

# CS155 Final Kaggle Competition Codebook

The goal of this challenge is to attempt to predict the wait time based off of the predictors in this dataset.

The training dataset is sorted by the EnterQueueTimeStamp column which contain timestamps (the month and day values have all been set to 1 to preserve data ambiguity). It contains different queues which in this case we can abstract to be different physical locations or different types of queue flows. You may create a different model for each queue or a single model for all.

The testing dataset is in a random order.

Response: The response variable is 'actual.wait div 60000', which is how long (in minutes) a consumer has waited from the time they enter the queue until the time they have a (ticket issued/are summoned). This is what you are trying to predict, both initially at the moment the entity joins the queue, as well as in real time after that (the forecast may change).

The features are described below.

## 1 EnterQueueTimeStamp

This is the timestamp when the consumer entered the queue. It is the start of their wait experience. This TimeStamp plus the actual wait time delta gives the time they were summoned.

### 1.1 EnterQueueTimeStamp\_year

This is the year (all are 2018 in this dataset) of the datapoint.

### 1.2 EnterQueueTimeStamp\_month

This is the month (all are reset to 1 in this dataset) of the datapoint.

### 1.3 EnterQueueTimeStamp\_day

This is the day of the month (all are reset to 1 in this dataset) of the datapoint.

### 1.4 EnterQueueTimeStamp\_day\_of\_week

This is the day of the week (an integer from 0 to 6 where Monday is 0 and Sunday is 6) of the datapoint.

### 1.5 EnterQueueTimeStamp\_hour

This is the hour of the datapoint.

### 1.6 EnterQueueTimeStamp\_minute

This is the minute of the datapoint.

## **2 party\_size**

This is the number of people present as a single entity in the queue.

## **3 queue**

This is the id of the relevant queue. We have one-hot encoded this so that there are 5 features: queue\_62868, queue\_69991, queue\_66421, queue\_75252, and queue\_5000000899, each of which has a 1 if the queue is the queue whose id is the number following 'queue\_' and a 0 otherwise.

## **4 time\_since\_last\_ticket**

This is the time (in seconds) since the last ticket was issued.

## **5 pushed\_back\_by\_consumer/host/system**

These features are the amount of time (in minutes) the consumer was pushed back in the line, called upon by either consumer, host, or system.

## **6 entering\_host**

This is populated by the id of the staff member that helped the consumer enter the line.

## **7 result**

This is the outcome of their experience in the Queue. We have one-hot encoded the possibilities.

### **result\_arrived**

Means that they waited in line until they were summoned and then eventually started a service transaction at the location.

### **result\_no\_show**

Means that they waited in line until they were summoned, but they did not show up and start a service transaction

### **result\_removed**

Means that they were removed from the line at some point during their wait by a staff member

### **result\_left**

Means that they left the line before they were summoned.

### **result\_removed\_by\_system**

Means that they were removed from the line by the system itself.

## **8    `time_to_redeem`**

This is the amount of time (in seconds) it took them to redeem/start service transaction after they were summoned by a staff member.

## **9    `time_to_confirm`**

This is the amount of time it took them to confirm the ticket issued, which is how long it took the consumer to respond that they were on their way after they were summoned.

## **10   `pulled_status_updates`**

This is the amount of status updates they asked for while waiting in line. Status updates are updates consumers can ask for to see how much time they have until they are summoned.

## **11   `summoning_host`**

This is the id of the staff member that summoned the consumer whom entered the queue. Different hosts may have different average transaction durations and different idle times between transactions.