# **Overview**

In this exercise, you will use your machine learning experience to solve a straightforward but challenging 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.

In Part 1, we would love to have you exhibit your modeling skills. You will be evaluated on the following - performance on the test set, feature engineering choices including features used and encoding of features, data processing, choice of models used, description of model performance and insights and observations from the​ ​model.

Part 2 is your chance to show off your software engineering skills. This includes performance of the application, adherence to common software engineering patterns (unit tests, modular code, etc.) and ability to make educated​ ​trade-offs​ ​based​ ​on​ ​the​ ​given ​ constraints.​

**NOTE:** 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.

# **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. In this exercise, you will build a model to predict the estimated time taken for a delivery and write 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
* **data\_to\_predict.json**​: ​ ​Json​ ​list​ ​of​ ​deliveries​ ​that​ ​you​ ​must​ ​predict​ ​on​ ​(for ​​the​ ​second​ ​part)
* **data\_description.txt**:​ ​ description​ ​ of​ ​ all​ ​ columns​ ​ in​​ ​**historical\_data.csv**​ ​and​ ​details​ of​ **data\_to\_predict.json**

**Requirements**

# Part ​ 1​

* Build a model to predict the total delivery duration seconds (as defined above). Feel free to generate

additional​ ​features​ ​from​ ​the ​ given​ ​ data​ ​ to​ ​ improve​ ​ model​ ​ performance.​

* Explain a) model(s) used, b) how you evaluated your model performance on the historical data, c) any data processing you performed on the data, d) feature engineering choices you made and e) other information​ ​you​ ​would​ ​like​ ​us​ ​to​ ​know​ ​about​ ​your​ ​modeling​ ​approach.
* Based ​ on​ ​ the​ ​ ​findings ​​from​ ​the​ ​model,​ list​​ ​recommendations​ to​​ ​reduce​ ​delivery​ ​time

## Deliverables

* Submit one document that includes a write-up explaining your model, choices made and discussion on the​ ​questions ​ above.​
* Submit​ ​the​ ​code​ ​used​ ​for​ ​this​ ​part

# Part ​ 2​

* Write an application that accepts data from the json file (data\_to\_predict.json), uses the model to make a prediction for each delivery in the json file and writes out predictions to a new ​**tab separated file with**​​**columns**​​**-**​​**delivery\_id,** ​ **predicted\_delivery\_seconds**​ .​
* Your predictions on this test data set will be evaluated using RMSE (Root Mean Squared Error) and your ​ score​ ​ must​ ​ exceed​ ​ a​ ​ baseline​ ​ set​ ​ for​ ​ the​ ​ ​task.

**NOTE:** 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.​
* Submit your application code. This application (that makes predictions) must be runnable from the command line with data\_to\_predict.json passed as input. Include instructions for running the code (dependencies,​ ​packages​ ​required,​ ​etc.)

## Notes

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

**Thank**​​**you**​​**for**​​**your**​​**hard**​​**work!**​​**Please**​​**let**​​**us**​​**know**​​**if**​​**you**​​**have**​​**any**​​**questions.**​​**Good**​​**luck!**