Aarogya Setu: Big Data Analytics Case Study

Four Types of Data Analysis in Public Health Technology

Case Background

Aarogya Setu, launched in April 2020, became one of the world's fastest-adopted contact tracing applications with over 230 million users. The app generated massive volumes of structured and unstructured data including:

- **Location data**: GPS coordinates, Bluetooth proximity signals
- **Health data**: Self-reported symptoms, test results, vaccination status
- **Behavioral data**: App usage patterns, compliance metrics
- **Network data**: Contact graphs, movement patterns
- **Real-time streams**: Continuous location updates, proximity alerts

Big Data Characteristics (7 V's):

- **Volume**: Petabytes of location and proximity data from 230M+ users
- **Velocity**: Real-time contact tracing requiring sub-second processing
- **Variety**: GPS, Bluetooth, health records, demographics, survey responses
- **Veracity**: Data quality challenges from self-reporting and device variations
- **Value**: Extracting actionable insights for public health decision-making
- **Variability**: Inconsistent data patterns due to changing user behaviors and policies
- **Visualization**: Complex multi-dimensional data requiring advanced visual analytics

---

1. Descriptive Analytics: "What Happened?"

Objective

Understanding patterns in COVID-19 spread, user behavior, and app effectiveness using historical big data.

Big Data Techniques & Tools

Data Processing Stack:

- **Apache Spark** for distributed processing of massive location datasets
- **Hadoop HDFS** for storing petabytes of historical contact data
- **Apache Kafka** for real-time data ingestion from millions of devices
- **Elasticsearch** for fast querying of location and contact events

Analytical Methods:

1. **Spatial-Temporal Analysis**
   - Heat maps showing infection density across geographic regions
   - Time-series analysis of daily active users and contact events
   - Geospatial clustering using Apache Spark MLlib
2. **Network Analysis**
   - Contact graph construction using GraphX (Spark's graph processing)
   - Degree centrality analysis to identify super-spreader nodes
   - Community detection in contact networks
3. **Behavioral Analytics**
   - User engagement patterns using clickstream analysis
   - Compliance rates with quarantine recommendations
   - App usage correlation with demographic factors

Sample Insights Generated:

- "Mumbai recorded 2.3M contact events daily during peak pandemic period"
- "Users aged 25-40 showed highest app engagement (avg 8.5 sessions/day)"
- "Contact networks averaged 12 degrees of separation before containment measures"

Implementation Approach

The technical implementation utilized distributed computing frameworks to handle the massive scale of data processing required for real-time contact tracing and analytics.

---

2. Diagnostic Analytics: "Why Did It Happen?"

Objective

Understanding root causes of COVID-19 transmission patterns and identifying factors influencing app effectiveness.

Big Data Techniques & Tools

Advanced Analytics Stack:

- **Apache Spark MLlib** for correlation analysis on large datasets
- **R with SparkR** for statistical modeling on big data
- **Apache Drill** for interactive analysis across multiple data sources
- **Jupyter notebooks** with PySpark for exploratory data analysis

Analytical Methods:

1. **Correlation Analysis at Scale**
   - Cross-correlation between mobility patterns and infection rates
   - Feature correlation analysis using distributed computing
   - Time-lagged correlation analysis for transmission chains
2. **Causal Inference**
   - Propensity score matching for treatment effect analysis
   - Difference-in-differences analysis comparing regions with different adoption rates
   - Natural experiments using policy intervention timestamps
3. **Anomaly Detection**
   - Isolation Forest algorithms for detecting unusual transmission patterns
   - Statistical process control for identifying outbreak signals
   - Graph-based anomaly detection in contact networks

Key Diagnostic Questions Answered:

- Why did transmission rates vary significantly across similar demographic regions?
- What factors contributed to lower app adoption in certain communities?
- Why were some contact tracing alerts more effective than others?

Sample Findings:

- "High-density urban areas with >70% app adoption showed 23% faster outbreak detection"
- "Transmission clusters correlated strongly with public transport usage patterns (r=0.78)"
- "False positive rates increased 40% in areas with high Bluetooth interference"

Implementation Approach

Machine learning pipelines were designed to process correlation analysis and causal inference on distributed datasets, enabling the identification of transmission patterns and policy effectiveness factors.

---

3. Predictive Analytics: "What Will Happen?"

Objective

Forecasting COVID-19 spread, predicting high-risk areas, and anticipating resource needs using machine learning on big data.

Big Data Techniques & Tools

ML Pipeline:

- **Apache Spark MLlib** for scalable machine learning
- **TensorFlow on Spark** for deep learning models
- **Apache Airflow** for ML pipeline orchestration
- **MLflow** for model versioning and deployment
- **Apache Kafka** for real-time feature streaming

Predictive Models:

1. **Time Series Forecasting**
    - LSTM networks for multi-variate infection rate prediction
    - Prophet models for seasonal trend analysis
    - ARIMA models for short-term transmission forecasting
2. **Spatial Prediction Models**
    - Geographically Weighted Regression for location-based risk assessment
    - Spatial autoregressive models using contact network topology
    - Graph Neural Networks for transmission pathway prediction
3. **Risk Scoring Models**
    - Gradient Boosting (XGBoost) for individual risk assessment
    - Ensemble methods combining multiple data sources
    - Real-time scoring using streaming data

Prediction Scenarios:

- 7-day ahead infection rate forecasts by district
- Individual risk scores updated in real-time
- Hospital capacity requirements based on predicted case loads
- Optimal resource allocation for contact tracing teams

Implementation Approach

Real-time machine learning pipelines were established using streaming data frameworks to enable continuous model updates and prediction generation for millions of users simultaneously.

Model Performance Metrics:

- Infection rate prediction: MAPE of 12% for 7-day forecasts
- Individual risk scoring: AUC of 0.84 on validation set
- Hospital capacity prediction: 89% accuracy for 14-day forecasts

---

4. Prescriptive Analytics: "What Should We Do?"

Objective

Providing actionable recommendations for policy makers, healthcare systems, and individuals using optimization algorithms on big data.

Big Data Techniques & Tools

Optimization Stack:

- **Apache Spark with OptaPlanner** for large-scale optimization
- **Google OR-Tools** for constraint programming
- **Apache Flink** for real-time decision making
- **Redis** for caching optimization results
- **Apache Beam** for batch and stream processing

Prescriptive Methods:

1. **Resource Optimization**
   - Linear programming for optimal testing center placement
   - Vehicle routing problems for vaccine distribution
   - Staff scheduling optimization for contact tracing teams
2. **Policy Recommendation Engine**
   - Multi-objective optimization balancing health outcomes and economic impact
   - Simulation-based policy testing using agent-based models
   - Dynamic programming for adaptive lockdown strategies
3. **Personalized Interventions**
   - Reinforcement learning for personalized health recommendations
   - Recommendation systems for optimal behavior modification
   - Dynamic treatment regimen optimization

Decision Support Systems:

1. **Real-time Alert Optimization**
   - Minimize false positives while maximizing true positive detection
   - Optimize alert timing based on user behavior patterns
   - Dynamic threshold adjustment based on local transmission rates
2. **Resource Allocation**
   - Optimal placement of testing facilities based on predicted demand

- Healthcare worker deployment optimization
- Vaccine distribution logistics optimization

Implementation Approach

Large-scale optimization engines were deployed to process multiple competing objectives and constraints, providing real-time decision support for resource allocation and policy recommendations.

Prescriptive Outcomes:

- Reduced transmission rate by 18% through optimized alert timing
- 25% improvement in testing efficiency through optimal facility placement
- $50M cost savings in healthcare resource allocation
- 30% increase in user compliance through personalized recommendations