Analytics Cup 2019/2020
Classifying Rush-Hour Taxi Trips

The Challenge

You are given data about recent taxi trips taken in the city of Chicago. Your objective is to build a model that correctly classifies whether trips in the test set happened during 'rush hour' or not.

You are given 200 000 training instances that contain (almost) exact timestamps of the trips (see below). The test set contains of 50 000 instances for which you only know the date, but not the exact time of the trip. For each of these 50 000 trips, your model must make a prediction about whether it happened during rush hour (prediction=1) or not (prediction=0).

Definition of Rush Hour

'Rush hour' in the context of this challenge is defined as follows:

A trip is a rush-hour trip if it started between 7am and 9am or between 4pm and 6pm on a working day. *A working day is defined as a weekday (Monday-Friday) that is* ***not*** *a public holiday in the city of Chicago, i.e. neither an Illinois public holiday or a United States federal public holiday. 'between' is defined as inclusive on both sides, i.e. trips started at 7:00:00am or 9:00:00am on a workday* ***both*** *count as rush hour trips.*

Evaluation

Your predictions will be evaluated based on the performance measure of **balanced accuracy** – the arithmetic mean of Sensitivity and Specificity.

	Truth (trip was during rush hour)		
Your Prediction		YES	NO
	YES	True Positive	False Positive
	NO	False Negative	True Negative
		Sensitivity = True Prositive Rate = TP/(TP+FN)	Specificity = True Negative Rate = TN/(FP+TN)
		Balanced Accuracy = BAC = (Sensitivity + Specificity) / 2	

The Data

Train and Test Dataset – Chicago Taxi Trips

The main dataset (train.csv and test.csv) contains information about recent taxi trips taken in the city of Chicago and contains the following columns:

Column	**Description**
ID	(Integer) A unique identifier for the trip.
Taxi ID	(String) A unique identifier for the taxi.
Trip Start Timestamp	(Training: Date & Time, Test: Date only) When the trip started, rounded to the nearest 15 minutes.
Trip End Timestamp	(Training: Date & Time, Test: Date only) When the trip ended, rounded to the nearest 15 minutes.
Trip Seconds	(Number) Time duration of the trip in seconds.
Trip Miles	(Number) Distance of the trip in miles.
Pickup Community Area	(Integer) The Community Area where the trip began. This column will be blank for locations outside Chicago.
Dropoff Community Area	(Integer) The Community Area where the trip ended. This column will be blank for locations outside Chicago.
Fare	(Number) The fare for the trip in USD.
Tips	(Number) The tip for the trip. Cash tips generally will not be recorded.
Tolls	(Number) The tolls for the trip.
Extras	(Number) Extra charges for the trip.
Trip Total	(Number) Total cost of the trip, the total of the previous columns.
Payment Type	(String) Type of payment for the trip.
Company	(String) The taxi company.
Pickup Centroid Latitude	(Number) Approximate coordinates of start/end of the trip. The given latitude/longitude is that of the center of the pickup/dropoff census tract [which is more fine grained than community area] or of the community area if the census tract contained too few trips (In that case the census tract location will be hidden for privacy.). This column often will be blank for locations outside Chicago.
Pickup Centroid Longitude	
Dropoff Centroid Latitude	
Dropoff Centroid Longitude	

Additional Data – Socioeconomic Data of Chicago Community Areas

You may want to use additional non-trip data for feature engineering. As a starting point, we have provided you with a dataset (ca_data.csv) of socioeconomic data about the Community Areas listed in the main dataset. You can find more information about this dataset here: https://data.cityofchicago.org/Health-Human-Services/Census-Data-Selected-socioeconomic-indicators-in-C/kn9c-c2s2

Notes:
1. Use of additional data is entirely optional. Using it for additional features might (or might not! – we honestly don't know in this case) be useful if you want to fine-tune your model to improve it as much as possible. Building a model without use of the extra data is perfectly reasonable.

2. You may be interested in using further additional datasets not given to you by us. This is allowed under strict restrictions on reproducibility. ***Please check the rules section below!*** *You may for example include a data set of relevant public holidays that you find online or additional data about Chicago community areas, etc. https://data.cityofchicago.org/ is a good place to look.*

The Rules

Submissions

A valid submission contains of a csv-file containing predictions and a script that generates these predictions from the data that you're given. Your submitted script **must be self-contained and reproducible**, more on that below. Your prediction file will be graded automatically and judged based on the performance measure of **balanced accuracy** it achieves on the test set. Your team can make **up to 10** valid submissions. Only the ***best valid submission*** from your team will be evaluated for grading.

We have provided you with a sample submission file (with entirely random predictions) which you can use to check whether the format of your generated submission is correct.

Prohibitions

The following things are strictly prohibited and will result in disqualification:

- You may **NOT** hard-code predictions for any instances in the test set. All predictions must be based on your model output.
  This applies both to individual predictions (i.e. **forbidden:** prediction[id==200001] <- 1)
  as well as to fixed rules (**forbidden:** prediction[weekday==Sunday] <- 0).
  Note: Hard-coding ***features*** *to be used in the model is allowed.*
- You may **NOT** work together with other teams. If we find that you copied work or cooperated, both teams will be disqualified.
- Needless to say – while you may use external data in general – you must **NOT** reverse engineer the private training set based on the original dataset underlying this challenge.

If you are unsure about whether something is allowed or not, please reach out to us or ask in the moodle forum! In cases of disambiguities, we reserve final judgment on whether a given submission violates the rules above!

Reproducibility

All submissions must be **reproducible**, i.e. the submitted R script must reproduce the same prediction file, even when run on a different machine at a different time. In order to ensure this, your scripts should (at least) follow the following guidelines:

- Import all packages that you use at the very top of the file.
  If you use a package via an mlr-learner, please explicitly import the library anyway.
- At the top of your script, right after the imports, set **`set.seed(2020)`** to seed R's random number generator (rerunning the script will then give you the same results in random operations). Some machine learning packages (such as h2o) manage their own random number generator that's not managed by R. If you use such packages, set the seed in the same manner.
- Do NOT change the file names of the training and test data sets. Your script should `read` the files (and write submissions) from/to its own directory.
- Do NOT modify the content of the data files provided. All data preparation should happen within the provided script.
  You may want to save intermediate results (data, models, etc) to disk and read them again. That is fine for prototyping, but not for the final script you submit.
- If you (optionally) want to use any external data, getting that data must be self-contained in the script. In your submitted script, you must not read any data from disk, except those files provided with the training data. (i.e. train.csv, test.csv, ca_data.csv). If you do use additional data, we suggest one of the three following ways to include it in your script:
  - You can read such data directly from its public URL rather than from disk.
  - You can hard-code the table containing external data. (E.g. a small data frame of public holidays). In this case please provide a comment with the Source (URL) of the data.
  - You can 'dump' any R-object to get R-code that recreates it from scratch using the built-in function `dput`. Including the resulting code may be useful if you want to use medium-size (<1MB) external data.

The following last point will not be handled as strictly but you should nevertheless adhere to it:

- Your submitted script should be a (reasonably) minimal implementation to generate your model. We don't expect you to spend any time on optimizing this, but please use good judgment to avoid unnecessary computation in evaluation.
  Example 1: *To find your perfect model, you performed a hyper-parameter search that took 3 days to run. Your submitted script should then only train your final model using the (hard-coded) final hyperparameters that you found. Don't include the search in your file.*
  In such a case, add a short comment about how you arrived at the hyperparameters (or comment out the code for the search)
  Example 2*: You trained 20 models and decided on your favorite one to create a submission at the end. Your submitted script should* ***only*** *run training of your favorite model, not all 20 models. (Delete or comment out the code for the other models in your submission.)*
- Although good solutions should be possible in <<10 min runtime on modest hardware (e.g. 5-year old laptops), some groups might have models that take longer. If your script takes a very long time to run, please include a comment at the top of your script that includes approximate runtime and info about your computer. (e.g. `#1.5 hours on dual-core laptop with 4GB RAM`).

Languages other than R

Some students have asked whether they may use other languages than R (such as python) for the Analytics Cup. This is permitted in general, but we cannot provide you with any support and you must adhere to the same standards of reproducibility found above.

If you want to use python, you must submit a single, self-contained python script. For grading of python scripts, all packages you use must be installable via conda.

If you want to use any language other than R or python, please contact us beforehand.