Monash University

FIT5202 - Data processing for Big Data 2025 SSB

Assignment 2: Building Models for Realtime Food Delivery Prediction

Weight: 30% (15% A2A+15% A2B)

Background

Food delivery services have become an integral part of modern society, revolutionising the way we consume meals and interact with the food industry. These platforms, accessible through websites and mobile apps, provide a convenient bridge between restaurants and consumers, allowing users to browse menus, place orders, and have food delivered directly to their doorstep with just a few taps. In today's fast-paced world, where time is a precious commodity, food delivery services offer an invaluable solution, catering to busy lifestyles, limited mobility, and the ever-present desire for convenience.

In the food delivery industry, accurate on-time delivery prediction is paramount. Big data processing allows companies to achieve this by analysing vast datasets encompassing order details, driver performance, real-time traffic, and even weather.

Objective of the Project

In the **first assignment (A1)**, we performed data analysis with the datasets to uncover key trends and patterns related to delivery times, order volumes, and other crucial metrics.

In **assignment 2A**, we will harness the power of Apache Spark's MLLib to construct and train machine learning models. We will focus on accurately and efficiently predicting delivery times.

Finally, assignment **2B** will utilise Apache Spark Structured Streaming and our ML model from 2A to process live data streams and dynamically make predictions.

Key Information

This is a two-part assignment (A2A and A2B) that requires staged submissions. In part A2A, you are going to use the provided dataset, complete the assignment tasks, and build your ML model; then, in part A2B, the trained ML model will be used in combination with streaming data to make real-time predictions.

A2A Due Date: **(23:55 Friday 31/Jan/2025, End of Week 5)**

A2B Due Date: **(23:55 Wednesday 5/Feb/2025, Mid of Week 6)**

Submission links can be found in Moodle.

Weight: 30% of Final Marks (15% each for 2A and 2B) A2A and A2B will be marked separately.

A2B has a compulsory interview/demo component, which will be conducted during the last lab. The teaching team only marks A2B submissions during your demo session. Failure to attend this demo will result in 0 marks (for A2B).
(Please pay attention to the unit announcement in the final teaching week. If you have an extension/special consideration, more demo sessions will be arranged.)

The Datasets:

- **order.csv**
- **driver.csv**
- **delivery_address.csv**
- **restaurant.csv**
- **new_order.csv (for A2B)**

What you need to achieve

Use Case 1 (A2A)	Based on the historical dataset, build a ML model that can predict food delivery time.	Regression
Use Case 2 (A2B)	Use streaming data to perform real-time prediction and visualise the results	Spark Structured Streaming

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.

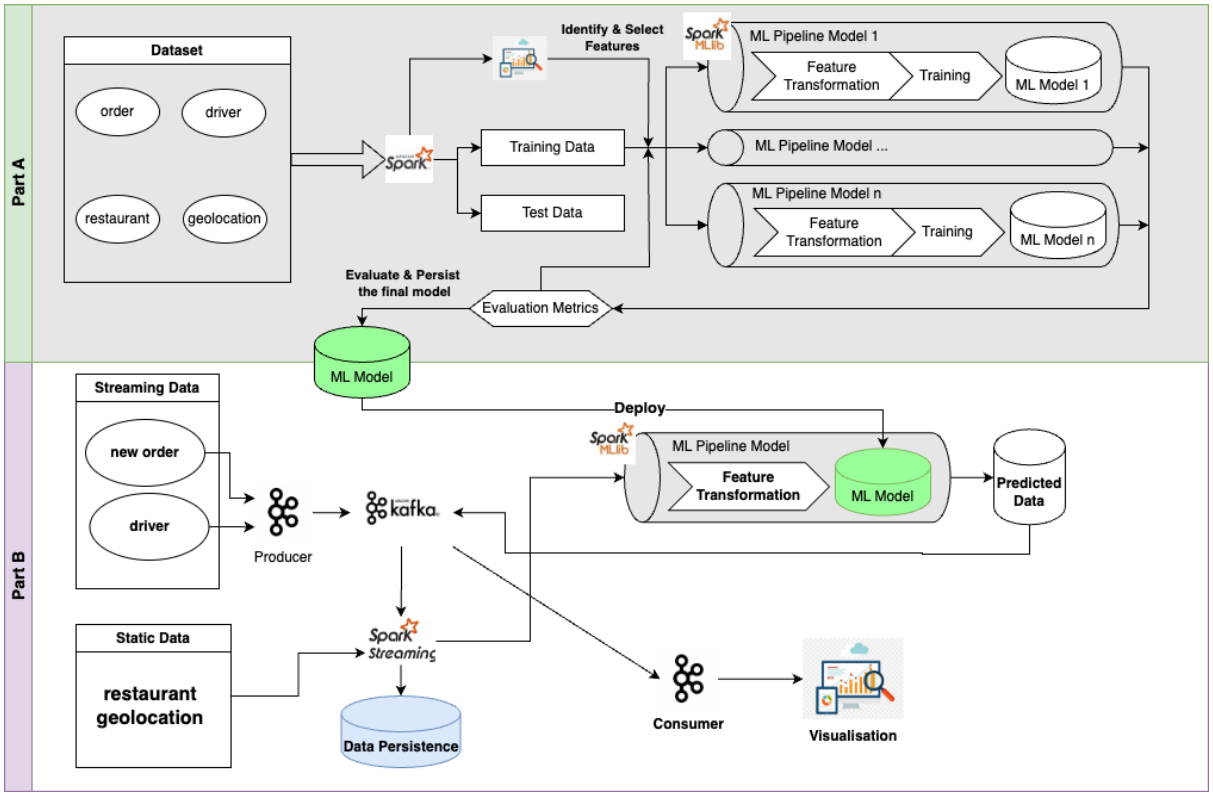

Fig 1: Overall Architecture for Assignment 2

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 codes 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 cells. 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 or appendix.

A2A Part 1: Data Loading, Transformation and Exploration (40%)

In this section, you must load the given datasets into PySpark DataFrames and use *DataFrame functions* to process the data. Spark SQL usage is discouraged, and you can only use pandas to format results. 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. Please use a SparkConf object to configure the Spark app with a proper application name, to ensure the maximum partition size does not exceed 16MB, and to run locally with **4** CPU cores on your machine[1]. (2%)
2. Write code to define the schemas for the datasets, following the data types suggested in the metadata[2]. Then, using predefined schemas, write code to load the CSV files into separate data frames. Print the schemas of all data frames. (3%)

1.2 Data Transformation to Create Features (10%)

Feature engineering involves transforming, combining or extracting information from the raw data to create more informative and relevant features that improve the performance of your ML models. In our food delivery use case, the order_ts is not very useful when it is treated as a timestamp. However, it provides more information if you perform transformation and extract valuable information from it, for example, extracting the day of the week (it may tell you how busy a restaurant is) or hours (peak hours may have bad traffic conditions).
(Note: Some tasks may overlap with A1, feel free to use/reuse your **own** code/UDF from A1.)

Perform the following tasks based on the loaded data frames and create a new one. We will refer to this as feature_df, but feel free to use your own naming. **(2% each) Please print 5 rows from the feature_df after each step.**

1. Extract the day of the week (Monday-Sunday) and hour of the day (0-23) from order_ts, and store the extract information in 2 columns.
2. Create a new boolean column (**isPeak**) to indicate peak/non-peak hours. (Peak hours are defined as 7-9 and 16-18 in 24-hour format.)
3. Join the geolocation data frame of the restaurant and delivery location, get suburb information and add two columns.
4. Join data frames to add restaurant information to the feature_df: primary_cuisine, latitude, longitude, suburb and postcode.
5. Add columns you deem necessary from the dataset (at least one column is required). (hint: delivery driver's vehicle type may affect the delivery time.)

1.3 Exploring the data (25%)

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

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

each non-numeric column, display the top-5 values and the corresponding counts; c) For each boolean column, display the value and count. **(5%)**

2. Explore the dataframe and write code to present **two plots**, describe your plots and discuss the findings from the plots. **(20%)**
    - One of the plots must be related to our use case (predicting delivery time).
    - Hint 1: You can use basic plots (e.g., histograms, line charts, scatter plots) to show the relationship between a column and the label or use more advanced plots like correlation plots.
    - Hint 2: If your data is too large for plotting, consider using sampling before plotting.
    - 150 words max for each plot's description and discussion
    - Feel free to use any plotting libraries: matplotlib, seabon, plotly, etc.

A2A Part 2. Feature extraction and ML training (50%)

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, should not be used to process the data; however, it's fine to use them to display the result or evaluate your model.

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 plan to create/transform them[3].
    - 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, whether to remove some columns
2. Write code to create/transform the columns based on your discussion above.

2.2 Preparing Spark ML Transformers/Estimators for features, labels, and models (10%)

1. 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) model.
    - **Please DO NOT fit/transform the data yet.**
2. Write code to include the above Transformers/Estimators into two pipelines.
    - **Please DO NOT fit/transform the data yet.**

(Some students may be confused about the differences between 2.1.2 and 2.2.1. Task 2.1.2 is for the new or customised feature you discussed and created, while 2.2.1 is for the "standard"

[3] 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.

features in the dataset like road_condition or weather_condition, which obviously affect the delivery time.)

2.3 Preparing the training data and testing data (5%)

1. Write code to split the data for training and testing, using **2025** as the random seed. You can decide the train/test split ratio based on the resources available on your laptop. Note: Due to the large dataset size, you can use random sampling (say 20% of the dataset).

2.4 Training and evaluating models (25%)

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[4]
2. For both models (RF and GBT): with the test data, decide on which metrics to use for model evaluation and discuss which one is the better model (no word limit; please keep it concise). You may also use a plot for visualisation (not mandatory).
3. Save the better model (you'll need it for A2B).

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

A2A Part 3. Hyperparameter Tuning and Model Optimisation (10%)

3.1 Apply the techniques you have learnt from the labs, for example, CrossValidator, TrainValidationSplit, ParamGridBuilder, etc., to perform further hyperparameter tuning and model optimisation.

The assessment is based on the quality of your work/process, not the quality of your model. Please include your thoughts/ideas/discussions.

(A2 Part B Specification: To be released in Week 5)

Submission A2A

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**
- **A2A Due date: (23:55 Friday 31/Jan/2025)**

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.

[4] 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

Submission A2B

You should submit your final version of the assignment solution via Moodle. You must submit the following:

- A **zip** file named based on your authcate name (e.g. abcd1234). The zip file should contain
    - ***Assignment-2B-Task1_producer_authcate*.ipynb**
    - ***Assignment-2B-Task2_spark_streaming_authcate*.ipynb**
    - ***Assignment-2B-Task3_consumer_authcate.ipynb***

    The file in submission should be a ZIP file and *not any other kind of compressed folder (e.g. .rar, .7zip, .tar)*. Please do not include the data files in the ZIP file.
    The A2B due date is **23:55 Wednesday 5/Feb/2025**

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 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 for which you will be accountable.
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 assistance from GenAI. Generative AI tools may produce inaccurate content, which could have a negative impact on students' comprehension of big data topics.

Appendix: Metadata of the Dataset Schema

(note: Some self-explanatory columns are not included. i.e. street_name, postcode)

delivery_address.csv	
gid	ID of deliver address geolocation
latitude	Latitude, Decimal(8,6)
longitude	Longitude, Decimal(8,6)
geom	Geometry point on maps
delivery_id	ID of a delivery address, this ID shall be used in join queries.
driver.csv	
driver_id	Unique identifier of delivery person/driver
age	Delivery driver's age (Integer), range is 18-60.
rating	Overall rating of the driver (float, 0-5 scale)
year_experience	Years of delivery experience, which may affect delivery speed
vehicle_condition	A driver's vehicle condition(from 0 - Good, 1 - Fair, 2 - Poor)
type_of_vehicle	Motorcycle, Scooter, electric_scooter, etc. (String)
order.csv	
order_id	Unique identifier of an order
delivery_person_id	Unique ID of the driver delivering an order
order_ts	timestamp when an order is placed
ready_ts	timestamp when a restaurant finishes preparing an order, i.e. ready for the delivery driver to pick up.
weather_condition	Weather condition at the time of order (Sunny, Windy, Storm, etc.) (String)
road_condition	Road traffic conditions (Low, Medium, Jam, etc.)
type_of_order	Snacks, Meal, Drinks, etc. (String)
order_total	Total value of the order.
delivery_time	Time taken for the delivery, not including preparation time. This column can be used as our label column.
Travel_distance	Total travel distance for the delivery.

restaurant_id	ID of a restaurant.
delivery_id	ID of a delivery address.
restaurants.csv	
row_id	Row id of the restaurant in database (not used as a dataset).
Restaurant_code	Internal code of a restaurant
Chain_id	If a restaurant belongs to a chain (empty if not).
Primary_cuisine	Primary Cuisine of the restaurant
geom	Geometry point of the restaurant
Restaurant_id	ID of a restaurant (used for join as primary key).