Monash University

FIT5202 - Data processing for Big Data

Assignment 2A: Building models to predict pedestrian traffic

Due: Sunday, Jan 31, 2021, 11:55 PM (Local Campus Time)
Worth: 10% of the final marks

Background

MelbourneGig is a start-up incubated in Monash University to provide services to the performers in the Music & Entertainment industry. The team would like to hire us as the *Analytics Engineer* to analyse the pedestrian count open data from the City of Melbourne using big data tools.

Building on top of the findings from assignment 1, we will need to further develop machine learning models to predict the pedestrian traffic. In addition, the machine learning models would be further integrated into the streaming platform using Apache Kafka and Apache Spark Streaming to perform prediction, in order to recommend busy locations for street art performers. In this part A of the assignment, we would process the static data and train machine learning models based on them.

Required Datasets (available in Moodle):

- Two data files
  - Pedestrian_Counting_System_-_Monthly_counts_per_hour.csv
  - Pedestrian_Counting_System_-_Sensor_Locations.csv
- A Metadata file is included which contains the information about the dataset.
- These files are available in Moodle under Assessment 2A data folder

Information on Dataset

Two data files from the City of Melbourne are provided, which captures the hourly count of pedestrians recorded by the sensors and the corresponding sensor locations. The data is also available on the website https://data.melbourne.vic.gov.au/.

What you need to achieve

The MelbourneGig company requires us to build models for predicting whether the potential count would go above the threshold of 2000 and also predicting the possible count. So we would need binary classification models and regression models.

Use case 1	Predict whether count would go above 2000 for the hours **between 9:00am and midnight**	Binary classification

Use case 2	Predict the possible count for the hours **between 9:00am and midnight**	Regression

- To build the binary classification models, use the column “Hourly_Count” to create a binary label
- To build the regression models, use the column “Hourly_Count” 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, building and persisting the machine learning models.

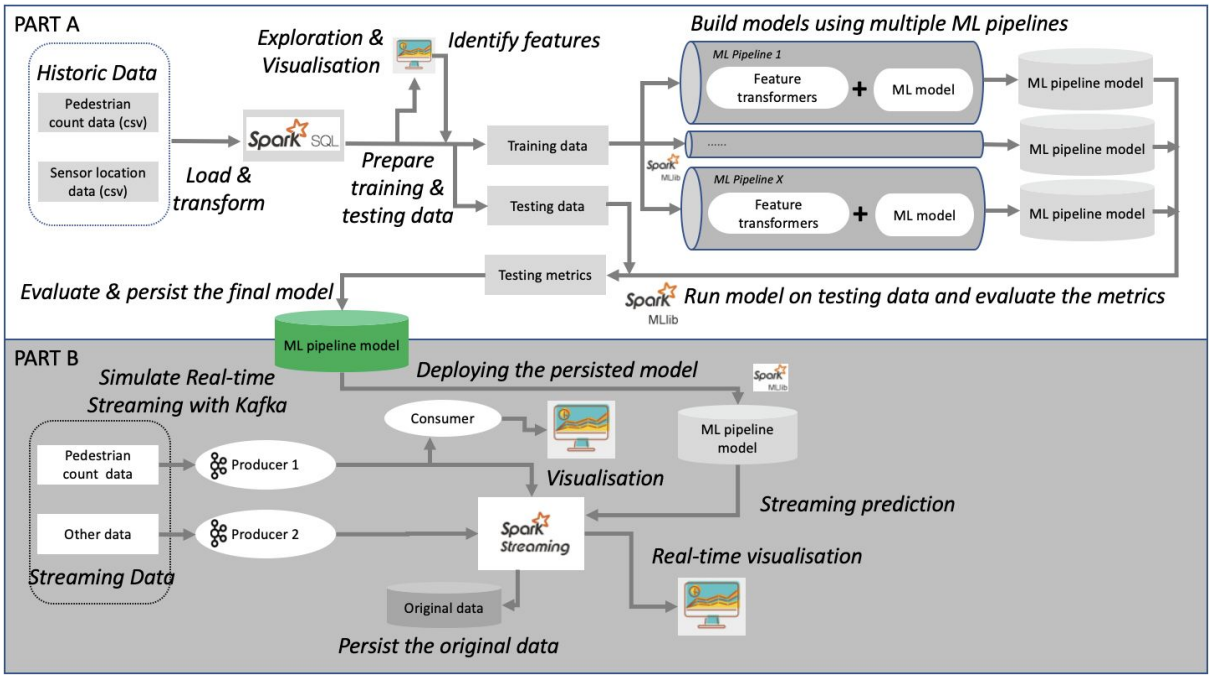

Fig 1: Overall architecture for assignment 2

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

Getting Started

- Download the datasets from moodle.
- Create an ***Assignment-2A.ipynb*** file in Jupyter Notebook to write your solution for processing data.

You will be using Python 3+ and PySpark 3.0+ for this assignment.

IMPORTANT:

Please answer each question using **BOTH codes and notebook markdown descriptions**.

In-line reference is required to acknowledge any ideas or codes that you referenced from others, or no marks would be awarded.

After completing the notebook, please also **export the notebook into a PDF file for submission** to Turnitin.

Your notebook runtime should be within 20min given the VM setup. Long-running notebook would result in 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 would be successful in the provided VM setup.

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 use UTC as the session timezone, 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 both pedestrian count CSV file and the sensor location file, following the data types suggested in the metadata file[2], with the exception of the "location" columns
    a. Use StringType for "location" column

3. Using predefined schema, write code to load the pedestrian count csv files into a dataframe, and load the sensor location csv file into another dataframe. Print the schema of both dataframe after transformation

4. Write code to create an additional column "above_threshold" in the pedestrian count dataframe to indicate whether the hourly count is above 2000 or below. Use label 0 for count below 2000, and label 1 for count above or equal to 2000

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

[2] In this assignment, the "Date_Time" should be directly read as Timestamp format and the "installation_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. For the pedestrian count dataframe, write code to show the basic statistics (including count, mean, stddev, min, max, 25 percentile, 50 percentile, 75 percentile) for each numeric column, except for the columns of “above_threshold”, “Date_Time”

2. Write code to show the count of above-threshold and below-threshold based on the column “above_threshold”
    - Do you see any class imbalance? Describe what you observe and discuss how it could impact classification

3. Write code to display a histogram to show the distribution of the hourly counts with log-scale for the frequency axis, and a line-plot to show the trend of the average daily count change by month
    - Describe what you observe from the plots.

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 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. Considering the data exploration from 1.2 and the nature of time-series data, we would be performing a one-step time-series prediction, meaning that the model’s prediction for the next hour count would be based on the previous pedestrian count(s)[4]. And the prediction is only needed for the hours **between 9:00am and**

[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.

[4] For some background information on how to perform time-series prediction, you can refer to https://doi.org/10.1016/j.eswa.2012.01.039, https://doi.org/10.1016/j.knosys.2018.10.009

midnight. Which columns are you planning to use as features? Discuss the reasons for selecting them and how you create/transform them[5]

- 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

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**

3. For the **Decision Tree classification** model you have created, explain the purposes of the hyperparameters of maxDepth and maxBin, and how they impact the model in theory and in this use case

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

1. Write code to split the data for training and testing purpose - use the data between 2014 and 2018 (including 2018) for training purpose and the data in 2019 as testing purpose[6]; then cache the training and testing data
   - Note: From task 2.1.1, the model training and the prediction is only needed for the hours **between 9:00am and midnight**.

2.4 Training and evaluating models (38%)

Use case 1

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 perform predictions on the testing data from 2.3[7]

[5] 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.
[6] The data from year 2020 is excluded due to the Covid19 lockdown in Melbourne.
[7] 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

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.
    - Discuss which is the better model, and persist the better model.

above_threshold	prediction	count
1	0.0	XXXX
0	0.0	XXXX
1	1.0	XXXX
0	1.0	XXXX

3. For the Decision Tree classification model in use case 1, write code to print out the leaf node splitting criteria and the top-3 features with each corresponding feature importance. Describe the result in a way that it could be understood by your potential users (e.g. street art performers)

4. How to improve the prediction for use case 1? Propose at least two suggestions, elaborate on why each could improve the models, and also briefly explain how to implement it with code snippets (no need for full implementation)
    - Hint - your suggestion should assume that model training is run on a Spark cluster with the data being in either Spark RDD or Dataframe format; you can also suggest using additional packages which are compatible with Spark.
    - 600 words max for the discussion

Use case 2

5. 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[8]

6. For both models' results in use case 2, compute the RMSE, R-squared
    - Discuss which is the better model, and persist the better model.

[8] 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. 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 result parallelism using the KMeans clustering as an example

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[9]

```
customer_df = spark.createDataFrame([
   (0,19,15,39),
   (0,21,15,81),
   (1,20,16,6),
   (1,23,16,77),
   (1,31,17,40),
   (1,22,17,76),
   (1,35,18,6),
   (1,23,18,94),
   (0,64,19,3),
   (1,30,19,72),
   (0,67,19,14),
   (1,35,19,99),
   (1,58,20,15)],
   ['gender', 'age', 'annual_income', 'spending_score'])

assembler = VectorAssembler(
   inputCols=['gender', 'age', 'annual_income', 'spending_score'],
   outputCol='features')
kmeans = KMeans(k=4).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

[9] Data extracted from the Mall Customer dataset of Udemy Machine Learning A-Z https://www.udemy.com/course/machinelearning/

Assignment Marking

The marking of this assignment is based on quality of work that you have submitted rather than just quantity. The marking starts from zero and goes up based on the tasks you have successfully completed and it's quality for example how well the code submitted follows *programming standards, code documentation, presentation of the assignment, readability of the code, reusability of the code, organisation of code and so on*. Please find the PEP 8 -- Style Guide for Python Code here for your reference.

Submission

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

- A PDF file (created from the notebook) to be submitted through Turnitin submission link
    - Use the browser's print function to save the notebook as PDF.
- A zip file of your Assignment 2A folder, named based on your authcate name (e.g. psan002). This should contain
    - **Assignment-2A.ipynb**

      This 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 assignment submission should be uploaded and finalised by Sunday, Jan 31, 2021, 11:55 PM (Local Campus Time).
- Your assignment will be assessed based on the contents of the Assignment 2 folder you have submitted via Moodle. When marking your assignments, we will use the same ubuntu setup (VM) as provided to you.

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 **10% penalty per day including weekends** for the late submission.

Note: Assessment submitted more than 7 calendar days after the due date will receive a mark of zero (0) for that assessment task. Students may not receive feedback on any assessment that receives a mark of zero due to 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.