Machine Learning

Trip Duration Prediction

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Kaggle: New York City Taxi Trip Duration

- Task: predicts the total **ride duration** of taxi trips in New York City

- **id** - a unique identifier for each trip
- **vendor_id** - a code indicating the provider associated with the trip record
- **pickup_datetime** - date and time when the meter was engaged
- **dropoff_datetime** - date and time when the meter was disengaged
- **passenger_count** - the number of passengers in the vehicle (driver entered value)
- **pickup_longitude** - the longitude where the meter was engaged
- **pickup_latitude** - the latitude where the meter was engaged
- **dropoff_longitude** - the longitude where the meter was disengaged
- **dropoff_latitude** - the latitude where the meter was disengaged
- **store_and_fwd_flag** - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
- **trip_duration** - duration of the trip in seconds

My Changes

- Given the competition is completed, we will do a few changes
- Data to download [don't download or use kaggle data]
  - Find in projects directory in the drive
  - project-nyc-taxi-trip-duration/split [represents 1+ million of records]
    - test.csv.zip [need password]
    - train.csv
    - val.csv
  - project-nyc-taxi-trip-duration/split_sample [a small sample sample for fast checks]
    - train.csv
    - Val.csv
- Code starting point
  - project-nyc-taxi-trip-duration/**code_warmstart**.py
- Target variable
  - **np.log1p**(train.trip_duration)

Models, Hyperparameters and features

- For models
  - We will fix the model to **Ridge**
  - And fix its alpha to 1
  - Why? In practice, people just try bunch of models and find their best parameters
    - Very systematic
- 80% of tabular data ML is spent around the data itself
- Do your own job to understand the data (EDA)
- Your goal is to maximize the **r2-score** (best is 1)
- To get high performance, you probably need to invent **new** features

Requests

- Develop a **jupyter** notebook that focus on EDA (no modeling)
  - Make it looks like elegant
  - Write your logic and conclusion
  - Avoid aggressive printing that doesn’t help
- Provide a python code in pycharm to train and save a model
  - Be professional in coding and commenting
- Provide a code that can load your model and test on sample data (csv)
  - Once you are ready, I will give you the password for the test set
  - You are allowed to only test once and report your test-score
- Provide a PDF that is short but descent
  - Summarize your EDA findings and features introduced
  - Report your training and performance information

Flow

- Stage 1: Decide a time window (e.g. 30 hours)
- In this period, DON’T see any other solutions/notes
  - Imagine, you are in the company
  - You have to do your best!
- Stage 2: Explore the available work. Get inspired by their ideas
  - DON’T COPY their code snippets
  - If you used an idea, you must really understand its logic
- Finalize your best Ridge(alpha=1) model
  - You will notice people use many tree-based models (e.g. catboost, lightgbm, xgboost)
- The effort/difficulties of such projects, will build your ML tabular skills
- Your project score will be part of your **certificate** evaluation

Cognitive Distortions: Overgeneralizing

- *"I failed in a job interview, I will never get a job!"*
- *"I married a horrible man. Don't get married!"*
- It is easy to make **overgeneralizing** while doing ML projects
- It is very critical to remember that:

  every single project is only one datapoint
- You will learn some lessons in each ML project
  - Project #1: Polynomial regression did not work. Don't try again
  - Project #2: Undersampling doesn't help. Useless techniques!
- Keep trying the same ideas from a project to another
  - The more the same conclusion repeats, the more confidence about it!

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."