# "Real-Time Electricity Consumption Forecasting in Office Buildings using Ensemble Learning with Spark and Apache Kafka"

**Group Members:**

Shubham Kumar(IIT2020007)
Raj Chhari (IIT2020010)
Shashikant Thakur(IIT2020024)
Nilesh Singh (IIB2020038)
Ankit Kumar (IIT2020011)

# Tables Of Contents

# Introduction

1. Ensemble learning, especially of the random forest algorithm, can be used towards real time energy prognosis in office buildings for efficiency and sustainability purposes of electricity consumption.

2. Ensemble learning reduces limitations on individual models and leads to better accuracy in prediction using several learning algorithms.

3. Ensemble learning combined with apache spark for data processing and apache kafka for real time streaming is a proposed solution which promises accurate and prompt forecasts. Such technique maximise the efficiency as well fits into environmental consciousness as a modern management theory of today.

4. Finally, the combination of ensemble learning and state-of-the-art technology is targeted towards increasing efficiency and reducing costs in handling office buildings.

# LITERATURE REVIEW

- [1] Ensemble Learning for Electricity Consumption Forecasting in Office Buildings

This paper introduces three ensemble learning models for short-term load forecasting in dynamic power systems. The study compares gradient boosted regression trees, random forests, and an adapted Adaboost model using real data from an office building. Results reveal that the modified Adaboost model outperforms reference models in accurately forecasting electricity consumption for the next hour.

- [2] Improvising Processing of Huge Real Time Data Combining Kafka and Spark Streaming

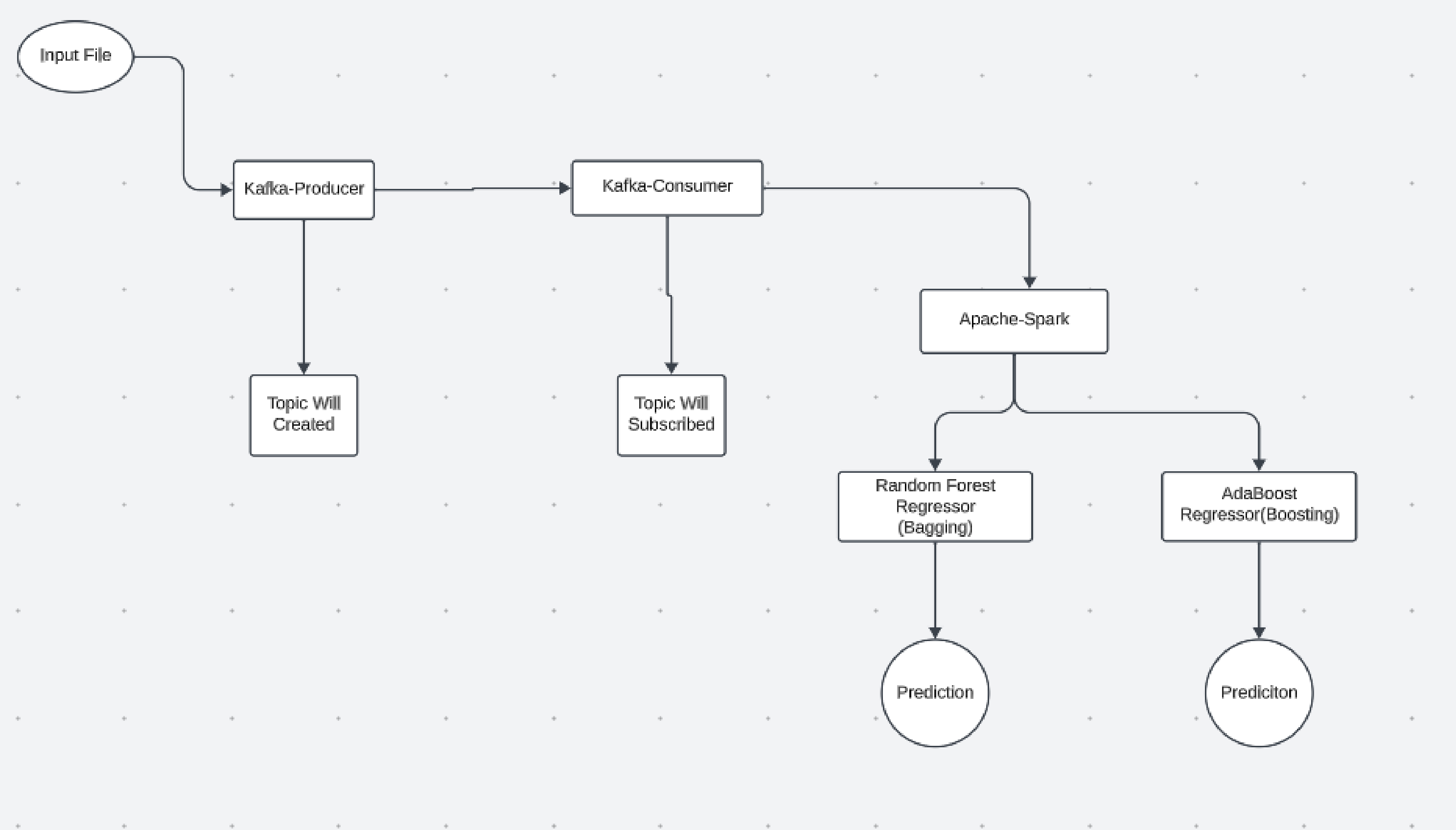
The Cloud Computing era introduces innovations in data processing, storage, and internet security. Knowledge Discovery is crucial, with the World Wide Web serving as a hub for these advancements. The surge in data has strained the WWW, necessitating efficient processing. Stream processing, exemplified by Apache Spark and Kafka, addresses real-time data challenges. Experimental results show Spark's efficiency with large datasets, while Kafka combined with Spark's execution time depends on dataset size.

# LITERATURE REVIEW

**[3]Forecasting Electricity Consumption in a Moroccan Educational Institution**

This paper highlights the significant role of predictive analytics in ensuring a reliable power supply. It focuses on benchmarking commonly used forecasting models for predicting electrical energy consumption in educational institutions. Utilizing a real use case and Big Data ecosystem based on SMACK architecture, the study analyzes six years of data sets, including planning and meteorological data, impacting the energy consumption of the National School of Applied Sciences in El Jadida, Morocco. The objective is to assess the prediction performance of various models and identify the most accurate one for forecasting electricity consumption in educational settings.

# FlowChart


Input File
Kafka-Producer
Kafka-Consumer
Topic Will Created
Topic Will Subscribed
Apache-Spark
Random Forest Regressor (Bagging)
AdaBoost Regressor(Boosting)
Prediction
Prediciton

# 3. Ensemble Learning Models

## 1.Bagging (Bootstrap Aggregating):

**Example: Random Forest**

```
rf = RandomForestRegressor(labelCol='Global_active_power', featuresCol="features", numTrees=100)
(trainingData, testData) = data1.randomSplit([0.8, 0.2])
```

## 2.Boosting

**Example: AdaBoost(Adaptive Boosting)**

```
adaboost_model_1 = AdaBoostRegressor(base_regressor, n_estimators=100, learning_rate=0.1, random_state=42)
adaboost_model_1.fit(X_train, y_train)
```

# 4. Model Training and Evaluation

```
Mean Squared Error: 0.25
R-squared: 0.90
```

```
Mean Squared Error: 0.26
R-squared: 0.84
```

```
Mean Squared Error: 0.26
R-squared: 0.84
```

```
Mean Squared Error: 0.25
R-squared: 0.89
```

AdaBoost

Getting MSE

```
from pyspark.ml.evaluation import R
evaluator = RegressionEvaluator(lab
r2 = evaluator.evaluate(predictions
print(r2)
```

```
1.822015572618956
```

Random Forest

# Methodology

# 1. Data Collection and Streaming

In the first instance, electric power consumption data will be gathered(In our case simulated)

This information will be sent through in real time, and it will use a software called Apache kafka.

Accordingly, Kafka supports efficient mechanisms of handling streams in distributed systems

```
print(data1.show())

+-----+-----+-------------------+---------------------+-------+----------------+--------------+--------------+--------------+-----------------+
| Date| Time|Global_active_power|Global_reactive_power|Voltage|Global_intensity|Sub_metering_1|Sub_metering_2|Sub_metering_3|         features|
+-----+-----+-------------------+---------------------+-------+----------------+--------------+--------------+--------------+-----------------+
|13498|62640|              4.216|                0.418| 234.84|            18.4|           0.0|           1.0|          17.0|[13498.0,62640.0]|
|13498|62700|               5.36|                0.436| 233.63|            23.0|           0.0|           1.0|          16.0|[13498.0,62700.0]|
|13498|62760|              5.374|                0.498| 233.29|            23.0|           0.0|           2.0|          17.0|[13498.0,62760.0]|
|13498|62820|              5.388|                0.502| 233.74|            23.0|           0.0|           1.0|          17.0|[13498.0,62820.0]|
|13498|62880|              3.666|                0.528| 235.68|            15.8|           0.0|           1.0|          17.0|[13498.0,62880.0]|
|13498|62940|               3.52|                0.522| 235.02|            15.0|           0.0|           2.0|          17.0|[13498.0,62940.0]|
|13498|63000|              3.702|                 0.52| 235.09|            15.8|           0.0|           1.0|          17.0|[13498.0,63000.0]|
|13498|63060|                3.7|                 0.52| 235.22|            15.8|           0.0|           1.0|          17.0|[13498.0,63060.0]|
|13498|63120|              3.668|                 0.51| 233.99|            15.8|           0.0|           1.0|          17.0|[13498.0,63120.0]|
|13498|63180|              3.662|                 0.51| 233.86|            15.8|           0.0|           2.0|          16.0|[13498.0,63180.0]|
|13498|63240|              4.448|                0.498| 232.86|            19.6|           0.0|           1.0|          17.0|[13498.0,63240.0]|
|13498|63300|              5.412|                 0.47| 232.78|            23.2|           0.0|           1.0|          17.0|[13498.0,63300.0]|
|13498|63360|              5.224|                0.478| 232.99|            22.4|           0.0|           1.0|          16.0|[13498.0,63360.0]|
|13498|63420|              5.268|                0.398| 232.91|            22.6|           0.0|           2.0|          17.0|[13498.0,63420.0]|
|13498|63480|              4.054|                0.422| 235.24|            17.6|           0.0|           1.0|          17.0|[13498.0,63480.0]|
|13498|63540|              3.384|                0.282| 237.14|            14.2|           0.0|           0.0|          17.0|[13498.0,63540.0]|
|13498|63600|               3.27|                0.152| 236.73|            13.8|           0.0|           0.0|          17.0|[13498.0,63600.0]|
|13498|63660|               3.43|                0.156| 237.06|            14.4|           0.0|           0.0|          17.0|[13498.0,63660.0]|
|13498|63720|              3.266|                  0.0| 237.13|            13.8|           0.0|           0.0|          18.0|[13498.0,63720.0]|
|13498|63780|              3.728|                  0.0| 235.84|            16.4|           0.0|           0.0|          17.0|[13498.0,63780.0]|
+-----+-----+-------------------+---------------------+-------+----------------+--------------+--------------+--------------+-----------------+
only showing top 20 rows

None
```

# 2. Data Preprocessing

- When data is sent to Kafka, Apache Spark processes the data since it is a powerful data processing system
- Preprocessing of data involves cleaning, normalization, and feature engineering
- This involves working on handling missing data, disposing outliers, and features converting non numerical data like date and time to numerical dataset.

# Methodology

1. **Preprocessing** : Removing null values and converting date and time to numerical format.
2. **VectorAssembler** : Converting input dataset to vectorassembler to output a single feature.
3. **Random Forest :** The output of vectorAssembler is passed to randomForestRegressor, using 100 trees.
4. Adaboost : preprocessed data is passed to adaBoost model.
5. **Output** : The MSE is calculated and predictions are made.

# AdaBoost(Methodology)

**Weak Learners (Base Estimators):** These are the models that AdaBoost combines to create a strong learner. In the case of AdaBoost Regressor, decision trees are commonly used as weak learners.

**Weighted Training:** AdaBoost assigns weights to each data point in the training set. Initially, all weights are set equally. After each iteration, the weights of the misclassified points are increased, and the weights of the correctly classified points are decreased. This allows the algorithm to focus more on the difficult-to-predict instances.

**Combining Weak Learners:** At each iteration, AdaBoost fits a weak learner to the data, and the model's performance is evaluated. The weak learner's contribution to the final model is determined based on its accuracy. The more accurate the weak learner, the more influence it has in the final combined model.

**Final Model Prediction:** The final prediction is a weighted sum of the predictions from all the weak learners. The weights are determined by the accuracy of each weak learner.

# Results

```
# Show predictions
predictions.select("Date", "Time", "prediction").show()
```

```
+-----+-----+-----------------+
| Date| Time|       prediction|
+-----+-----+-----------------+
|13498|62640|3.186776598021442|
|13498|62700|3.186776598021442|
|13498|62760|3.186776598021442|
|13498|62880|3.186776598021442|
|13498|63360|3.186776598021442|
|13498|63840|3.186776598021442|
|13498|64440|3.186776598021442|
|13498|64560|3.186776598021442|
|13498|65160|3.190753371662782|
|13498|65220|3.190753371662782|
|13498|65280|3.190753371662782|
|13498|65580|3.190753371662782|
|13498|65880|3.190753371662782|
|13498|66180|3.190753371662782|
|13498|67260|3.190753371662782|
|13498|67620|3.190753371662782|
|13498|67860|3.166534562093986|
|13498|67980|3.166534562093986|
|13498|68100|3.166534562093986|
|13498|68160|3.166534562093986|
+-----+-----+-----------------+
only showing top 20 rows
```

# Results

| | Code | n_estimators | learning_rate | MSE | R-squared |
|---|---|---|---|---|---|
| 0 | Code 1 | 100 | 0.1 | 0.247019 | 0.892165 |
| 1 | Code 2 | 50 | Default | 0.258944 | 0.838683 |
| 2 | Code 3 | 150 | 0.01 | 0.245315 | 0.899805 |

# Conclusion

In conclusion, the method of predicting electricity usage in office buildings in time using a combination of learning, Apache Spark and Apache Kafka is a highly effective and adaptable solution

It empowers building managers with timely insights to optimize energy consumption

By utilizing learning techniques alongside the real time capabilities of Apache Kafka and Spark this approach offers a solution, for enhancing energy efficiency reducing costs and contributing to a more sustainable future

# References


- [1] https://www.researchgate.net/publication/341251832_Ensemble_Learning_for_Electricity_Consumption_Forecasting_in_Office_Buildings
- [2] https://norma.ncirl.ie/4249/1/jeevantikalingalwar.pdf
- [3] https://www.researchsquare.com/article/rs-248534/v1
- [4] https://site.ieee.org/pes-iss/data-sets/
- [5] https://www.sciencedirect.com/science/article/pii/S0925231220307372

# "Thank You"