School of Computing and Information Systems

MAST30034: Applied Data Science

Assignment 3 - Group Project

Due date: Final Player Submission no later than 11:59am on Monday 21st October 2019

Due date: Final Group Report 11:59pm on Thursday 24th October 2019

Weight: 50%

Project Overview

The aim of this project is to build a predictive model for a New York City Taxi driver to maximise their revenue (earnings). Your model will be evaluated in the context of a turn-based game using the real New York Taxi data to simulate a week as a taxi driver. Groups will compete against each other, with a portion of the marks being awarded based on the leader board of the final run of the game, which will take place after the submission deadline.

You are free to choose the tools and techniques you use to perform the analysis. Support has been provided for both R and Python for interfacing with the Game platform. Note: your model can be created independently of the player, provided you can save the output of your model into a readable format. You will be required to prepare a self-contained group report of up to 50 pages, detailing the steps taken in designing, building, and refining your model. Your player/model must be submitted via GitLab in order to be run on the Game platform.

Overview of the Game

The game will be played on a grid overlaid onto the New York area:

Figure 1: Game Board

Each cell (square) on the board represents an area players can occupy. Each cell has an id attribute and a neighbours attribute. The id attribute is of the form column:row, for example, 15:20 is the cell in the 15th column of the 20th row. Cells are labelled from top left to bottom right, based on a full grid, as shown in Figure 2. The reduced board has been constructed by removing cells that received no pick-ups during 2015, and cells that do not intersect with a road, whilst still maintaining connectivity to all cells. The full grid contains 87 columns and 141 rows, labelled from "0:0" in the top left, to "86:140" in the bottom right. However, as can be seen from Figure 1, a significant number of the cells have been removed to simplify the board.

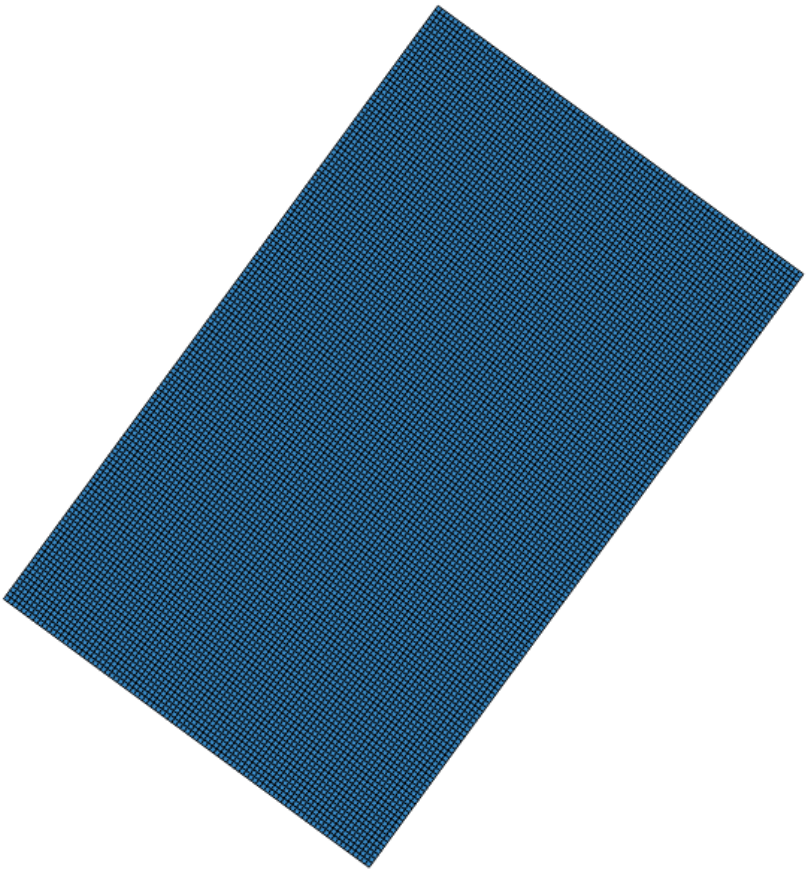

Figure 2: Full Board

The game will use real world taxi data, including both Green and Yellow Taxis. The player is considered to be a Yellow Taxi and is therefore free to operate across the entire board.

Each game will run over a simulated period of 1 week, with each round corresponding to 1 minute in time on the game clock. Each player can undertake 6 shifts of up to 12 hours each, the start day and time of each shift can be decided by the player, however, there must be at least 8 hours of rest time between each shift. The total time permitted to be spent transporting passengers in any one shift is 10 hours. Time between pick-ups does not count towards this 10 hour period. At the start of a shift the player decides where they want to start (i.e. which cell they wish to start in). For each

subsequent round of the game (1 minute interval) a player must decide the following properties:

- Availability: *For Hire* or *Unavailable*
- Default Action: *Stay* or *Move*

Availability

Each round a player can decide whether they wish to be *For Hire* or *Unavailable* for that round.

For Hire If a player chooses to be *For Hire* they will be eligible to pick-up a trip that started in that cell at that time in the data set. They are not guaranteed a trip, since only trips that appeared in the data set will be available, and priority for who is awarded a trip is based on the order in which the players arrived in the cell. If no trips are available, or all trips have been taken by other players, the player will perform the default action. A player is not permitted to reject a trip - in the same way a New York Taxi driver is not permitted to refuse a fare if they are *For Hire*.

Unavailable If a player does not want to be eligible for pick-ups they can set themselves to be *Unavailable*. This will cause the default action to be performed. A scenario where this could be used is when in an area that contains a high number of unprofitable trips and where it is deemed more advantageous to move to another area before looking for further trips to pick-up.

Taken This is an automatically set state, which occurs when a player is awarded a trip. When a player has been awarded a trip they will not receive any further turns until the end time of the trip, at which point they will reappear on the board in the drop-off location, and be able to choose their next action.

Default Action

This is the action that will be taken if a player either does not get a trip, or has chosen to be unavailable.

Stay If a player decides to stay they will remain in the cell they are currently in for the next round of the game. If a player stays in a cell, they will have higher priority than any other player moving into that cell during the current or future rounds.

Move A player can choose to move to an adjoining cell should they either by *Unavailable*, or *For Hire* and not awarded a trip. A list of neighbouring cells will be provided to the player during each turn to facilitate selection of a valid neighbour.

At the end of the shift, after either 12 hours working time or 10 hours of transporting passengers, any trips awarded prior to the end of shift time will be completed even if they finish after the allowed time. After the final trip, or default action, the player must choose when and where to start their next shift, which must be at least 8 hours after the end of their current shift. During this period the player is effectively dormant, off the board, and will reappear on the board at their chosen time and location to resume their next shift. Each player can work a maximum of 6 shifts during the week. Any shifts not completed by the end of the game are lost.

Game Platform

Each group will be assigned a GitLab repository in which they should store their work. Once setup, each night an automatic process will pull any updates made to the master branch of your GitLab repository and play the game with your, and any other groups, player. The results will be published on the Game Result repository `https://gitlab.eng.unimelb.edu.au/mast30034-2019/nycgameoutput` to allow you to evaluate and refine your model.

Each player has 5 seconds to play their turn. If they fail to respond during that period, or the player code crashes, throws an exception, or returns an invalid response, the player will be excluded from the rest of the run of that game. Players will be started at the beginning of the game and are expected to run until the end of the game. Communication will be performed via StdIn and StdOut. As such, a player may maintain state (within available memory resources) between turns.

Log files for your player will be provided via a shared CloudStor folder that will be provided to each group following the setup of the GitLab repository.

A local version of the Game Platform will provided in due course to allow you to run the game against your player locally. Instructions on downloading and running the Game Platform will be made available from `https://gitlab.eng.unimelb.edu.au/MAST30034/GamePlatform` when it is released. Any updates to the platform will be pushed to that repository and groups notified.

Winning the Game

Players will be ranked on total earnings - fare + tips. Taxes, tolls, and surcharges will not be included in earnings. Note: this is limited by the data set, since it only provides tip information for credit card transactions. Total earnings will be taken to be the mean across the multiple runs of the game. As such, if the game is run 10 times, it will be the mean total earnings across all 10 runs that will be the final value that is evaluated. The order in which players start is randomised, so that over multiple runs each player should get the opportunity to go first. Additionally, the trips that are available in any given cell during a round are also randomised. As such, being the only player in the same cell, at the same time, during different runs of the game, may not result in the same trip being awarded.

The final game will be run using an extract from an undisclosed week.

GitLab Usage

Please do not push large data files to GitLab, you should exchange/store the data files outside of your repository. The model should be small enough to be able to be loaded into memory within reasonable resource limits. You should not be building the model from scratch during the game.

Provided Resources and Restrictions

A number of resources will be provided for you to allow you to focus on the development of your model. These include the following:

- Labelled data for July 2015 through to June 2017 - each row will have the pick-up and drop-off cell as additional columns - where the trip falls outside the grid the value will be blank. This data has not been cleaned, only labelled where appropriate. You must only use this period

of Taxi Data to train your model. The test period will fall outside of this period.

- You are free to use additional data if you wish, for example, weather data. You will be provided with the date and time each round and can therefore look-up any additional data - but recall you only have a limited amount of time to complete your go.
- The final game will be played on one or more weeks within a 6 month period of the training data, i.e. Jan 2015-June 2015 or July 2017-December 2017. If multiple weeks are used the average result of all games will be taken as the final result.
- Sample game data for 1 week - a sample of game data will be provided to allow local games to be run
- Local Game Platform - allows games to be run on your local machine `https://gitlab.eng.unimelb.edu.au/MAST30034/GamePlatform`
- Board Shapefile - a geoJSON file containing the board as features will be provided. Each feature will include the cell id (co-ordinate column:row), as well as list of cell ids for the neighbours of the cell
- Sample player files for R and Python 2 & 3. These will include skeleton code for handling communication between the player and the Game Platform, and a sample Random Walker implementation of a player
- Each group will be allocated a group GitLab repository, in which source code, documents, and issues should logged

There will be a staggered release of the resources as we want to you to spend time planning what it is you are going to do before starting to build your models.

Assessment

The assessment of the project accounts for 50% of your overall mark. That 50% will be divided as follows:

- Group Presentation 12% - based on content, plan, and presentation delivery

- Final Report 30% - a final report of up to 50 pages detailing the design, building, and refinement of the model
- Up to 8% will be awarded based on the final leader board

Final Leader Board Allocation

The 8% will be distributed as follows:

- 1st place 8%
- 2nd place 7%
- 3rd place 6%
- 4th down will start at 5% and decrement by 0.5% for each place until it reaches 0.5% or no more teams remain. For example, if you come 5th you will received 4.5%, 7th will receive 3.5%. If you come 13th or below you will receive 0.5%

Group Presentation

Your group presentation should be the culmination of the formulation part of the group project. It should contain a plan for how you intend to approach the problem, and build your model. It should be no longer than 10 minutes in length, with group members prepared to answer up to 5 minutes of questions at the end of their presentation. Presentations will take place during Week 9, exact times will be arranged with groups in advance. It should include:

- High-level problem description
- High-level description of approach to solve the problem
- Planned techniques to be developed to build your model
- Broad allocation of tasks to group members

All members of the group are required to take part in the presentation, with the time available split evenly between group members.

You will be required to upload your presentation slides to your GitLab repository after you have delivered your presentation. Your implementation

should be consistent with the plan presented during your group presentation. Significant changes to the plan must be discussed with the Subject Coordinator.

In addition, you will be required to submit a Self Reflection report (worth a further 10%), which is due during the exam period. Further details will be provided closer to the time about the required contents of that report. Your individual final mark will take into consideration the self reflection report, as well as a confidential peer assessment that will be conducted at the end of the project. In addition, we will analyse the usage of the team working tools used, Slack, GitLab, and the documented evidence of team work to determine the final mark for each individual in the group.

Group Work

A key component of the group project is the application of group working skills, which will be included in your assessment. As such, we expect a number of group work activities to take place on a regular basis, and for there to be documented evidence of how well you are working as a team. We expect the following to take place:

- Regular communication via either your dedicate group Slack channel or GitLab issues page
- Weekly meetings (ideally face-to-face) - you can use workshop time for this if convenient, or arrange at other times during the week
- Weekly meetings should be minuted, with the minute taking responsibilities rotating between group members. Each member of the group must take the minutes on at least one occasion.
  - Minutes should be uploaded to your group GitLab repository within a few days of each meeting
  - Minuted actions must additionally be recorded as Issues within GitLab, with appropriate due dates and assignees
- GitLab issues are expected to be correctly managed, updated, and closed to provide evidence of contribution and team work

Teaching staff will monitor your GitLab repository, Slack channel, and may ask to sit in on your weekly meetings. In the event of disagreement

within a group, or concern about contribution or participation, it is important to notify the Subject Coordinator as soon as possible, providing an explanation and evidence of the concern.

It is important to use the team working tools in order to provide evidence of contribution and team working. Failure to use the tools could adversely affect your final mark.

Report

Your group report should be a maximum of 50 pages and cover at least the following items:

- Problem identification - what it is that your are looking to optimise, with a high-level overview of why you believe it will deliver a good result, as well as describing any additional data sources or material you have used in order to define the problem.

- Planned approach to building your model - what type of model are you going to build, will it be a single model or multiple models for different areas/times etc. Are you using a statistical model, or a machine learning, or something else. Are you going to develop several models and then evaluate which performs best? This section should describe the planned steps, stages, and processes you plan to use.

- Detailed description of the techniques underpinning your model - describe the techniques that are underpinning your model, whether it be statistical models or machine learning techniques. Why have you selected those techniques, and what about them makes them suitable for the problem you have identified.

- Refinement and Evaluation of your model - evidence of the steps you took to refine and evaluate your model. This could include test results or statistical analysis and verification of your model. If you have dropped one approach in favour of another, describe why in this section, and provide some evidence for that decision.

- Limitations and suggested future improvements of the model - what you have done differently, are there particular weaknesses that you have identified in your model, for example, does it perform well in

some areas of New York but not others? Suggest how the model could be improved further or what could be done differently to improve it.

Submission details

Your model code and player implementation must be committed to your group GitLab repository by the deadline. The last commit prior to the deadline will be taken as your final submission. Your report must be submitted via Turnitin on the LMS.

- Late submissions will incur a deduction of 2 marks per day (or part thereof).
- If you submit late, you MUST email the subject coordinator, Chris Culnane, cculnane@unimelb.edu.au.

Extension policy: If you believe you have a valid reason to require an extension you must contact the subject coordinator, Chris Culnane cculnane@unimelb.edu.au at the earliest opportunity, which in most instances should be well before the submission deadline.

Requests for extensions are not automatic and are considered on a case by case basis. You will be required to supply supporting evidence such as a medical certificate. In addition, your git log file should illustrate the progress made on the project up to the date of your request.

Plagiarism policy: You are reminded that all submitted group project work in this subject is to be the work of members of your group. Automated similarity checking software will be used to compare submissions against each other and known public source code. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student concerned.

Further Hints

- Do not train you model to specific dates, you should exclude the test period from your training data to ensure you build a reliable model that will perform well in the final game
- Clean the data to ensure you are only building your model on trips that will appear in the game

- Consider pre-processing your data before training/building your model, e.g. don't train on the raw data, train on binned aggregates - take advantage of the inherent binning in the game to reduce the quantity of data you need to process. Pre-process once, store the output, and then build your model using that aggregate data to reduce overall processing time.
- Consider building multiple models based on different weekdays or time periods (morning/afternoon/evening).
- If you find the problem space is too big, consider dividing the area up and building separate models for different geographic areas.
- Use the combined experience of your group to gain insight into trends, discuss your previous analysis from Assignment 1 and 2 with each other to increase collective awareness of features in the data set.
- Try to plan a staged approach, build something simple, and then gradually refine it with more intricate models based on more fine grained data.
- Consider how you will route your taxi back to desirable locations, don't always assume the best option is to be for hire, and consider efficient way of navigating the board.
- You might consider creating an animated visualisation of your log files to get a better idea of what your player is doing during the game.