Monash University

FIT5202 - Data processing for Big Data (S2 2025)

Assignment 1: Analysing Australian Property Market Data

Due Date: 23:55 Friday 05/Sep/2025 (End of week 6)

Weight: 15% of the final marks

Background

Australia's property market is a cornerstone of its economy and a topic of national obsession. Like many other nations, homeownership is a deeply ingrained part of the Australian dream. The market is vast and diverse, encompassing everything from sprawling houses in suburban areas to high-rise apartments in bustling city centres and rural homesteads in the outback.

A complex interplay of factors influences the market's performance, but the market has a reputation for resilience and long-term growth. Despite occasional downturns, property values have historically demonstrated an upward trend over the long term. This has made real estate a popular investment vehicle for many Australians, leading to a significant number of properties being held by investors rather than owner-occupiers.

Traditionally, property market analysis relied on a combination of anecdotal evidence, local agent expertise, and government-published reports. While these sources provide valuable insights, they often lack the scale and granularity needed to understand the market's complexities fully. The advent of the internet and digital technology has fundamentally changed this landscape.

Today, a vast amount of data is generated at every stage of the property lifecycle. This includes listing data from real estate portals, sales records from government land registries, loan application data from financial institutions, demographic information from census data, and a myriad of other sources. This sheer volume of information has given rise to the field of big data processing, which is revolutionising how we analyse the Australian property market.

Big data processing provides the tools and techniques to handle the massive volume, variety, and velocity of modern property data. It's not just about having more data; it's about being able to process, analyse, and derive meaningful insights from it at a scale and speed that was previously impossible.

This semester, in our big data processing unit, we will explore the Australian property market. We will perform a simple historical data analysis using Apache Spark in A1.

In this assignment(A1), we will perform historical data analysis using Apache Spark. We will use RDD, DataFrame and SQL API learnt from week 1-4.

The Dataset

The dataset can be downloaded from Moodle.

You will find the following files after extracting the zip file:

1) nsw_property_price.csv: Main CSV file containing the core dataset.
2) council.json, property_purpose.json, and zoning.json contain information about those fields (id and name mapping).

The metadata of the dataset can be found in the appendix at the end of this document.

Assignment Information

The assignment consists of three parts: Working with **RDD**, Working with **Dataframes**, and Comparison of two forms of Spark abstractions. In this assignment, you are required to implement various solutions based on RDDs and DataFrames in PySpark for the given queries related to property market data analysis.

Getting Started

- Download your dataset from Moodle.
- Download a template file for submission purposes:
    - ***A1_template.ipynb*** file in Jupyter notebook to write your solution. Rename it into the format (for example: ***A1_xxx0000.ipynb*****.** This file contains your code solution(xxx0000 is your authcode).
- For this assignment, you will use Python 3+ and PySpark 3.5.0. (The environment is provided as a Docker image, the same one you use in labs.)

Part 1: Working with RDDs (30%)

In this section, you need to create RDDs from the given datasets, perform partitioning in these RDDs and use various RDD operations to answer the queries.

1.1 Data Preparation and Loading (5%)

1. Write the code to create a SparkContext object using SparkSession. To create a SparkSession, you first need to build a SparkConf object that contains information about your application. Use Melbourne time as the session timezone. Give your application an appropriate name and run Spark locally with **4 cores on your machine**.
2. Load the CSV and JSON files into multiple RDDs.
3. For **each** RDD, remove the header rows and display the total count and the first 8 records.
4. Drop records with invalid information: **purpose_id** or **council_id** is null, empty, or 0.

1.2 Data Partitioning in RDD (15%)

1. For **each** RDD, using Spark's default partitioning, print out the total number of partitions and the number of records in each partition (5%).

2. Answer the following questions:
    a. How many partitions do the above RDDs have?
    b. How is the data in these RDDs partitioned by default when we do not explicitly specify any partitioning strategy? Can you explain why it is partitioned in this way?
    c. Assuming we are querying the dataset based on **property price**, can you think of a better strategy for partitioning the data based on your available hardware resources?

    Write your explanation in Markdown cells. (5%)
3. Create a user-defined function (UDF) to transform the date strings from ISO format (YYYY-MM-DD) (e.g. 2025-01-01) to Australian format (DD/Mon/YYYY) (e.g. 01/Jan/2025), then call the UDF to transform two date columns **(iso_contract_date and iso_settlement_date**) to **contract_date** and **settlement_date**(5%)

1.3 Query/Analysis (10%)

For this part, write relevant **RDD operations** to answer the following questions.

1. Extract the Month (Jan-Dec) information and print the total number of sales by contract date for each Month. (5%)
2. Which **5** councils have the largest number of houses? Show their name and the total number of houses. (Note: Each house may appear multiple times if there are more than one sales, you should only count them once.) (5%)

Part 2. Working with DataFrames (45%)

In this section, you need to load the given datasets into PySpark DataFrames and use *DataFrame functions* to answer the queries.

2.1 Data Preparation and Loading(0%)

1. Load the CSV/JSON files into separate dataframes. When you create your dataframes, please refer to the metadata file and think about the appropriate data type for each column.
2. Display the schema of the dataframes.

When the dataset is large, do you need all columns? How to optimise memory usage? Do you need a customised data partitioning strategy? (**Note: Think about those questions, but you don't need to answer them.**)

2.2 Query/Analysis (45%)

Implement the following queries using **dataframes**. You need to be able to perform operations like transforming, filtering, sorting, joining and group by using the functions provided by the DataFrame API. **For each task, display the first 5 results where no output is specified**.

1. The area column has two types: (H, A and M): 1 H is one hectare = 10000 sqm, 1A is one acre = 4000 sqm, 1 M is one sqm. Unify the unit to sqm and create a new column called **area_sqm**. (5%)
2. The top five property types are: Residence, Vacant Land, Commercial, Farm and Industrial. However, for historical reason, they may have different strings in the database. Please update the primary_purpose with the following rules:
    a) Any purpose that has “HOME”, “HOUSE”, “UNIT” is classified as “Residence”;
    b) “Warehouse”, “Factory”, “INDUST” should be changed to “Industrial”;
    c) Anything that contains “FARM”(i.e. FARMING), should be changed to “Farm”;
    d) “Vacant”, “Land” should be “Vacant Land”;
    e) Anything that has “COMM”, “Retail”, “Shop” or “Office” are “Commercial”.
    f) All remaining properties, including null and empty purposes, are classified as “Others”.

    Show the count of each type in a table. (10%)

    (note: Some properties are multi-purpose, e.g. “House & Farm”, it’s fine to count them multiple times.)
3. Find the top 20 properties that make the largest value gain, show their address, suburb, and value increased. To calculate the value gain, the property must have been sold multiple times, “value increase” can be calculated with **the last sold price – first sold price**, regardless the transactions in between. Print all 20 records. (10%)
4. For each **season**, plot the **median house price** trend over the years. Seasons in Australia are defined as: (Spring: Sep-Nov, Summer: Dec-Feb, Autumn: Mar-May, Winter: Jun-Aug). (10%)
5. (Open Question) Explore the dataset freely and plot **one diagram of your choice**. Which columns (at least 2) are highly correlated to the sales price? Discuss the steps of your exploration and the results. (No word limit, please keep concise.) (10%)

Part 3: RDD vs DataFrame vs Spark SQL (25%)

Implement the following complex queries using RDD, DataFrame in SparkSQL **separately(choose two)**. Log the time taken for each query in each approach using the “%%time” built-in magic command in Jupyter Notebook and **discuss the performance difference between these 2 approaches of your choice**.

(notes: You can write a multi-step query or a single complex query, the choice is yours. You can reuse the data frame in Part 2.)

(Complex Query)

A property investor wants to understand whether the property price and the settlement date are correlated. Here is the conditions:

1) The investor is only interested in the last 2 years of the dataset.
2) The investor is looking at **houses under $2 million**.
3) Perform a bucketing of the settlement date (settlement – contract date range (15, 30, 45, 60, 90 days).
4) Perform a bucketing of property prices in $500K(e.g. 0-$500K, $500K-$1M, $1M-$1.5M, $1.5-$2M)
5) Count the number of transactions in each combination and print the result in the following format (note: It's fine to count the same property multiple times in this task, it's based on sales transactions).

Year	Price Range	Settlement Days	Count
….	….	….	….

(Note: You shall show the full table with 40 rows, 2 years *4 price bucket * 5 settlement bucket; 0 count should be displayed as 0, not omitted.)

a) Implement the above query using two approaches of your choice separately and print the results. (Note: Outputs from both approaches of your choice are required, and the results should be the same.). (20%)

b) Which one is easier to implement, in your opinion? Log the time taken for each query, and observe the query execution time, among DataFrame and SparkSQL, which is faster and why? Please include proper references. (Maximum 500 words.) (5%)

Submission

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

- Your jupyter notebook file **(e.g., A1_authcate.ipynb**).
- **A pdf file** saved from jupyter notebook with all output following the file naming format as follows: **A1_authcate.pdf**

Note that both submitted (jupyter 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 work.

Assignment Marking Rubric

For each task individually, you'll be marked based on the quality of your work on a 3-level scale (0%, 50% and 100%).

- 0%: No attempt or incorrect answer with poor attempt;
- 50%: Partial mark for a good attempt but incorrect result;
- 100%: Full mark for correct attempt.

In your submission, the jupyter notebook file should contain the **code and its output**. It should follow *programming standards, readability of the code, organization 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 up to 10% applies if your code is hard to understand with insufficient comments.**

Late submissions

Late Assignments or extensions will not be accepted unless you submit a special consideration form. ALL Special Consideration, including within the semester, is now to be submitted 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.

A late submission is subject to a 5% penalty per day, including weekends. The cut-off date is 7 days after the due date. No submission (i.e. 0 mark) will be accepted after the cut-off date unless you have a special consideration.

If you submit the wrong file and ask for the submission to be reopened, the new timestamp of resubmission will be used to calculate late penalty.

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).

Other Information

Where to get help

You can ask questions about the assignment in the Assignments section in the Ed Forum 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 not solved.

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

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 encourages students to apply their own critical thinking and reasoning skills when working on the assessments with an assistant from GenAI. Generative AI tools may produce inaccurate content, which could negatively impact students' comprehension of big data topics.

Data source acknowledgement:

The dataset is a remix based on several real-world dataset. All name, age, dob, salary etc. are randomly generated synthetic datasets.

Appendix: Metadata of the Dataset Schema

Column Name	Description
id	A unique identifier for the transaction record itself.
property_id	Unique identifier for a specific property. One property may have multiple sales at different date.
purchase_price	The final sale price of the property.
address	The full street address of the property.
post_code	The postal code for the property's location.
property_type	The type of property (e.g., apartment/unit, house).
strata_lot_number	The strata lot number for properties with strata title.
property_name	The name of the property or building.
area	The size of the property.
area_type	The type of area (e.g., floor area, land area).
iso_contract_date	The date the contract was signed, in ISO 8601 format.
iso_settlement_date	The date the transaction was settled, in ISO 8601 format.
nature_of_property	The nature of the property.
legal_description	A formal legal description of the property.
council_id	The ID of the local council to which the property belongs.
purpose_id	An identifier to classify the purpose of the property record.
zone_id	The land zoning classification for the property.

The JSON files have a simple mapping from id to name; for example, council.json has council_id and council_name as its data.