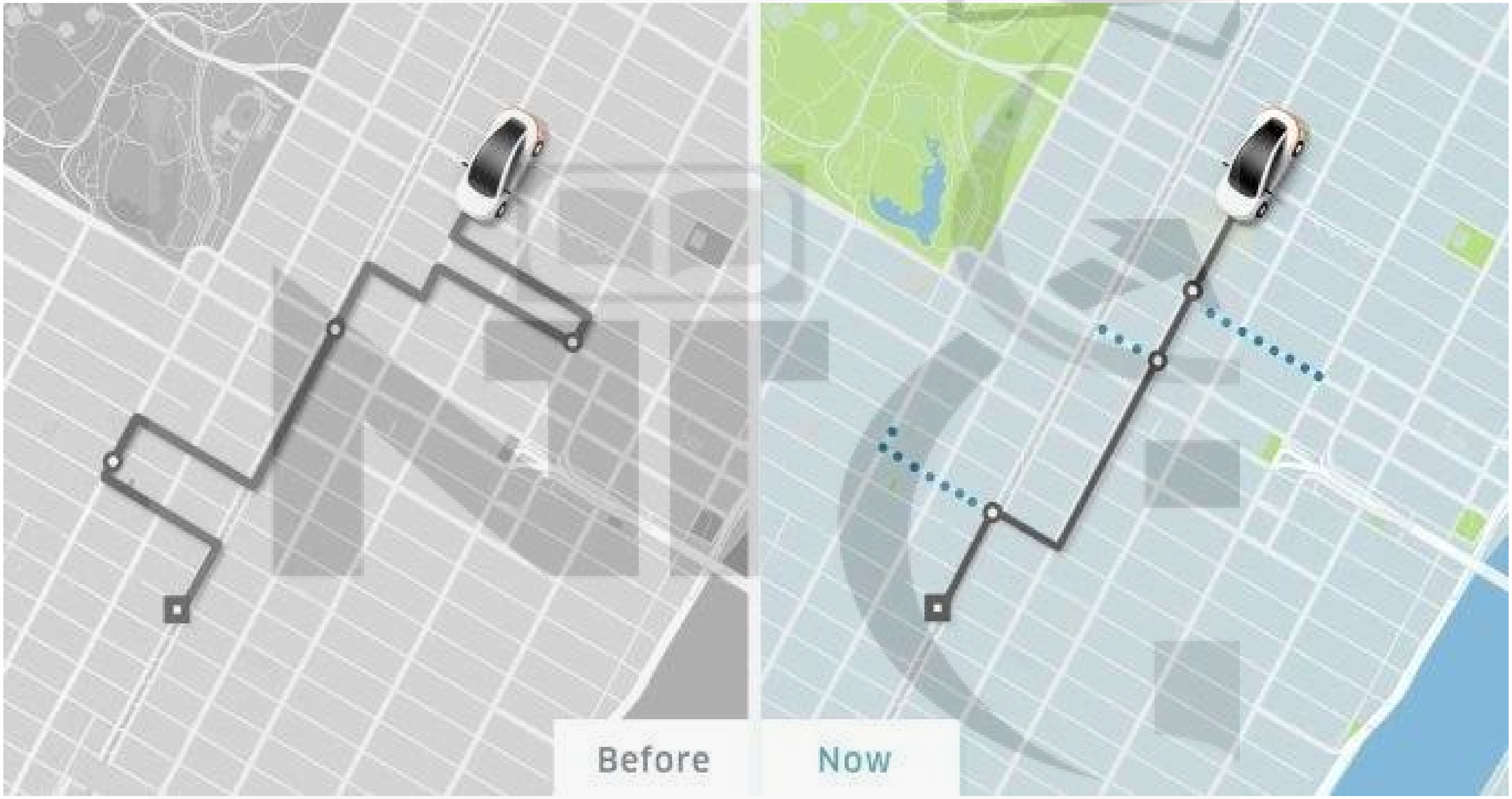



Diabetic Retinopathy Diagnosis
Cancer Detection
Cardiac Registration
Diabetes Detection
Brain Tumor Detection
Lung Nodule Classification
Thyroid Diagnosis
Fetal Localization
Prostate Image Segmentation

Transportation



Optimizing Rides and Routes



Advantages

- Improved Operational Efficiency
- Enhanced Customer Satisfaction
- Dynamic Pricing (Revenue Maximization)

Improved Operational Efficiency

- **How Data Science Helps**:
  - Predicts demand in specific areas using historical data, weather patterns, and events.
  - Optimizes routes in real-time to minimize delays and fuel consumption.
- **Business Impact**:
  - Reduces operational costs (e.g., fuel, vehicle maintenance).
  - Increases fleet utilization, ensuring more rides or deliveries per vehicle.
- **Example**:
  - **Uber** uses predictive analytics to dispatch drivers to high-demand areas, reducing idle time.

Enhanced Customer Satisfaction

- **How Data Science Helps**:
  - Reduces wait times for passengers or delivery customers.
  - Suggests accurate arrival times by analyzing traffic conditions and driver locations.
- **Business Impact**:
  - Higher customer retention and loyalty due to better service experiences.
  - Positive reviews and increased recommendations.
- **Example**:
  - **DoorDash** uses data science to optimize delivery routes, ensuring food arrives hot and fresh.

Dynamic Pricing

- **How Data Science Helps**:
  - Uses algorithms to adjust pricing based on supply and demand (e.g., surge pricing during peak hours or bad weather).
- **Business Impact**:
  - Maximizes revenue during high-demand periods while balancing driver supply.
  - Encourages more drivers to participate during peak times, improving availability.
- **Example**:
  - **Uber and Lyft** increase fares during concerts, sporting events, or bad weather to match demand.