Project

DATA 622

Overview

In this project, you will use real COVID-19 data to train machine learning models that can predict future numbers of confirmed cases, deaths, and recovered cases. While you have up-to-date COVID-19 data from the entire world at your disposal, the objective is to make accurate predictions for the 10 Canadian provinces and Canada as a whole (for recovered cases, there aren't provincial data so only predict Canada as a whole). You are also welcome to pull in other data sources you think can help with the predictions. Keep in mind, however, that you need to include in your submission everything required to run your code.

Unlike the assignments, you are asked to tell a data-driven story in a Jupyter notebook. While making accurate predictions through machine learning is an important part of this project, you will also need to clearly and convincingly communicate which information is most predictive, any patterns you identified in the data, your feature engineering strategy, and justification for your machine learning model design. Your notebook should tell the story step-by-step using both text and code, just like any good Jupyter notebook tutorial you would find online.

Data Source

You will use the publicly available COVID-19 dataset from the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) which you can find in their GitHub repository. Navigate to "COVID-19/csse_covid_19_data/csse_covid_19_time_series/" and you will find the following data files that you will need for this project:

- time_series_covid19_confirmed_global.csv
- time_series_covid19_deaths_global.csv
- time_series_covid19_recovered_global.csv

These data files are updated daily (around midnight in UTC) so make sure you keep refreshing as you work on this project for the next several days. Everyday, a new column for that day is added to all three data files.

Note that provincial-level data are only available for confirmed cases and deaths.

You don't need to include these data files in your submission. Your code can assume that these data files, as of April 19, 2020 (project deadline), will be in the same folder as your Jupyter notebook.

Optionally, you are free to use data from other external sources if you think they can help with the prediction. Just make sure your notebook executes without any issue. Either submit the external data as part of your submission or fetch them on the fly within your code.

Prediction Problem

The objective is to predict the cumulative numbers of confirmed cases, deaths, and recovered cases for April 20-22, 2020 (i.e., the three days immediately following the project deadline) for each of the 10 Canadian provinces and Canada as a whole (for recovered cases, there aren't provincial data so only predict Canada as a whole). For confirmed cases and deaths, predictive performance will be evaluated via mean squared error averaged over 3 days x (10 provinces + Canada) = 33 predictions. For recovered cases, mean squared error will be averaged over 3 days x Canada = 3 predictions. Confirmed cases, deaths, and recovered cases are equally important as far as predictive performance evaluation is concerned.

Prediction Output

For each of confirmed cases, deaths, and recovered cases, your code should print the predictions to the screen in three tables and save them to three separate csv files (file names: confirmed_predicted.csv, deaths_predicted.csv, recovered_predicted.csv) in the following format (the table for recovered cases should only have one row for Canada):

	April 20	April 21	April 22
Canada	...	...	...
Alberta	...	...	...
British Columbia	...	...	...
Manitoba	...	...	...
New Brunswick	...	...	...
New Foundland and Labrador	...	...	...
Nova Scotia	...	...	...
Ontario	...	...	...
Prince Edward Island	...	...	...
Quebec	...	...	...
Saskatchewan	...	...	...

The above table should be populated with predicted cumulative numbers of either confirmed cases, deaths, or recovered cases. Remember to generate three tables.

Feature Engineering and Machine Learning

Predictor variables and machine learning model types/architectures/hyperparameters are completely up to you. As mentioned above, you can even bring in data from other sources if you think they can help with the predictions. You will need to justify your approaches, however, either narratively or by providing empirical evidence generated by your code (or both).

Your predictive models must be primarily machine learning based, although it is okay if some small parts of your model are based on non-machine learning techniques (e.g., statistical, mathematical, heuristics).

If training your machine learning model takes too long (> 10 min), especially if you are using large datasets from external sources, you should save and submit your trained model (save it using ***pickle*** in Python, for example) and load it in your Jupyter notebook. You should still have the training code in the notebook, however, and explain your modeling choice. If you are only using the COVID-19 dataset, your model training would most likely finish within a few minutes. If this is the case, you can just do the training in the notebook and there is no need to submit a saved model.

Some Advice

- Before jumping into machine learning, make sure you explore the dataset first to understand it. Your Jupyter notebook should contain key data exploration (e.g., visualizations) to build a case for your feature engineering and machine learning approaches.
- While you can just use the past data of the same jurisdiction to predict (e.g., use past Alberta data to predict for Alberta), data from other Canadian or foreign jurisdictions could be helpful especially if other jurisdictions are ahead with similar patterns.
- You may consider utilizing the associations among confirmed cases, deaths, and recovered cases. In general, deaths and recovered cases lag confirmed cases because they usually become confirmed cases first.
- Since the COVID-19 data are sequential, sequential models are obviously well suited. However, standard supervised learning models may also work.
- Keep in mind that this is a challenging, real-world prediction problem (if this was an easy problem, we would already know when the pandemic would peak and finish). This means that your model's performance will be limited.

- Utilize the project discussion board on D2L to ask/answer questions and bounce off ideas. However, since grading includes performance ranking (see "Grading Scheme" below), you may want to keep your secret recipe to yourself.
- You may get inspiration from a similar ongoing Kaggle competition.

Grading Scheme

This project is worth **30% of your total grade** in the course, broken down as follows:

- **15%**: Clear, effective, and convincing description of the overall strategy, data exploration, feature engineering, predictive information, and modeling choices. Your data-driven story should try to address the following questions:
  - Are there any interesting patterns in the data?
  - What is the most important predictive information?
  - Are the feature engineering methods and modeling choices well justified?
  - Is the predictive model primarily machine learning based?
  - Is this even a feasible prediction problem?
- **10%**: Error-free code that generates the predictions in the correct format. If your code results in errors and does not execute to completion, or the predictions are presented in incorrect format, partial marks will be given depending on the severity of the issues.
- **5%**: Predictive performance ranking. Your model's predictive performances for confirmed cases, deaths, and recovered cases will be ranked separately against your classmates (using average mean squared error as described above under "Prediction Problem") and weighted equally for this grade component.

Deliverables and Deadline

- Submit your ***project_studentid.ipynb*** file and optionally any other files that are required to run your code (there is no limit to the number of files you can submit) to the **Project Dropbox on D2L**.
  - Submitted Jupyter notebook should be well polished. It should effectively tell a story with a good mix of code and text boxes. All code should be well commented. There should NOT be any rough work that should not be part of the final submission.
  - Replace "studentid" in the file name with your student ID
- Due at **11:55pm on Sunday, Apr. 19**.