COMP.CS.320 Data Intensive Programming – Tampere University

Olli Könönen

Prediction Modeling Using Weather and Energy Data

1. Introduction

The aim of this project was to explore machine learning approaches for predicting temporal patterns in energy consumption and related weather conditions. Using time series data from a research project (ProCem: https://www.senecc.fi/projects/procem-2), I experimented with two popular classification models, **Random Forest** and **Naive Bayes** to predict:

1. **Month of the year (1–12)**
2. **Hour of the day (0–23)**
3. **Day of the week (Monday–Sunday)**

The models were trained using features derived from both weather measurements (temperature, humidity, wind speed) and building power consumption (tenants, maintenance systems, solar panels), with electricity price included in some cases.

2. Technologies Used

- **Scala –** Programming language used for scripting, data preparation, and machine learning pipelines in Spark. Scala's strong functional programming features make it ideal for Spark jobs.
- **Apache Spark (MLlib) –** Distributed data processing engine. I used Spark's MLlib for machine learning pipelines, feature transformations, and evaluation metrics.
- **Databricks –** Cloud-based platform for running Spark notebooks, managing clusters, and interactive development.
- **Random Forest Classifier –** Ensemble decision tree model suitable for multi-class classification. It provides robustness against overfitting and handles non-linear relationships. More on: https://www.geeksforgeeks.org/dsa/random-forest-classifier-using-scikit-learn/
- **Naive Bayes Classifier –** Probabilistic classifier based on Bayes theorem. It assumes feature independence and is efficient for categorical or discretized

numerical features. More on: https://www.geeksforgeeks.org/machine-learning/naive-bayes-classifiers/

3. Dataset

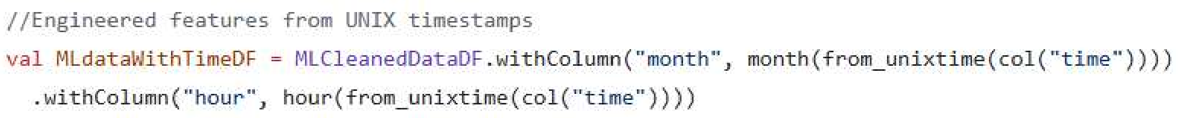
The dataset consisted of 13 months of minute-level measurements collected on a Tampere University campus building: Kampusareena. Each row represents the average value for one minute. Key columns included:

Table 1: Features for predicting

Column	Type	Description
time	long	UNIX timestamp (seconds)
temperature	double	Weather station temperature (°C)
humidity	double	Weather station humidity (%)
wind_speed	double	Wind speed (m/s)
power_tenants	double	Electricity consumption by tenants (W)
power_maintenance	double	Electricity usage for building systems (W)
power_solar_panels	double	Power generated by solar panels (W)
electricity_price	double	Market price of electricity (€/MWh)

Target variables were generated from UNIX timestamps by

Picture 1: Target variables



```
//Engineered features from UNIX timestamps
val MLdataWithTimeDF = MLCleanedDataDF.withColumn("month", month(from_unixtime(col("time"))))
  .withColumn("hour", hour(from_unixtime(col("time"))))
```

Target Variables:

- `month` – Month of the year (1–12)
- hour – Hour of the day (0–23)
- `WeekDay` – Name of the day (Monday–Sunday)

4. Methodology

The workflow began with data preparation, which involved loading the parquet dataset into a Spark DataFrame, removing missing values, and extracting the temporal features. For model input, feature columns were assembled into a single vector using Spark's VectorAssembler, while categorical targets were indexed using StringIndexer.

The machine learning models were implemented as Spark pipelines. Each pipeline consisted of a feature assembler, a label indexer, and a classifier. Random Forest was configured with eight trees to classify the multi-class targets. Naive Bayes was configured with the default parameters for multi-class classification. The pipelines allowed consistent handling of data transformations, model fitting, and prediction in a reproducible and modular workflow.

Model evaluation was performed using both built-in Spark evaluators and custom metrics. Accuracy was calculated as the proportion of exact correct predictions. Additional metrics included the proportion of predictions within one or two units of the true value and the average probability assigned to the correct class. These metrics provide a more detailed view of model performance, particularly for cyclical targets such as hours of the day or months of the year.

5. Analysis of Model Performance

In the prediction tasks, Random Forest consistently showed stronger performance than Naive Bayes for predicting the month and the hour of the day. This was due to its ability to capture complex, non-linear relationships between weather and power consumption features. Including more relevant features, particularly power consumption measurements, improved accuracy in the Random Forest model because these features contain structured patterns related to human activity and building usage that correlate with time, while the ensemble nature of Random Forest helps mitigate overfitting.

Naive Bayes, on the other hand, performed slightly better for predicting the day of the week. This is likely because day-of-week patterns are more regular and less dependent on complex feature interactions, which suits the probabilistic assumptions of Naive Bayes. Random Forest still performed well, but its advantage diminishes when the target depends more on broad trends than on detailed feature interactions.

Overall, combining both weather and power consumption features increased model accuracy across tasks, as the combination of environmental conditions and building usage provides a richer representation of temporal patterns. Random Forest was the stronger model for tasks requiring modeling of non-linear, correlated inputs, while Naive Bayes remained competitive when the patterns were simpler and more regular.

6. Conclusion

The project demonstrates the practical application of Spark ML pipelines for predicting temporal patterns in energy consumption and related weather variables. Data preparation, pipeline construction, and model evaluation are organized to ensure reproducibility and clarity. Random Forest provides a flexible framework capable of handling complex interactions among features, while Naive Bayes offers a probabilistic approach that is computationally efficient for simpler patterns. The methodology established in this project taught me a good basis for further predictive modeling in energy systems and time series analysis.