ads_course_assessment

December 10, 2024

1 Assessment for Advanced Data Science

2 *Data Pipelines*

2.1 Radzim Sendyka, Christian Cabrera, Carl Henrik Ek and Neil D. Lawrence

2.1.1 19 November 2024

Welcome to the course assessment for the Advanced Data Science unit. In this assessment, you will build two data pipelines and prediction systems.

The maximum total mark for this assessment is 20. That mark is split into two miniprojects below. - The first miniproject is worth 7 marks. - The second miniproject is worth 13 marks.

The check Session for this Assessment is 03 December 2024.

3 Task 1 (7 marks)

This example project builds on your experience from the practicals, using Open Street Maps, and connecting them to real-world datasets.

Your prediction system will be based on data from the 2021 UK Census, and Open Street Maps.

3.0.1 Build a data pipeline and prediction system.

Structure your system around the Access-Assess-Address framework for automated data science. Use the framework of Data Readiness Levels to track the state of your datasets throughout the process. Visualise your data and predictions using appropriate charts and graphics. Remember, your notebook should tell a story.

1.1. Download the data from online sources. In this task you will need: - UK Census Data - The UK Census data file for and Socio-economic Classification (NS SEC) here or here. Use the most detailed area type (Output Areas). - One other UK Census data file of your choosing here or here. Use the most detailed area type available. - Geographic coordinates of Census Output Areas here. - Spatial data - Set up an API connection to Open Street Maps here. Optionally, to be able to use more data in training, you might want to also download the entire map for England (or more) here or here.

1.2. Structure the data, and upload it to your AWS database.

- Census datasets can be huge and unwieldy to manipulate directly in python frameworks such as pandas. As a result we will host the data in a relational database.

- Using a cloud hosted database (such as MariaDB hosted on the AWS RDS service), you will construct a database containing tables that contain per-output-area data, latitudes and longitudes of the output areas, and potentially, extracted selected open street maps features from the country-wide data file.

- Make sure you set the correct indices on your data, which will allow your code to run much faster. Hint: create an index on any columns you will be looking up, and joint indices if your lookup involves multiple columns (like coordinates).

You will likely find the following resources helpful.

- Lecture 1, 2 and 3.

- Lab class 1 and 2.

- *Note, this particular dataset could probably be handled directly in pandas, but you are explicitly tasked with using a database. This is a compromise from previous iterations of this course, where students spent quite a lot of time struggling from the size of the data. This is an opportunity for you to demonstrate your skills, as well as prepare for the second part of the assessment where databases might be necessary. Approach that do not use databases where needed will be penalised during marking.*

2.0. Explore the data.

- To make predictions you will augment your data with information obtained from Open Street Map: an open license source of mapping information. You will need to use the techniques you have learnt in the course to indentify and incorporate useful features for your prediction tasks.
- Look at the percentage of students (NS SEC code L15) in the individual output areas, and explore how it's related to various types of open street map features in those locations. Use the relationships you found to suggest the features to be used in a system that predicts percentage of students in a given area.
- Do the same for an additional census variable (e.g., the average age in a given area).

3.0. Make a prediction system. - Use Open Street Maps to predict Socio-Economic Status - in particular, the percent of full time students, from OSM features. Your function should take as input a coordinate pair (you can assume it will be in England), and return an estimate for the share of students in that area.

- Do the same for the additional census variable you selected in the previous step.

Hints:

- *Some of the census data sheets are very detailed. You might want to try "boiling them down" to some aggregate value or selecting just the most important columns.*

- *This assignment is focused on data pipelines, rather than machine learning, so we do not expect any advanced architectures here - a linear model or similar is perfectly fine.*

- *For data exploration, or picking your second dataset, you may find useful the interactive map of the census data.*

3.0.2 Story.

- Remember the notebook you create should tell a story, any code that is not critical to that story can safely be placed into the associated analysis library and imported for use (structured as given in the Fynesse template). Make sure your decisions are well substanciated and flow nicely throughout the notebook, with clear explanations.
- Explain everything you do. Why is your question important, why you selected the datasets you did, why did you join them how you did, why did you look at the things you looked at.

Make sure in your notebook you discuss the design decisions you made in your pipeline and prediction system. How did you structure your notebook? How did you track data readiness? Which aspects of your system could be improved further?

Make sure you compare how you predict student population and your second chosen property. What differences are there, in the underlying data, process, and results?

Make sure you discuss the reusability of your data pipeline and prediction system. What changes to the underlying data/system is it resilient to, and what isn't it? What work would need to be done to adjust it to answer questions about other statistics, or a different country, instead?

3.0.3 Fynesse library.

- Alongside your implementation you will provide a short repository overview describing how you have implemented the different parts of the project and where you have placed those parts in your code repository. You will submit your code alongside a version of this notebook that will allow your examiner to understand and reconstruct the thinking behind your analysis. This notebook is structured to help you in creating that description and allow you to understand how we will allocate the marks. You should make use of the Fynesse framework (https://github.com/lawrennd/fynesse_template) for structuring your code.
- One artefact to be included in your submission is a python library structured according to the "Access, Assess, Address" standard for data science solutions. You will submit this library alongside your code. Use the cell below to perform the necessary installation instructions for your library. You should base your module on the template repository given by the Fynesse template repository. That should make it pip-installable as below. `%pip install git+https://github.com/lawrennd/fynesse_template.git`. You can either replace fynesse with the name you've given your analysis module or you can leave the name as fynesse if you prefer.
- Remember the notebook you create should tell a story, any code that is not critical to that story can safely be placed into the associated analysis library and imported for use (structured as given in the Fynesse template). Remember to include you fynesse library in your submission.

3.1 Access

#1.1

Firstly, we will download the following UK Census Data: 1. TS067: National Statistics Socio-economic Classification 2.