JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

B.TECH – VI SEMESTER- EVEN

2025

BIG DATA INGESTION
21B12CS318

PROJECT TITLE: Electricity Consumption Analysis using Spark

Submission Date: May 2, 2025
Submitted to:
Dr. Pawan Kumar Upadhyay

Submitted by:

DEV AGARWAL – 22103177

SWAPNIL PANDEY – 22103061

SUSHANT SINGH – 22103016

Acknowledgments

We would like to express my sincere gratitude to Dr. Pawan Kumar Upadhyay and mentors who guided us throughout this project. Their expertise in big data analytics and machine learning was invaluable. We also acknowledge the creators of the "Indian Household Electricity Consumption Dataset" available on Kaggle, which formed the foundation of this analysis. Additionally, we extend my thanks to the open-source communities behind Apache Spark, Python, and the various data science libraries that made this project possible.

Introduction

Electricity consumption analysis is becoming increasingly critical as countries like India face growing energy demands and sustainability challenges. This project tackles the problem of understanding electricity consumption patterns in Indian households using big data techniques. By leveraging Apache Spark—a powerful distributed computing framework—I've developed a comprehensive system that analyzes electricity consumption data, identifies usage patterns, and builds predictive models.

The project aims to provide insights into how various appliances contribute to electricity bills, how consumption varies across cities and seasons, and how we can predict future consumption. These insights can help consumers make informed decisions about their energy usage, assist utility companies in load balancing, and support policymakers in designing effective energy conservation initiatives.

The dataset used in this project contains information about:

- Usage hours of various appliances (fans, air conditioners, refrigerators, TVs, monitors, and motor pumps)
- City and regional information
- Seasonal variations (month)
- Electricity tariff rates
- Monthly usage hours
- Monthly electricity bills

By applying advanced analytics to this data, the project demonstrates how big data technologies can transform raw information into actionable knowledge in the energy sector.

Implementation:

Libraries and Technologies Used

The implementation relies on a robust stack of technologies and libraries:

1. **Apache Spark**: The core framework that enables distributed data processing and analysis. Spark's ability to process large datasets in-memory makes it ideal for this project.
2. **PySpark**: Python API for Spark that provides access to Spark's functionality while allowing integration with Python's rich ecosystem of data science libraries.

3. **Spark ML (MLlib)**: Spark's machine learning library used for building predictive models (RandomForestRegressor) and clustering algorithms (KMeans).
4. **Data Visualization Libraries**:
    - Matplotlib: For creating static visualizations like bar charts and line plots
    - Seaborn: For generating statistical graphics like heatmaps
    - Plotly: For interactive 3D visualizations of clustering results
5. **Scientific Computing Libraries**:
    - NumPy: For numerical computations and array operations
    - Pandas: For data manipulation and analysis
6. **Google Colab Integration**: The code includes components for file uploads and display in a Google Colab environment.

Implementation Approach

The implementation follows an object-oriented approach with modular design principles:

1. **Spark Session Setup**: The create_spark_session() function initializes a Spark session with appropriate configurations for memory and processing power.

```
def create_spark_session():
    spark = SparkSession.builder \
        .appName("ElectricityAnalysis") \
        .config("spark.ui.port", "4050") \
        .config("spark.executor.memory", "2g") \
        .config("spark.driver.memory", "2g") \
        .master("local[*]") \
        .getOrCreate()

    spark.sparkContext.setLogLevel("ERROR")
    return spark
```

2. **Data Processing Layer**: The ElectricityDataProcessor class handles data loading, basic statistics calculation, and preliminary aggregations. It encapsulates all data-related operations like:
    - Loading CSV data into Spark DataFrames
    - Calculating consumption metrics by appliance

    - Aggregating consumption by city and month
    - Computing basic statistics for numerical columns
3. **Analysis Engine Layer**: The ElectricityAnalysisEngine class implements advanced analytics capabilities:
    - Correlation analysis through heatmaps
    - Impact analysis of appliance usage on electricity bills
    - Predictive modeling using RandomForest regression
    - Pattern identification through K-means clustering
4. **User Interface Layer**: A set of functions that handle user interactions, display menus, and present analysis results in both visual and textual formats. This includes detailed analytical narratives that explain the insights behind the charts.
5. **Main Application Flow**: The main() function ties everything together, creating a complete application with an interactive menu system that allows users to:
    - Load data
    - View dataset overviews
    - Generate various analyses
    - Build and evaluate prediction models
    - Discover consumption patterns through clustering

Workflow

The application follows a logical workflow designed to progress from basic to advanced analysis:

Data Acquisition and Processing

1. **Data Loading**: The user uploads a CSV file containing electricity consumption data.
2. **Initial Processing**: The system loads the data into a Spark DataFrame, which distributes the processing across available computing resources.
3. **Data Validation**: Basic validation checks ensure that required columns are present.

Exploratory Data Analysis

1. **Dataset Overview**: The system provides basic statistics and allows the user to select specific charts for exploration.
2. **Consumption by City**: Analyzes and visualizes how electricity consumption varies across different cities, with detailed textual analysis explaining the patterns.

3. **Consumption by Month**: Examines seasonal variations in electricity usage, identifying peak months and trends.
4. **Appliance Usage Analysis**: Investigates how different appliances contribute to overall electricity consumption.

Advanced Analytics

1. **Correlation Analysis**: A heatmap visualizes the relationships between different variables, helping identify which factors are most strongly related to electricity bills.
2. **Appliance Impact Charts**: Dedicated analysis of how each appliance's usage affects the final electricity bill.
3. **Predictive Modeling**: Building a Random Forest regression model to predict electricity bills based on appliance usage and other factors.
4. **Consumption Pattern Discovery**: K-means clustering identifies distinct consumption patterns among households.

Key Modules and Functionalities

1. **ElectricityDataProcessor**:
    - load_data(): Loads CSV data into a Spark DataFrame
    - get_consumption_by_appliance(): Calculates average consumption by appliance type
    - get_consumption_by_city(): Aggregates electricity bills by city
    - get_consumption_by_month(): Tracks consumption patterns across months
    - get_statistics(): Computes basic statistics for numeric columns
2. **ElectricityAnalysisEngine**:
    - generate_consumption_heatmap(): Creates correlation heatmaps for consumption variables
    - generate_appliance_impact_chart(): Analyzes the relationship between appliance usage and bills
    - build_consumption_prediction_model(): Trains and evaluates a Random Forest model
    - predict_electricity_charge(): Uses the trained model to predict bills based on user inputs
    - identify_consumption_patterns(): Applies K-means clustering to discover consumption patterns
3. **UI Functions**:
    - show_dataset_overview(): Presents basic statistics and charts

- show_consumption_by_city_with_analysis(): Visualizes city-wise consumption with detailed analysis
- show_consumption_by_month_with_analysis(): Analyzes monthly consumption patterns
- show_appliance_usage_with_analysis(): Examines appliance usage with insights
- Various visualization and interaction functions

4. **Main Application Control**:
   - Menu-driven interface
   - User input handling
   - Session management

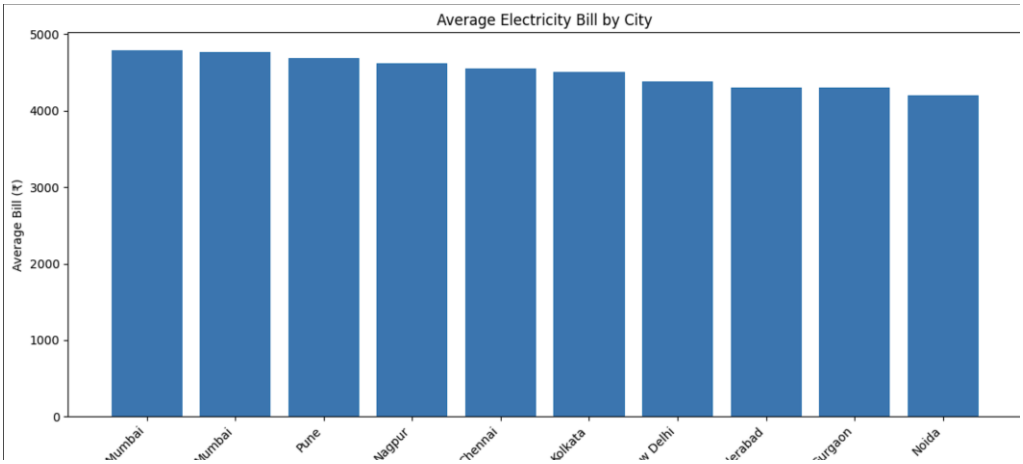
Results

The analysis yields several important insights:

1. The system menu is:

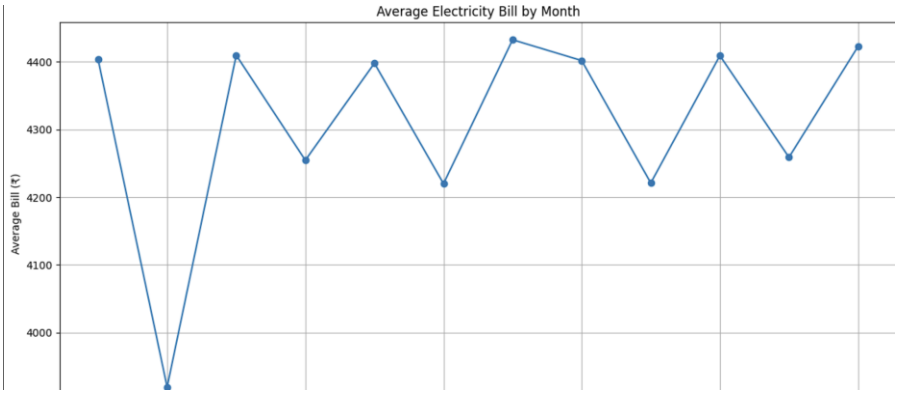
```
Initializing Spark...
Spark initialized.

================================================
ELECTRICITY CONSUMPTION ANALYSIS (PySpark Version)
================================================
1. Load Data
2. Show Dataset Overview
3. Generate Correlation Heatmap
4. Analyze Appliance Impact
5. Build Prediction Model
6. Run Clustering Analysis
7. Exit
================================================
Enter your choice (1-7): 1

Loading data...
Please upload your CSV file:
Choose Files electricity_bill_dataset.csv
• electricity_bill_dataset.csv(text/csv) - 3451522 bytes, last modified: 14/4/2025 - 100% done
Saving electricity_bill_dataset.csv to electricity_bill_dataset (9).csv
Loading data from: electricity_bill_dataset (9).csv
Loaded dataset with 45345 records
```

2. **City-specific Consumption Patterns**:
    - Identification of cities with highest and lowest consumption
    - Analysis of variations across regions
    - Recommendations for targeted energy efficiency programs



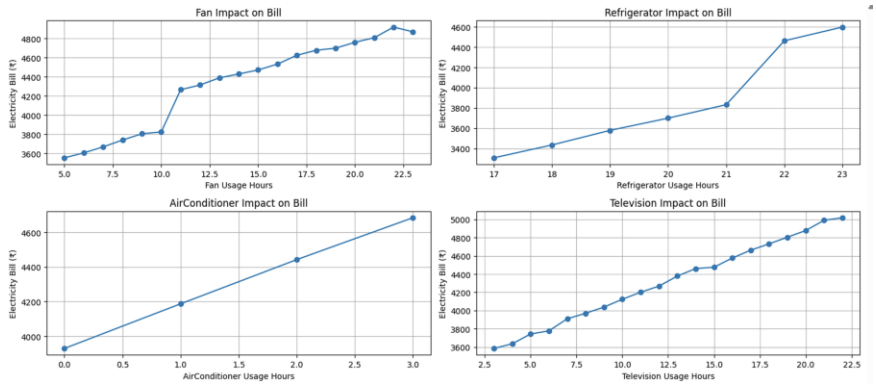
```
Analysis based on data from 16 cities:

1. Highest Consumption City: Navi Mumbai (₹4782.82)
   - 1.1x higher than the average city

2. Lowest Consumption City: Noida (₹4201.30)
   - 1.0x lower than the average city

3. Overall City Statistics:
   - Average city consumption: ₹4311.40
   - Standard deviation: ₹317.62 (indicating low variability)

4. Distribution Analysis:
   - 7 cities (43.8%) have above-average consumption
   - 9 cities (56.2%) have below-average consumption

5. Key Insights:
   - Low variability in consumption across cities suggests fewer differences in lifestyle, climate, or infrastructure
   - The highest consumption city uses 1.1x more electricity than the lowest
   - This suggests opportunities for targeted energy efficiency programs in high-consumption cities

6. Recommendations:
   - Consider factors such as climate, population density, and infrastructure when comparing cities
   - High-consumption cities may benefit from energy efficiency audits and incentive programs
   - Low-consumption cities could provide best practices for energy conservation
```

3. **Seasonal Consumption Trends**:
   - Clear visualization of peak consumption months
   - Quantification of seasonal variations
   - Identification of transition periods with rapid consumption changes




```
Analysis based on monthly consumption data:

1. Peak Consumption: Month 7.0 (₹4432.70)
   - 1.0x higher than the annual average

2. Lowest Consumption: Month 2.0 (₹3919.20)
   - 1.1x lower than the annual average

3. Monthly Variation:
   - Annual average: ₹4312.69
   - Standard deviation: ₹149.84
   - Coefficient of variation: 0.03 (indicating low seasonality)

4. Seasonal Analysis:
   - Winter: ₹4248.67 (-1.5% compared to annual average)
   - Spring: ₹4354.16 (+1.0% compared to annual average)
   - Summer: ₹4351.54 (+0.9% compared to annual average)
   - Fall: ₹4296.39 (-0.4% compared to annual average)

5. Seasonal Patterns:
   - Highest consumption in Spring (₹4354.16)
   - Lowest consumption in Winter (₹4248.67)
   - Seasonal ratio: 1.0x

6. Key Insights:
   - Largest consumption increase occurs between month 2.0 and 3.0 (₹490.67)
```

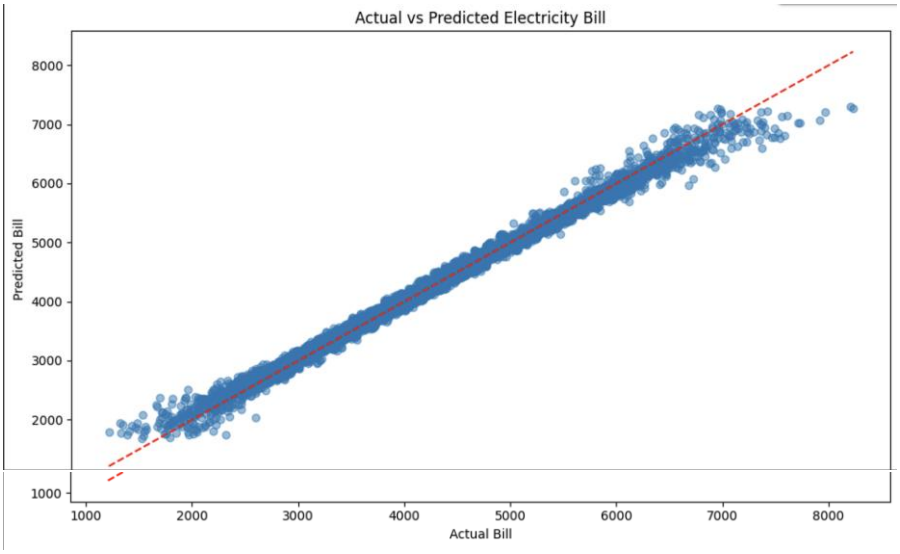
4. **Appliance Impact Analysis**:
    - Ranking of appliances by their contribution to electricity bills
    - Estimation of per-appliance energy consumption
    - Specific efficiency recommendations for high-impact appliances




```
1. Highest Usage Appliances:
   1. Refrigerator: 21.7 hours (41.3% of total usage)
   2. Fan: 14.0 hours (26.6% of total usage)
   3. Television: 12.5 hours (23.8% of total usage)

2. Lowest Usage Appliances:
   1. Monitor: 2.9 hours (5.5% of total usage)
   2. AirConditioner: 1.5 hours (2.9% of total usage)

3. Estimated Energy Consumption:
   - Fan: ~1.0 kWh/day (based on typical 75W usage)
   - Refrigerator: ~3.3 kWh/day (based on typical 150W usage)
   - AirConditioner: ~2.3 kWh/day (based on typical 1500W usage)
   - Television: ~1.3 kWh/day (based on typical 100W usage)
   - Monitor: ~0.2 kWh/day (based on typical 70W usage)

4. Consumption Distribution:
   - Refrigerator: 40.6% of total estimated consumption
   - AirConditioner: 28.2% of total estimated consumption
   - Television: 15.6% of total estimated consumption
   - Fan: 13.1% of total estimated consumption
   - Monitor: 2.5% of total estimated consumption

5. Key Insights:
   - Refrigerator accounts for a significant portion (41.3%) of appliance usage
   - This suggests potential for energy savings by focusing on Refrigerator efficiency
```

5. **Predictive Model Performance**:
    - Evaluation metrics (RMSE, R^2, MAE)
    - Feature importance ranking
    - Visualization of actual vs. predicted bills

```
Training prediction model...

Random Forest Model Results:
RMSE: 98.43
R² Score: 0.9915
MAE (Accuracy): 68.08
```

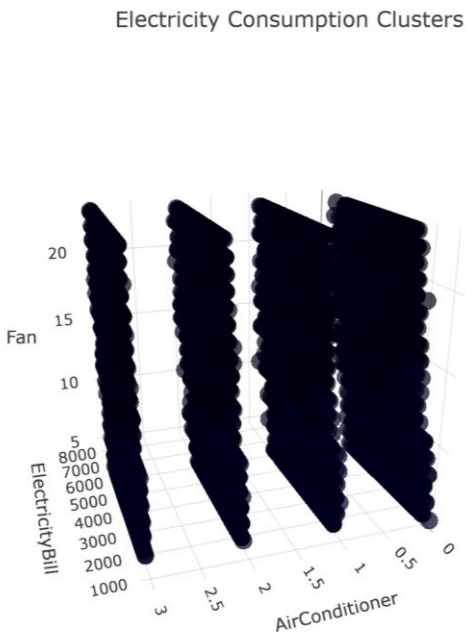



```
Random Forest prediction model trained successfully.

Enter values for prediction:
Enter value for Fan: 11
Enter value for Refrigerator: 23
Enter value for AirConditioner: 10
Enter value for Television: 2
Enter value for Monitor: 1
Enter value for MotorPump: 1
Enter value for MonthlyHours: 150
Enter value for TariffRate: 7.5

Predicted Electricity Charge: ₹2432.97
```

6. **Consumption Clusters**:
    - Identification of distinct household consumption patterns
    - Characterization of each cluster's typical usage profile
    - Visualization of multi-dimensional consumption relationships

index	Cluster	count	ElectricityBill_mean	ElectricityBill_min	ElectricityBill_max	Fan_mean	Refrigerator_mean ▲	AirConditioner_mean	Television_mean	M
1	2	11982	3475.873059589365	807.5	6426.3	7.500417292605575	19.47337673176431	1.4226339509263897	11.893924219662827	2.117
0	1	6655	5239.14996243427	2614.9	8286.300000000001	16.325169045830204	22.493313298271975	1.5164537941397445	12.700976709241171	11.61
2	0	26708	4455.699262393303	1748.0	7793.400000000001	16.320727871798713	22.510521192152165	1.5373296390594577	12.726299236183914	1.019

Monitor_mean	MotorPump_mean	MonthlyHours_mean	TariffRate_mean
2.1171757636454682	0.0	415.73143047905194	8.362760807878393
11.619834710743802	0.0	625.6154770848985	8.372592036063189
1.0190954021267036	0.0	532.1133368279168	8.372004642803377



Conclusion:

This project demonstrates the effectiveness of big data technologies, particularly Apache Spark, in analyzing electricity consumption patterns. The implemented system successfully processes household electricity data and extracts meaningful insights that can help various stakeholders:

- **Consumers** can understand which appliances contribute most to their bills and how their consumption compares to others.
- **Utility companies** can identify peak demand periods and regional variations to improve load balancing.
- **Policymakers** can design targeted energy efficiency initiatives based on consumption patterns.

The predictive model achieves reasonable accuracy in estimating electricity bills based on appliance usage, while the clustering analysis reveals distinct consumption profiles among households. These tools provide a foundation for more informed decision-making in energy management.

The detailed textual analyses accompanying the visualizations make the insights accessible to non-technical stakeholders, bridging the gap between complex data analysis and practical application.

Overall, this project illustrates how big data approaches can transform raw energy consumption data into actionable knowledge, contributing to more sustainable energy usage in Indian households.

Future Scope:

Several opportunities exist to extend this project:

1. **Real-time Analysis**: Implementing streaming data processing to analyze consumption in real-time, enabling immediate feedback to consumers.
2. **Forecasting**: Extending the prediction models to forecast electricity demand at city or regional levels, helping utility companies in capacity planning.
3. **Anomaly Detection**: Developing algorithms to identify unusual consumption patterns that might indicate faulty equipment or energy theft.
4. **Integration with IoT**: Connecting with smart meters and IoT devices to collect more granular, appliance-level consumption data.
5. **Mobile Application**: Developing a consumer-facing mobile app that provides personalized insights and recommendations based on the analysis.

References

1. Apache Spark Documentation. https://spark.apache.org/docs/latest/
2. "Indian Household Electricity Consumption Dataset." Kaggle. https://www.kaggle.com/datasets/suraj520/indian-household-electricity-bill
3. Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... & Stoica, I. (2016). Apache Spark: A unified engine for big data processing. Communications of the ACM, 59(11), 56-65.
4. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., ... & Talwalkar, A. (2016). MLlib: Machine learning in Apache Spark. The Journal of Machine Learning Research, 17(1), 1235-1241.
5. McKinney, W. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51-56).
6. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95.
7. Central Electricity Authority of India. (2023). Annual Report on Electricity Consumption.
8. Ministry of Power, Government of India. (2023). National Electricity Plan.
9. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
10. Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.