Monash University

FIT5202 - Data processing for Big Data

Assignment 2A: Building models to predict future retail sales

Due: Monday, Jan 30, 2023, 11:55 PM (Local Campus Time)
Worth: 10% of the final marks

Background

MelbourneGig is a start-up incubated at Monash University to provide services to the customers in the Retail & Sales industry. The team would like to hire us as the *Analytics Engineer* to analyse the feature, sales, and stores open data using big data tools.

Building on the findings from assignment 1, we will need to develop machine learning models further to predict the stores' weekly sales. In this part A of the assignment, we would process the static data and train machine learning models based on them. In addition, the machine learning models would be further integrated into the streaming platform using Apache Kafka and Apache Spark Streaming to perform prediction to recommend stores' future weekly sales.

Required Datasets (available in Moodle):

- Three data files
  - Features.csv
  - Sales.csv
  - Stores.csv
- A Metadata file is included, which contains the information about the dataset.
- These files are available in Moodle under the Assessment 2A data folder

Information on Dataset

The data is available on the website:
https://www.kaggle.com/datasets/manjeetsingh/retaildataset.

Please refer to the given website for more detailed information on the dataset.

What you need to achieve

The MelbourneGig company requires us to build models for predicting whether the sales per unit (Weekly_Sales of store / Size > 8.5) would go above the threshold of 8.5 and also predict the possible weekly sales. We would need binary classification models and regression models.

Use case 1	Predict whether the sales per unit (Weekly_Sales of store / Size > 8.5) would go above the threshold of 8.5	Binary classification
Use case 2	Predict the possible weekly sales	Regression

- To build the binary classification models, use the column “Weekly_Sales” and “Size” to create a binary label
- To build the regression models, use the column “Weekly_Sales” as your label

Architecture

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

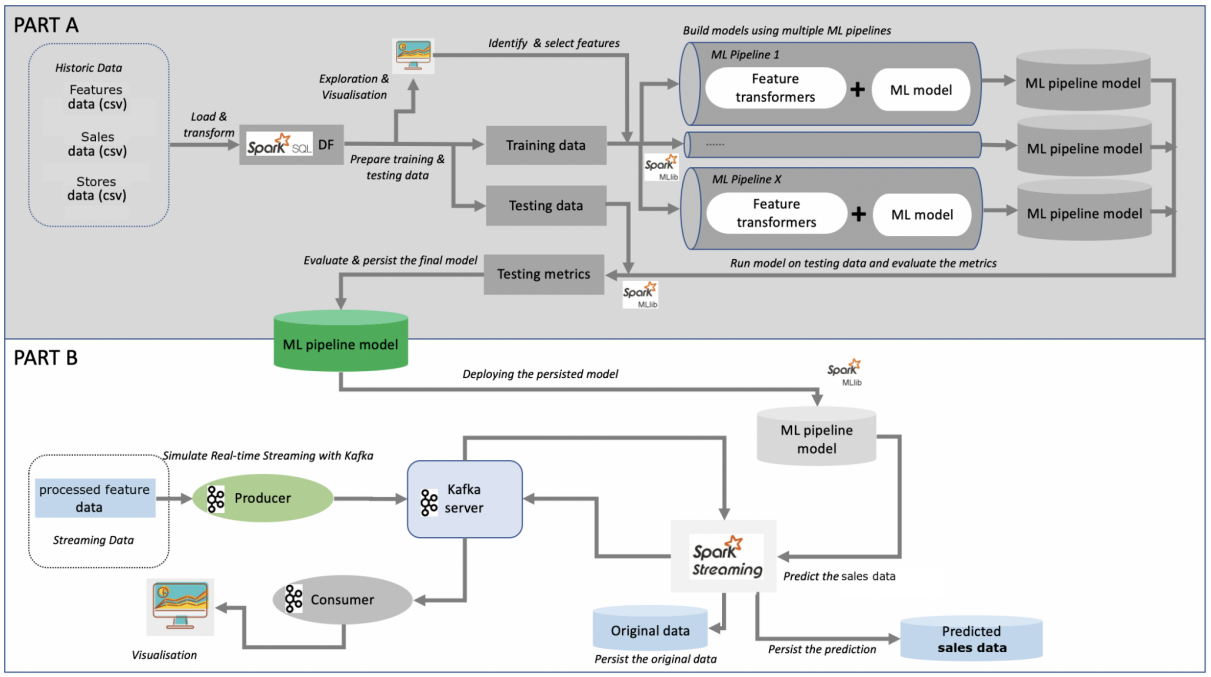


Fig 1: Overall Architecture for Assignment 2

In both parts, for the data pre-processing, and the machine learning processes, you are required to implement the solutions using PySpark SQL / MLlib / ML packages. For data visualisations, excessive usage of Pandas for data processing is discouraged. Please follow the steps to document the processes and write the codes in the Jupyter Notebook.

Getting Started

- Download the datasets from Moodle.
- Download two template files for submission purposes:
    - ***A2A_template.ipynb*** file in Jupyter notebook to write your

solution. Rename it into the format (for example, ***A2A_glii0039.ipynb*.** This file contains your code solution.

- Download document file ***A2A_template.docx*** to explain your jupyter notebook code (.ipynb) and convert it into pdf before submission. This file contains your code explanation in detail regarding the subsequent codes submitted above. The file naming format example is as follows: ***A2A_glii0039.pdf***

- You will be using Python 3+ and PySpark 3.3.0 for this assignment (This environment will be automatically set up if you follow the steps in moodle (Unit Information >> Software, Documentation, and Resources)).

IMPORTANT:

Please answer each question using **BOTH codes and their output** in your jupyter notebook file.

Acknowledge any ideas or codes you referenced from others in the documentation part (pdf document).

Your notebook runtime should be approximately maximum within 20 minutes using (e.g., Using 4 CPU cores). The long-running notebook would result in a mark deduction.

1 Data Loading and exploration (24%)

In this section, you will need to load the given datasets into PySpark DataFrames and use *DataFrame functions* to process the data. Excessive usage of Spark SQL or pandas is discouraged. For plotting, different visualisation packages can be used, but you need to ensure that you have included instructions to install the additional packages and the installation will be successful in the provided docker container.

1.1 Data Loading (10%)

1. Write the code to get a SparkSession. For creating the SparkSession, you need to use a SparkConf object to configure the Spark app with a proper application name, to enable the maximum partition size not exceed 10MB, and to run locally with as many working processors as local cores on your machine[1]

2. Write code to define the data schema for features, sales, and stores datasets, following the data types suggested in the metadata file[2].

3. Using predefined schema, write code to load the features, sales, and stores csv files into separate dataframes. Print the schemas for all of the dataframes.

---

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

[2] In this assignment, the "Date" should be directly read as Date format, instead of reading as String in assignment1. Sample usage of schema for reading CSV file can be found in https://docs.databricks.com/data/data-sources/read-csv.html

1.2 Exploring the data (14%)

1. Write code to show the total ‘null’ counts for each column in all three dataframes.

2. For each feature, sales, and stores dataframe, write code to show the basic statistics (including count, mean, stddev, min, max, 25 percentile, 50 percentile, 75 percentile) for each numeric column. For each non-numeric feature in each dataframe, display the top-5 values and the corresponding counts, except for the columns of “Store,” “Dept," and "Date."

3. Write code to display a histogram to show the distribution of the weekly sales for stores with log scale for the frequency axis. Describe what you observe from the plot.
    - Apart from that, Draw a line plot to show the trend of the average weekly sales of the month based on the different stores.

4. Explore the data provided and write code to present two plots[3] worthy of presenting to the MelbourneGig company, describe your plots and discuss the findings from the plots
    - Hint - 1: you can use the basic plots (e.g., histograms, line charts, scatter plots) for the relationship between a column and the label; or more advanced plots like correlation plots; 2: if your data is too large for the plotting, consider using sampling before plotting
    - 150 words max for each plot’s description and discussion
    - Please do not repeat the plots in task 1.2.3.
    - Please only use the provided data for visualisation

2. Feature extraction and ML training (70%)

In this section, you will need to 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. Excessive usage of Spark SQL is discouraged.

2.1 Discuss the feature selection and prepare the feature columns (12%)

1. As we need to perform a one-step time-series prediction, meaning that the model’s prediction for the next weekly sales would be based on the previous weekly sales. The model will be used for future prediction. Based on the data exploration from 1.2 and considering the situation we have, discuss which importance of those features (For example, which features may be useless and should be removed, which feature has a great impact on the label column, which should be transformed) which features

[3] This is an open question, in which you would need to decide what plots to show.
- You can combine multiple features into one plot, but the plot should be clear to be seen, and do not contain an overwhelming amount of information.
- If you use subplots, each subplot would be considered as one plot, and the two-plot limit would allow only two subplots for each activity data.

you are planning to use? Discuss the reasons for selecting them and how you create/transform them[4]

- 400 words max for the discussion
- Please only use the provided data for model building
- Hint - things to consider include whether to create more feature columns, whether to remove some columns, using the insights from the data exploration/domain knowledge/statistical models

2. Write code to create the columns based on your discussion above
    - Use case 1: We need a model to predict stores that achieve the goals, which means the weekly sales of the store divide the Store size is greater than 8.5, create a column called "achieve_goal" and use 1 to label those achieved data, 0 for others' data.
    - Use case 2: Join the DataFrames for our one-step time-series weekly sales prediction for stores. You should ensure the weekly sales of the previous week and Store Type as the feature of our model. For other columns, you can choose based on your answer in 2.1.1.

2.2 Preparing Spark ML Transformers/Estimators for features, labels, and models (16%)

1. Write code to create Transformers/Estimators for transforming/assembling the columns you selected above in 2.1, and create ML model Estimators for Decision Tree and Gradient Boosted Tree model for each use case
    - **Please DO NOT fit/transform the data yet**

2. Write code to include the above Transformers/Estimators into pipelines
    - A maximum of two pipelines can be created for each use case
    - **Please DO NOT fit/transform the data yet**

2.3 Preparing the training data and testing data (4%)

1. Write code to split the data for training and testing purposes - use the data in 2010 and 2012 for training purposes and the half data in the 2011 year for training and others for testing purposes; then cache the training and testing data.

2.4 Training and evaluating models (38%)

Use case 1

[4] 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 reference when you use arguments from third parties.

1. For use case 1, write code to use the corresponding ML Pipelines to train the models on the training data from 2.3. And then use the trained models to predict the testing data from 2.3[5]

2. For both models' results in use case 1, write code to display the count of each combination of above-threshold/ below-threshold label and prediction label in formats like the screenshot below. Compute the AUC, accuracy, recall, and precision for the above-threshold/below-threshold label from each model testing result using pyspark MLlib/ML APIs
    - Discuss which metric is more proper for measuring the model performance on predicting above-threshold events in order to give the performers good recommendations while reducing the chance of falsely recommending a location.
    - Draw a ROC plot for any model you want.
    - Discuss which is the better model, and persist with the better model.

```
+------------+----------+-----+
|achieve_goal|prediction|count|
+------------+----------+-----+
|           0|       0.0|     |
|           1|       0.0|     |
|           0|       1.0|     |
|           1|       1.0|     |
+------------+----------+-----+
```

Use case 2

3. For use case 2, write code to use the corresponding ML Pipelines to train the models on the cache training data from 2.3. And then use the trained models to perform predictions on the testing data from 2.3[6]

4. For both models' results in use case 2, compute the RMSE, R-squared
    - Discuss which is the better model, and persist the better model.
5. Write code to print out the features with each corresponding feature importance for the GBT model, ranked the result based on feature importance.

3. Knowledge sharing (6%)

In addition to building the machine learning models, the IT manager from MelbourneGig would like to learn more about parallel processing. You are expected to combine the theory from the lecture and the observation from Spark UI or Spark source code to explain the ideas of data parallelism, and the result from parallelism using the KMeans clustering as an example

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

3.1 How many jobs are observed when training the KMeans clustering model following the code below? Provide a screenshot from Spark UI for running a simple KMeans model training from the provided data[7]

```
customer_df = spark.createDataFrame(
    [
        [0, 35.3, 37.5],
        [1, 41.4, -23.5],
        [2, 28.3, -13.3],
        [3, 09.5, -9.0],
        [4, 62.8, -18.23],
        [5, 63.8, -18.33],
        [6, 82.8, -17.23],
        [7, 52.8, -13.43],
        [8, 72.8, 48.23],
        [9, 65.8, 15.43],
        [10, 42.8, -13.23],
    ],
    ["ID", "Att_1", "Att_2"],
)

assembler = VectorAssembler(inputCols=["Att_1", "Att_2"], outputCol="features")
kmeans = KMeans(k=2).fit(assembler.transform(customer_df))
```

3.2 Combining the parallelism theory from lecture, Spark source code, and the Spark UI, explain whether data parallelism or result parallelism is being adopted in the implementation of KMeans clustering in Spark (5%)

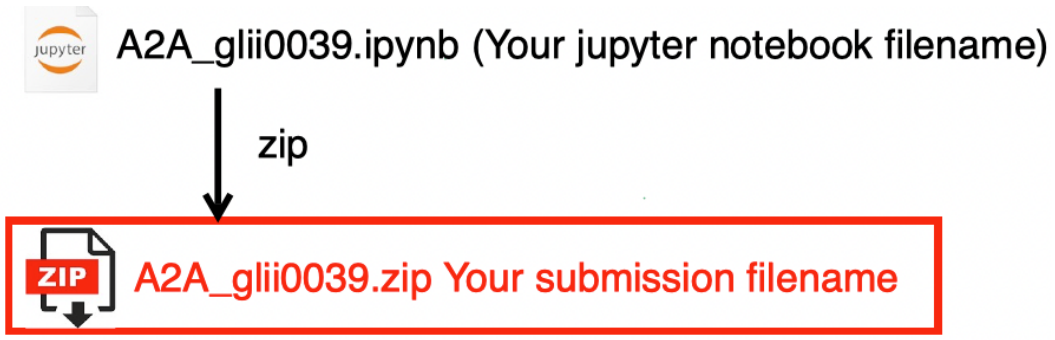
- 300 words max for the discussion
- Hint - you can also refer to the Spark source code on github https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/clustering/KMeans.scala

Submission

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

- **A Zip file** of your jupyter notebook file (e.g., A2A_username.zip contains A2A_username.ipynb). Note that the file naming format for both jupyter and zip files follows the following rules: A1_authcate.ipynb and A1_authcate.zip

[7] Data extracted from https://www.educba.com/pyspark-kmeans



- **A pdf file** following the file naming format as follows: **A1_authcate.pdf**

Note that both submitted (zip 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

The marking of this assignment is based on the quality of work you have submitted rather than just quantity. The marking starts from zero and goes up based on the tasks you have completed and their quality.

The marking rubric for both files is provided in moodle.

- The jupyter notebook file contains the **code and its output**. *It should follow programming standards, readability, and code organization*. Please find the PEP 8 -- Style Guide for Python Code for your reference. Here is the link: https://peps.python.org/pep-0008/ .
- The pdf file contains the *presentation of the assignment explanation of the jupyter notebook codes* (pipeline, variable input, output, comments, description, etc.). The details of the marking criteria are provided in the marking rubric.

Other Information

Where to get help

You can ask questions about the assignment on the Assignments section in the Ed Forum, accessible from the on the unit's Moodle Forum page. This is the preferred venue for assignment clarification-type questions. It is not permitted to ask assignment questions on commercial websites such as StackOverflow or other forms of forums.

You should check the Ed forum regularly, as the responses of the teaching staff are "official" and can constitute amendments or additions to the assignment specification. Also, you can visit the consultation sessions if the problem and the confusions 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

Late submissions

There is a **5% penalty per day including weekends** for the late submission.

Note: Assessment submitted more than 4 calendar days after the due date will receive a zero (0) mark for that assessment task. Students may not receive feedback on any assessment that receives a mark of zero due to a late-submission penalty.

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.