**Overview**

In this exercise, you get a chance to play with DoorDash’s data and use your machine learning experience to solve a typical DoorDash prediction problem. The exercise contains two parts

1. Building a machine learning model for a prediction task
2. Writing an application to make predictions using that model.

We require candidates to meet a high bar on this assignment to advance to the onsite round, so **we would encourage you to demonstrate the best of your machine learning and engineering toolkits.** We will look at more than just your performance on the test set.

**You will be evaluated on the following:**

For Part 1

1. The quality of your modeling approach- pre-processing, feature engineering, model selection, evaluation, etc. **Please make sure to show all of your work!**
2. The quality & insightfulness of your training script and the report (**see requirements below**).
3. Your algorithm’s performance on a test set where labels are withheld, evaluated on expected business impact.

For Part 2

1. Performance of the application when we run it on the test set
2. Quality and suitability of software engineering patterns used

**Problem Description**

When a consumer places an order on DoorDash, we show the expected time of delivery. It is very important for DoorDash to get this right, as it has a big impact on consumer experience. Order lateness/underprediction of delivery time leads to customer distrust, while overprediction of delivery times/order earliness leads to customers choosing not to place their order. From past experiments, we have found that underestimating delivery time is roughly twice as costly as overestimating it. In this exercise, you will build a model to predict the estimated time taken for delivery and an application that can make these predictions.

Concretely, for a given delivery you must predict the **total delivery duration seconds**, i.e., the time taken from

* *Start*: the time consumer submits the order (*created\_at*) to
* *End*: when the order will be delivered to the consumer (*actual\_delivery\_time*)

To help with this, we have provided

* **historical\_data.csv:** table of historical deliveries (your training set)
* **data\_to\_predict.json**: Json list of deliveries that you must predict on (label-free test set we will use for evaluation + what the application will predict data on)
* **data\_description.txt**: description of all columns in **historical\_data.csv** and details of **data\_to\_predict.json**

### Part 1

**Requirements**

* Build a model to predict the total delivery duration seconds (as defined above). You’ll likely find it helpful to generate additional features from the given data to improve model performance.
* Write a short report (typically 1-2 pages) on your approach, and results containing:
  + A high-level business summary explaining the key results to non-technical readers. Include some feature interpretation and evaluate model performance relative to what’s likely feasible.
  + A list of 3-5 additional features you believe would improve model performance if included in the training set. Please clearly articulate the value you believe adding these features would drive for the business.
  + Assume the model you designed would replace an existing model already in production. How would you assess your model’s performance relative to that of its predecessor before replacing the existing model?
  + Any other information you would like us to know about your modeling approach
  + **Don’t worry about walking us through your modeling process--we’ll get that from reviewing your code.**

**Deliverables**

* Your report
* Your code/notebook

### Part 2

* Write an application that accepts data from the JSON file with the label-free test set (*data\_to\_predict.json*), uses the model to make a prediction for each delivery in the dataset, and writes out predictions to a new **tab-separated file with columns - delivery\_id, predicted\_delivery\_seconds**.
* For this part, you must use a production-ready language like Python, Java, C++, Scala, Ruby, etc. If you are unsure if your language of choice is acceptable, please shoot us a note and we can clarify.

**Deliverables**

* Submit the output tsv file that gives the prediction for the **data\_to\_predict.json** data.
* Your code for the application. This application must be runnable from the command line with data\_to\_predict.json passed as input. Please include detailed instructions for running the code (including dependencies and installations required)

**Notes**

We expect the exercise to take 5-6 hours in total, but feel free to spend as much time as you would like. Feel free to use any opensource packages for the task.

**Thank you for your hard work! We hope you enjoy working on this fun problem. Please let us know if you have any questions. Good luck!**