Monash University

FIT5202 - Data processing for Big Data 2025 S2

Assignment 2A: **Building Models for Building Energy Prediction**

Due: **(23:55 Friday 26/Sep/2025 End of Week 9)**
Worth: 15% of the final marks

Background

Accurate energy consumption forecasting is critical for modern power grids. It enables utility companies to balance supply and demand, prevent outages, and integrate renewable energy sources more effectively. For consumers, understanding energy usage patterns can lead to significant cost savings and a reduced carbon footprint.

The way we get and use electricity is changing in a big way. For a long time, large power plants made electricity and sent it to our homes. This was a one-way street. Now, we have new goals. We need our power to be:

1. **Reliable:** We want to have electricity whenever we need it.
2. **Affordable:** We want the cost of power to be fair and not too expensive.
3. **Clean:** We need to use more clean energy, like solar and wind, to protect the environment.

To meet these goals, we are building a "Smart Grid." A smart grid is a modern power system that utilises computers and the internet to enhance its efficiency. A key part of this system is the **smart meter**. These devices are in people's homes, and they measure how much electricity is being used. They send this information back to the power company very often. This creates enormous amounts of data.

This data is very useful. By studying it, power companies can predict how much electricity people will need at any given time (**energy forecasting)**. Good forecasting is essential for several reasons:

- **Making the Right Amount of Power:** It helps companies make just enough electricity to meet everyone's needs, without wasting any or running short.
- **Using Clean Energy:** Power from the sun and wind can change a lot. Forecasting helps companies plan for these changes and use more clean energy.
- **Preventing Problems:** It helps companies manage busy "peak hours" when everyone is using a lot of power at once.

Big data processing enables us to analyse vast datasets from smart meters, weather sensors, and building characteristics, thereby building precise predictive models. By identifying key drivers of energy consumption—such as time of day, weather conditions, and appliance usage—we can forecast future demand with high accuracy.

In this assignment, you will assume the role of a data scientist working for a power company. You will use a real dataset that shows how much electricity different buildings use over time. Your job will be to use **Apache Spark** to study this data and build a program. This program will learn from the old data to predict how much electricity a building will use in the future.

Objective of the Project

This assignment is a two-part assignment:

In **Assignment 2A**, we will use Apache Spark's MLlib to construct and train machine learning models. Our focus will be on accurately predicting a building's aggregate energy consumption based on sensor and monitoring readings, as well as environmental data.

Finally, **Assignment 2B** will utilise Apache Spark Structured Streaming and our trained ML model from 2A to process a live stream of energy data and make dynamic, real-time consumption predictions.

The Datasets:

- building_information.csv: Contains building information.
- meters.csv: Contains energy meter reading.
- weather.csv: Contains weather information.
- Metadata: description and data type of the dataset, at the end of this document.

What you need to achieve

Use case 1	Based on the historical dataset, build an ML model that can predict aggregated building energy consumption.	Regression

Architecture

The following figure represents the overall architecture of the assignment setup. **Part A** of the assignment consists of preparing the data, performing data exploration and extracting features, and building and persisting the machine learning models.

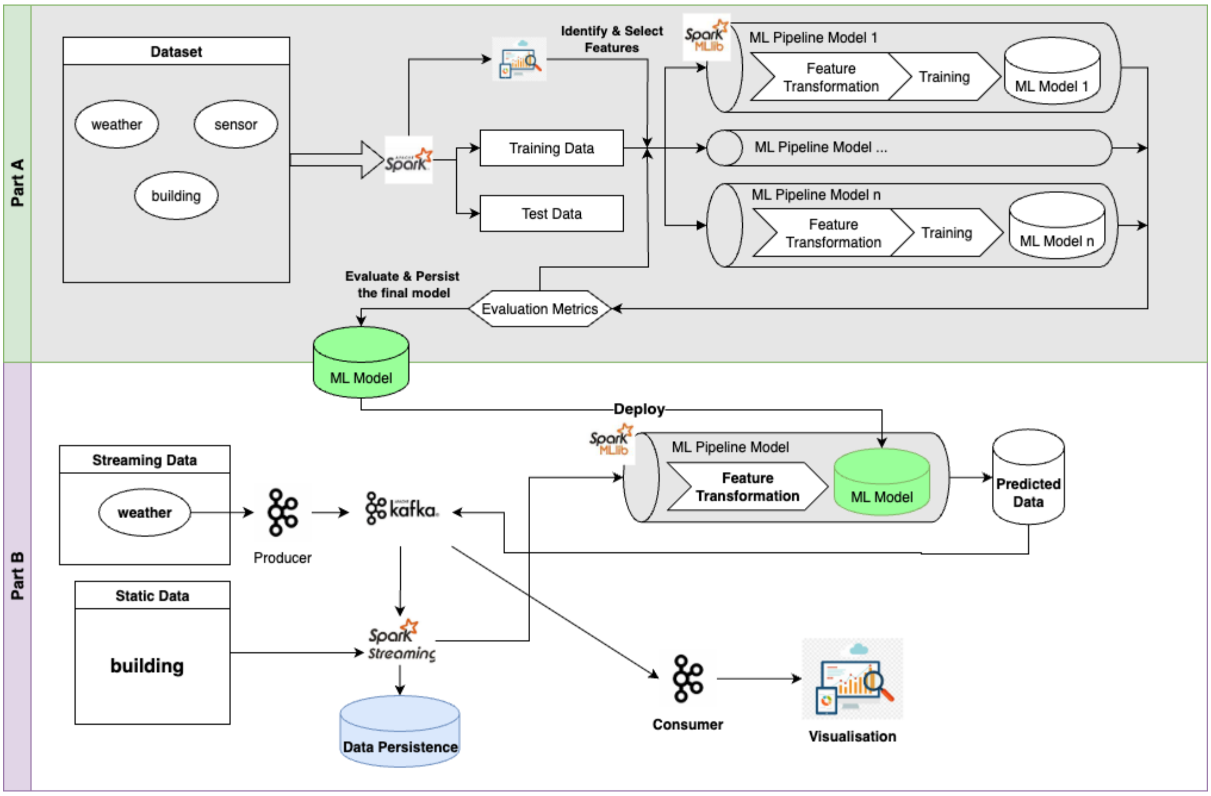

Fig 1: Overall Architecture for Assignment 2 (This assignment is Part A)

In both parts, you must implement the solutions using PySpark DataFrame/MLlib for the data pre-processing and machine learning pipelines. Excessive use of Pandas for data processing is discouraged. Please follow the steps to document the processes and write the code in your Jupyter Notebook.

Getting Started

- Download the datasets from Moodle.
- Download a template file for submission purposes:
    - ***A2A_template.ipynb*** file in Jupyter Notebook to write your solution. Rename it into the format (for example, ***A2A_xxxx0000.ipynb*. xxxx0000** is your authcate ID.
- You will use Python 3+ and PySpark 3.5.0+ for this assignment (This environment is the same as we used in labs.)

IMPORTANT:

Please answer each question in your Jupyter notebook file using code/markdown cell. Acknowledge any ideas or codes you referenced from others in the markdown cell or reference list.

If you use generative AI tools, all prompts you use should also be included in the reference section.

Part 1: Data Loading, Transformation and Exploration (35%)

In this section, you must load the given datasets into PySpark DataFrames and use *DataFrame functions* to process the data. For plotting, various visualisation packages can be used, but please ensure that you have included instructions to install the additional packages and that the installation will be successful in the provided Docker container (in case your marker needs to clear the notebook and rerun it).

1.1 Data Loading (5%)

1. Write the code to create a SparkSession. For creating the SparkSession, you need to use a SparkConf object to configure the Spark app with a proper application name, to ensure the maximum partition size does not exceed 32MB, and to run locally with all CPU cores on your machine[1] (note: if you have insufficient RAM, reducing the number of cores is acceptable.) (2%)
2. Write code to define the schemas for the datasets, following the data types suggested in the metadata file. (2%)
3. Using your schemas, load the CSV files into separate data frames. Print the schemas of all data frames. (1%)

1.2 Data Transformation and Feature Creation (15%)

In this section, we primarily have three tasks:

1) The dataset includes sensors with **hourly** energy measurements. However, as a grid operator, we don’t need this level of granularity and lowering it can reduce the amount of data we need to process. For each building, we will aggregate the metered energy consumption in **6-hour intervals (0:00-5:59, 6:00-11:59, 12:00-17:59, 18:00-23:59)**. This will be our **target (label) column** for this prediction. Perform the aggregation for each building.

In the **weather** dataset, there are some missing values (null or empty strings). It may lower the quality of our model. Imputation is a way to deal with those missing values. Imputation is the process of replacing missing values in a dataset with substituted, or "imputed," values. It's a way to handle gaps in your data so that you can still analyse it effectively without having to delete incomplete records.

2) Refer to the Spark MLLib imputation API and fill in the missing values in the weather dataset. You can use mean values as the strategy. https://spark.apache.org/docs/3.5.5/api/python/reference/api/pyspark.ml.feature.Imputer.html

We know that different seasons may affect energy consumption—for instance, a heater in winter and a cooler in summer. Extracting peak seasons (summer and winter) or off-peak seasons (Spring and Autumn) might be more useful than directly using the month as numerical values.

3) The dataset has 16 sites in total, whose locations may span across different countries. **Add a column** (peak/off-peak) to the weather data frame **based on the average air temperature.** The top **3 hottest months and the 3 coldest months** are considered “peak”, and the rest of the year is considered “off-peak”.

[1] More information about Spark configuration can be found in https://spark.apache.org/docs/latest/configuration.html

Create a data frame with all relevant columns at this stage, we refer to this data frame as **feature_df**.

1.3 Exploring the data (15%)

You can use either the CDA or the EDA method mentioned in Lab 5.

Some ideas for CDA:

a) Older buildings may not be as efficient as new ones, therefore need more energy for cooling/heating. It's not necessarily true though, if the buildings are built with higher standards or renovated later.

b) A multifloored or larger building obviously consumes more energy.

1. With the feature_df, write code to show the basic statistics: a) For each numeric column, show count, mean, stddev, min, max, 25 percentile, 50 percentile, 75 percentile; b) For each non-numeric column, display the top-5 values and the corresponding counts; c) For each boolean column, display the value and count. (note: pandas describe is allowed for this task.) **(5%)**
2. Explore the dataframe and write code to present **two plots of multivariate analysis**, describe your plots and discuss the findings from the plots. **(5% each)**
    - ○ 150 words max for each plot's description and discussion
    - ○ Note: In the building metadata table, there are some latent columns (data that may or may not be helpful, their meanings is unknown due to privacy and data security concerns).
    - ○ Feel free to use any plotting libraries: matplotlib, seabon, plotly, etc. You can refer to https://samplecode.link

Part 2. Feature extraction and ML training (40%)

In this section, you must use PySpark DataFrame functions and ML packages for data preparation, model building, and evaluation. Other ML packages, such as scikit-learn, would receive **zero** marks.

2.1 Discuss the feature selection and prepare the feature columns (10%)

1. Based on the data exploration from 1.2 and considering the use case, discuss the importance of those features (For example, which features may be useless and should be removed, which feature has a significant impact on the label column, which should be transformed), which features you are planning to use? Discuss the reasons for selecting them and how you create/transform them[2]
    - ○ 300 words max for the discussion
    - ○ Please only use the provided data for model building
    - ○ You can create/add additional features based on the dataset
    - ○ Hint - Use the insights from the data exploration/domain knowledge/statistical models to consider whether to create more feature columns, or whether to remove some columns
2. Write code to create/transform the columns based on your discussion above

[2] This is an open question in which you would need to decide what columns to use as features and what transformation(s) would be required for each feature. Include references when you use arguments from third parties or generative AI tools.

2.2 Preparing Spark ML Transformers/Estimators for features, labels, and models (5%)

Write code to create Transformers/Estimators for transforming/assembling the columns you selected above in 2.1, and create ML model Estimators for Random Forest (RF) and Gradient-boosted tree (GBT) models. Create **two** pipelines.

- **Please DO NOT fit/transform the data yet.**

2.3 Training and evaluating models (25%)

1. Write code to split the data for training and testing purposes (80/20), using seed=2025. Use the transformer and estimators in 2.2 to create two ML pipelines.(5%)
2. Implement a customised evaluation metric Root Mean Squared Logarithmic Error (RMSLE), defined as: (10%)

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

ϵ *is the RMSLE score.*

n *is the total number of datapoing in the dataset.*

p_i *is your prediction target.*

a_i*is the actual target.log log* (x) *is the natural logarithm of* x*.*

3. Train your model with **both** pipelines[3], evaluate the performance for both models (i.e. show RMSLE scores) and save the **better** model (you need it for Part B of Assignment 2). (10%)

(Note: You may need to go through a few training loops or use more data to create a better-performing model.)

Part 3. Performance Tuning (15%)

Using the better model identified in Part 2, apply hyperparameter tuning techniques to further optimise its performance.

1. Use tools such as ParamGridBuilder and CrossValidator (or TrainValidationSplit) to search for the best combination of hyperparameters for your chosen model.
2. Document your process, including the parameters you chose to tune, the range of values you tested, and the evaluation metric you used for tuning. Briefly discuss the results and the impact on model performance (no word limit but keep it concise).
3. Save your tuned model for part B of the assignment (if it’s better than part B).

Notes: The meters table contains missing values due to sensor or network failures, which prevented the recording of these values. To create a better model, you may consider imputing those values.

Part 4: Data Ethics, Privacy, and Security (10%)

In the era of big data, the convergence of vast quantities of information from various sources raises critical questions related to data ethics, privacy, and security. For example, in the case

[3] Each model training might take from minutes to hours, depending on the complexity of the pipeline model, the amount of training data, the computing power of your laptop and the code efficiencies

of privacy, many companies are collecting much more data than they need from customers. In our case, we used a real-world data set with real customer information. How do you utilise those datasets with ethics, privacy and security in mind?
In this part of the assignment, you are tasked to explore these topics within the context of big data processing, drawing on contemporary research, real-world examples, and ethical considerations.
(word limit: 500 words, please include references)

(mandatory): Define the concepts of data ethics, privacy, and security within the big data domain.
(Choose one or more topics, you can also create your own topic) Explain the significance of these issues in today's data-driven world.

Data Ethics:

- Analyse how data ethics can influence big data processing;
- Examine real-world examples of how data ethics has been handled, both positively and negatively.
- Analyse the balance between technological advancements and ethical responsibilities

Data Privacy:

- Discuss the challenges and importance of maintaining privacy in big data.
- Investigate regulations and laws that govern data privacy, such as GDPR.
- Evaluate tools and techniques used to ensure privacy, and suggest improvements or new methodologies.

Data Security:

- Explore the potential security risks associated with big data processing.
- Assess the measures currently in place to secure big data, including encryption, authentication, and authorisation.

(**mandatory**)Summarise the key findings of your analysis.

Submission

You should submit your final version of the assignment solution online via Moodle.
You must submit the files created:

- Your jupyter notebook file A2A_authcate.ipynb
- **A pdf file** saved from jupyter notebook with all output following the file naming format as follows: **A2A_authcate.pdf**

Note that both submitted (ipynb and pdf) files will be scanned using plagiarism detection software. The highest similarity score among students may be interviewed to prove the originality of the task.

Assignment Marking Rubric

Detailed mark allocation is available in each task. For complex tasks and explanation questions, you will receive marks based on the quality of your work.

In your submission, the jupyter notebook file should contain the **code and its output**. It should follow *programming standards, readability of the code, and organisation of code*. Please find the PEP 8 -- Style Guide for Python Code for your reference. Here is the link: https://peps.python.org/pep-0008/ Penalty applies if your code is hard to understand with insufficient comments.

Other Information

Where to get help

You can ask questions about the assignment in the Assignments section in the Ed Forum, which is accessible on the unit's Moodle Forum page. This is the preferred venue for assignment clarification-type questions. You should check this forum regularly, as the responses of the teaching staff are "official" and can constitute amendments or additions to the assignment specification. Also, you can attend scheduled consultation sessions if the problem and the confusion are still unresolved.

Searching and learning on commercial websites/forums (e.g. Quora, Stack Overflow) is allowed. However, you should not post/ask assignment questions on those forums.

Plagiarism and collusion

Plagiarism and collusion are serious academic offences at Monash University. Students must not share their work with any other students. Students should consult the policy linked below for more information.

https://www.monash.edu/students/academic/policies/academic-integrity

See also the video linked on the Moodle page under the Assignment block.

Students involved in collusion or plagiarism will be subject to disciplinary penalties, which can include:

- The work not being assessed
- A zero grade for the unit
- Suspension from the University
- Exclusion from the University

Late submissions and Special Consideration

ALL Special Consideration, including within the semester, is now handled centrally. This means that students MUST submit an online Special Consideration form via Monash Connect. For more details, please refer to the **Unit Information** section in Moodle.

There is a **5% penalty per day including weekends** for a late submission. Also, the cut-off date is 7 days after the due date. No submission will be accepted (i.e. zero mark) after the cut-off date unless you have a special consideration.

Mark Release and Review

- Mark will be released within 10 business days after the submission deadline.
- Reviews and disputes regarding the mark will be accepted a maximum of 7 days after the release date (including weekends).

Generative AI Statement

As per the University's policy on the guidelines and practices pertaining to the usage of Generative AI:

AI & Generative AI tools may be used SELECTIVELY within this assessment.

Where used, AI must be used responsibly, clearly documented and appropriately acknowledged (see Learn HQ).

Any work submitted for a mark must:

1. Represent a sincere demonstration of your human efforts, skills and subject knowledge that you will be accountable for.
2. Adhere to the guidelines for AI use set for the assessment task.
3. Reflect the University's commitment to academic integrity and ethical behaviour.

Inappropriate AI use and/or AI use without acknowledgement will be considered a breach of academic integrity.

The teaching team encourage students to apply their own critical thinking and reasoning skills when working on the assessments with assistance from GenAI. Generative AI tools may produce inaccurate content, and this could have a negative impact on students' comprehension of big data topics.

Data source acknowledgement:

The dataset is a combination based on several real-world and synthetic datasets. We thank the original contributors of the datasets.

1. Meters Table (meters.csv)

This table contains the time-series data for energy consumption for each meter in each building.

Column Name	Data Type	Description
building_id	Integer	A unique identifier for the building where the meter is located. Foreign key to the buildings table.
meter_type	Char(1)	The type of energy being measured from 4 different type of meters. (e.g., 'e', ‘c’, 's', 'h').
ts	Timestamp	The exact timestamp of the meter reading.
value	Decimal	The energy consumption reading value. The aggregated value of 4 meters is target variable for prediction.
row_id	Integer	A unique identifier for each individual reading in the table.

2. Buildings Table (buildings.csv)

This table contains the static metadata and descriptive features for each building.

Column Name	Data Type	Description
site_id	Integer	An ID for the geographical location of the building. Foreign key to the weather table.
building_id	Integer	A unique identifier for the building.
primary_use	String	The primary function of the building (e.g., 'Education', 'Office', 'Retail').
square_feet	Integer	The gross floor area of the building in square feet.
floor_count	Integer	The number of floors in the building.
row_id	Integer	A unique identifier for each building record in the table.
year_built	Integer	The year the building was constructed.
latent_y	Decimal	A latent feature with unknown meaning.
latent_s	Decimal	A latent feature with unknown meaning.

latent_r	Decimal	A latent feature with unknown meaning.

3. Weather Table (weather.csv)

This table contains the time-series weather data for each geographical site.

Column Name	**Data Type**	**Description**
site_id	Integer	A unique identifier for the geographical location/site. Each site may reside in different country with different seasons.
timestamp	Timestamp	The timestamp of the weather measurement.
air_temperature	Decimal	The temperature of the air in degrees Celsius.
cloud_coverage	Integer	The portion of the sky covered by clouds (e.g., from 0-8 oktas). May contain missing values.
dew_temperature	Decimal	The dew point temperature in degrees Celsius.
sea_level_pressure	Decimal	The air pressure in millibars, adjusted to sea level. May contain missing values.
wind_direction	Integer	The direction of the wind in degrees from true north (0-360). May contain missing values.
wind_speed	Decimal	The speed of the wind in meters per second.