IoT Smart Home Appliances Data Analytics

DA331 Final Project Report

Rishita Agarwal
220150016
3rd Year BTech, DSAI
IIT Guwahati

***Abstract*—The "IoT Smart Home Appliances Data Analytics" project aims to predict various things about smart home appliances based on various features like temperatures, humidity etc. The project uses a deep learning model to do the predictions and Apache Kafka for real time data streaming.**

I. Introduction

IoT smart home appliances refer to household devices that are connected to the internet and can be monitored or controlled from a remote location using a smartphone or other networked device. IoT stands for **"internet of things",** which further explains the above definition.

II. Objective

This project aims to analyze and optimize energy consumption for IoT devices using environmental data. It consists of two main components:

- **Price Prediction Using Delhi Weather Data**: The first part focuses on using a data set of daily weather conditions in Delhi to predict energy prices for each day within the test dataset, helping to better understand price fluctuations based on weather patterns.
- **Temperature Optimization Using Madrid Climate Data**: The second part involves analyzing the daily climate conditions in Madrid to determine the optimal temperature settings. This aims to improve energy efficiency and optimize the performance of IoT devices under varying climate conditions.

The aim is to implement a full-fledged model that takes care of the following **objectives**:

- Data collection and generation
- Data visualization
- Model for prediction
- Evaluation of the model

In the data pre-processing phase, the dataset is streamlined by selecting relevant features and engineering synthetic data to enhance the training process. This refined data set serves as the foundation for training several predictive models, including Linear Regression, Random Forest, and Gradient Boosting Trees.

Each model is evaluated to determine its effectiveness in predicting the optimal temperature settings for home appliances. The chosen model, demonstrating the best performance during training, is subsequently applied to test data. These test data are dynamically sourced from a producer and processed by a consumer, enabling real-time predictions to efficiently optimize appliance temperatures.

III. Data Generation

The datasets used in this project are -
1.Daily Delhi Data
2.Madrid weather dataset hourly 2019-2022

Data Descriptions of the 2 datasets above are -:

A. Daily Delhi Data

- **date**: Represents specific dates formatted in YYYY-MM-DD. Data samples include days from the start of 2017, providing a temporal context for weather measurements.
- **meantemp**: This column represents the average temperature for each day. The values are expressed in degrees Celsius and provide an insight into the daily thermal conditions.
- **humidity**: Indicates the daily average humidity, expressed in grams of water vapor per cubic meter of air.
- **windspeed**: This metric shows the average wind speed for each day measured in kilometers per hour (kmph).
- **meanpressure**: Represents the average atmospheric pressure for each day, measured in atmospheric units (atm).

B. Madrid weather dataset hourly 2019-2022

The data set includes time series records with features such as time, external temperature, and humidity. The other existing climate-indicating columns have been removed for simplicity purposes. Each record is time-stamped and includes hourly measurements. The data is structured as follows:

- **Time**: Timestamp of the data record.
- **Temperature**: External temperature at the time of the record.
- **Humidity**: Humidity level at the time of the record.
- **DayOfYear**: The day of the year extracted from the timestamp.
- **Hour**: The hour of the day extracted from the timestamp.

IV. Real-Time Data Handling with Apache Kafka

Apache Kafka plays a pivotal role in the "IoT Smart Home Appliances Data Analytics" project by facilitating the real-time streaming of IoT device data. This section outlines the configuration of the Kafka environment and the mechanisms for data production and consumption within our analytics system.

A. Kafka Setup and Configuration

For our project, Apache Kafka is configured to handle high-throughput and low-latency streaming of data generated from various IoT devices. The Kafka cluster is set up with multiple brokers to ensure data resilience and fault tolerance.

- **Number of Brokers**: 3
- **Zookeeper Ensemble**: 3-node setup for managing the Kafka cluster
- **Replication Factor**: 2, ensuring data is replicated across brokers for fault tolerance
- **Partitions**: Each topic is partitioned across brokers to enhance parallelism and performance.
- **Topics Created**: raw-climate-data

B. Data Production

IoT devices send data, such as temperature and humidity readings, directly to a Kafka topic via producers integrated within each device. The data is serialized in JSON format, which allows flexible data structures that are easy to extend and modify.

- **Producer Configuration**: Producers are configured with acknowledgment settings to ensure data durability.
- **Serialization**: JSON serialization is used for message encoding.
- **Data Publishing Frequency**: Devices publish data every minute, providing near-real-time analytics capabilities.

C. Data Consumption

The Kafka consumers are part of our data processing pipeline, which pulls data from Kafka topics, preprocesses it, and feeds it into our predictive models for real-time analytics.

- **Consumer Groups**: Multiple consumers are configured in groups to ensure efficient data processing.
- **Offset Management**: Automatic offset management is configured to ensure that no data is missed or processed more than once.
- **Data Processing**: Upon consumption, data are transformed and used to make predictions using a Spark-based analytics model.

D. Integration with Spark

Apache Spark is used to process and analyze the data consumed from Kafka. This integration allows for complex operations like windowed computations and machine learning model inferencing on streaming data.

- **Spark Streaming**: Utilizes Spark's structured streaming to consume data from Kafka, apply transformations, and output predictions in real-time.
- **Dataframe Operations**: Dataframes are used to handle operations on structured data efficiently.
- **ML Model Integration**: Spark MLlib is used for applying pre-trained models to streaming data for immediate predictions.

These components ensure that our IoT Smart Home Appliances Data Analytics project can leverage real-time data to optimize device performance and predict future conditions effectively.

V. Code Files

For the project, I made 4 files:

- **datapreprocessing.ipynb** This file contains handling of missing values, transforming the dataframe, dropping the irrelevant columns and applying an algorithm to create synthetic data for training and testing. Visualization of the datasets is also done by making different plots.
- **model.py** This file contains the model description for different models, whose RMSE values are seen to decide the best model to be used for prediction on the test data. GBT has the lowest RMSE value so, training of the model is done on this regressor.

A. Model Results

Model Type	RMSE
Linear Regression	2.905398592466343
Random Forest	2.1244958998723593
Gradient Boosting Trees	1.4908961618196768

TABLE I

Comparison of RMSE values for different regression models

- **producer.py** This file reads the test data and send it row by row to the kafka consumer.
- **consumer.py** It listens for incoming messages on specific Kafka topics, processes these messages in real-time, and feeds them into machine learning models for prediction. The script handles data deserialization, and is crucial for enabling real-time decision-making based on the latest sensor data. The dashboard shows real time data transfer and the plots for the test data prediction.

VI. Dashboard

The Streamlit dashboard designed to display real-time data from a Kafka topic titled "raw-climate-data" provides a comprehensive view of temperature data alongside predicted values using a machine learning model. This interactive tool is structured to assist in monitoring and analyzing temperature fluctuations in real-time and incorporates various features to enhance user engagement and data understanding:

A. Dashboard Features

1) **Real-time Temperature and Predicted Temperature Plot:** This plot dynamically displays both the actual and predicted temperatures as new data streams in from Kafka. It uses line markers to indicate each data point, allowing for easy tracking of changes over time.
2) **Histogram of Temperature:** This histogram provides a visual distribution of temperature values, helping to identify common ranges and outliers. It allows for a quick statistical insight into the data's spread and central tendencies.
3) **Scatter Plot of Temperature vs. Prediction:** By plotting actual temperatures against predicted values, this scatter plot helps in visualizing the accuracy of the predictions. Points closer to the line of perfect agreement indicate higher accuracy.
4) **Historical Temperature Plot:** This plot offers a historical view of temperature data, plotting all past temperatures in a simple line graph format. It's useful for observing long-term trends and patterns.
5) **Forecasting Future Temperatures:** Utilizing a moving average, this plot forecasts future temperatures based on historical data. It provides a smoothed line that helps predict trends and reduce noise in data interpretation.
6) **Latest Temperature and Prediction:** This section highlights the most recent temperature and its corresponding prediction. It updates with each new data point, providing immediate insight into current conditions.

B. Anomaly Detection

The dashboard automatically flags temperature readings that are significantly higher than usual (e.g., above the 99th percentile). These anomalies are reported in real time, drawing attention to potential issues or outliers that may require further investigation.

VII. Conclusion

This project effectively demonstrates the power of integrating IoT with smart home appliances to enhance the analytical capabilities using real-time data. By leveraging Apache Kafka for robust data streaming and Apache Spark for intensive data processing and predictive modeling, we have developed a scalable and dynamic system that can handle large volumes of data with low latency.

The implementation of various machine learning models to predict temperature and optimize energy consumption illustrates the potential for smart technologies to significantly improve efficiency and cost-effectiveness in managing home environments. The models tested show promising results, with Gradient Boosting Trees providing the most accurate predictions, evidenced by the lowest RMSE score among the models evaluated.

The real-time dashboard developed using Streamlit enhances the user experience by providing live insights into temperature data, prediction accuracies, and detecting anomalies. This not only helps in immediate decision-making but also in understanding long-term trends and patterns that could inform better strategies for energy management and device performance.