Bold Data Analyst Assignment

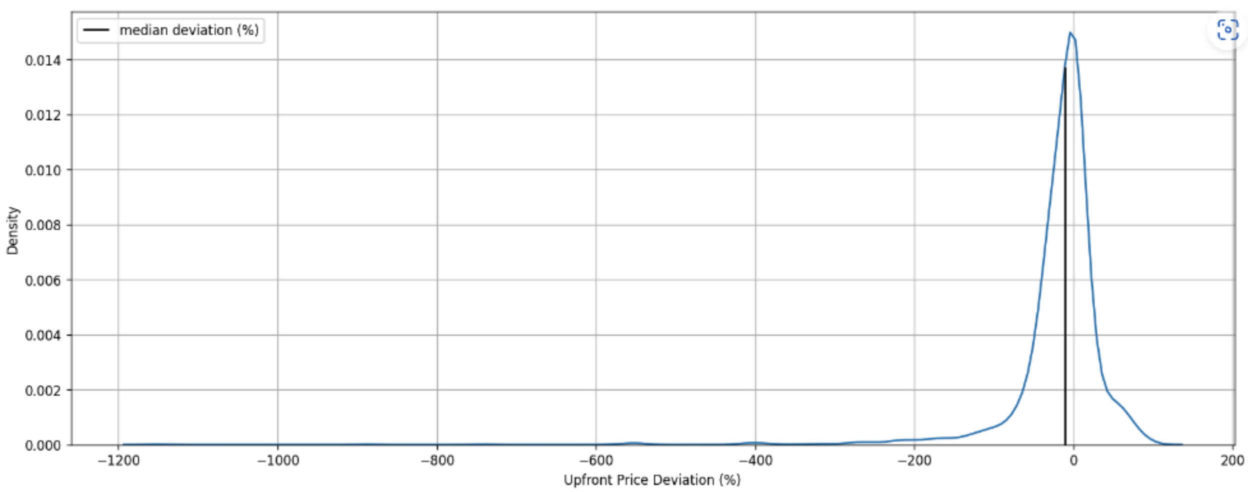
Introduction

With an effort to provide customers with better transparency of prices. Riding-hailing apps such as Bolt have resorted to providing upfront pricing to customers prior to them booking rides. However, over a few months in early 2020, Bolt's pricing algorithm tended towards predicting lower upfront prices, which leads to customers paying a higher price at the end of the ride – this ruins the customer experience. This assignment helps identify those cases and attempts to uncover their root causes.

Impact

What does the deviation look like?

After plotting the deviations between the *upfront price* and the *metered price*, i.e. (upfront_price – metered_price)/upfront_price, we get a distribution as such:



From the above, it's clear that most cases experience higher metered prices as the distribution is left-skewed and long-tailed. Furthermore, this isn't an experience you want as a customer – planning your journey only having to pay over 20% more.

How many customers does this deviation impact?

Assuming the dataset is independent and represents the population of the 4270, **70% see upfront pricing**. The others could result from pre-booking or using other means of transportation that don't necessarily display prices at the start of the journey.

Of the 70%, around **60%** of rides fall into a higher metered price than the upfront price, and **35%** of rides get charged more at the end of the journey[1]. Of the 35%, 4% went to the extent of complaining that they overpaid at the end of the journey.

Therefore, this problem is prevalent in the Bolt ecosystem and needs to be addressed.

[1] Assuming at least a 20% increase in deviation.

Identifying the source of the deviation

During upfront pricing, a change in the following factors could cause our services to update the metered pricing:

1. Geographical factors:
    a. Updating the destination – this causes the distance to update and requires an update in the predicted price
    b. New routes – if drivers take longer routes (deviating from the predicted one), customers are charged more.
    c. Tolls – unaccounted toll gates
    d. Inaccurate GPS – the final/source destination could be captured incorrectly
2. Traffic:
    a. Wait time due to incoming traffic
    b. Slower speeds and hence higher ride durations
3. Surge:
    a. Poor weather (such as rain) causes vehicles to slow down
    b. Increase demand or change in time of day – which causes pricing to start using a surge multiplier

Where do we observe the maximum deviation in values with our dataset, and by how much?

Note: Mean/Median Abs deviation (%) = Mean/Median of abs((upfront_price – metered_price) * 100/upfront_price)

GPS confidence

gps_confidence	Mean Abs Deviation (%)	count	Median Abs Deviation (%)	Standard Deviation	Coefficient of Variation
0	95.710549	306	55.415778	130.888029	1.367540
1	23.252140	2678	14.151863	35.531035	1.528076

There is an increase in the mean and median absolute deviation (%), as the GPS confidence is poor. **Could this be due to a particular manufacturer?**

device_manufacturer	Num Devices	Num Devices with 0 GPS conf	% 0 GPS conf devices
tecno	321	110.0	34.267913
infinix	85	27.0	31.764706
itel	43	12.0	27.906977
hmd	81	12.0	14.814815
sony	35	4.0	11.428571
samsung	1162	68.0	5.851979
iphone	310	18.0	5.806452
lenovo	20	1.0	5.000000

Manufacturers such as **Tecno, Infinix, and Itel constantly provide poor GPS confidence**. We could:

- Investigate if the app is running properly on these devices, e.g., using the GPS API correctly on Android.
- Warn customers about inaccurate GPS on these devices.
- Predict a certain percentage higher (explored in the next section)

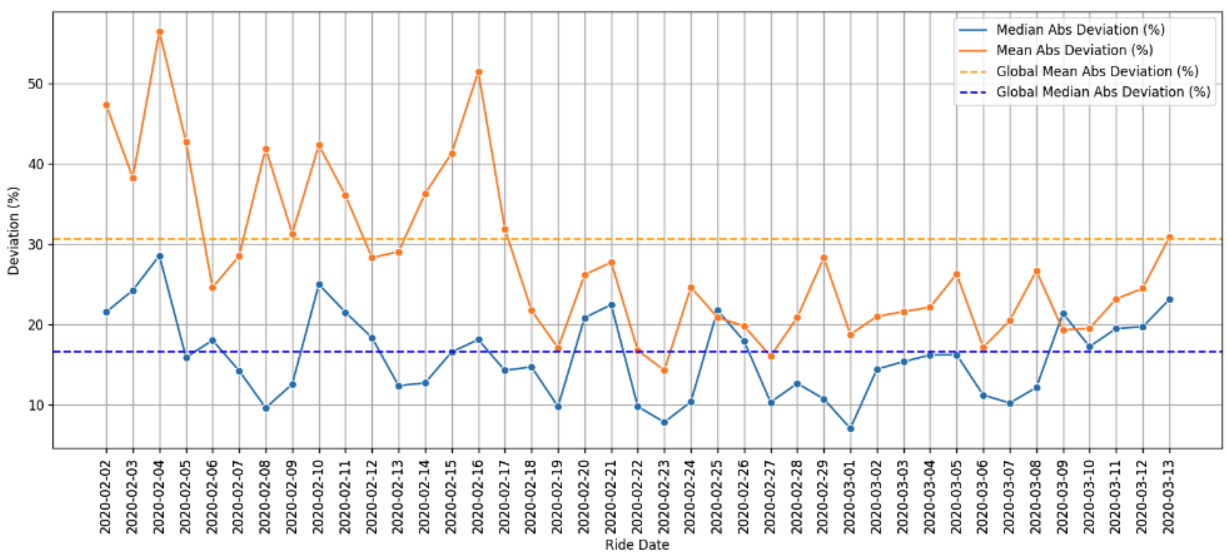
Device Manufacturer

device_manufacturer	Mean Abs Deviation (%)	count	Median Abs Deviation (%)	std
itel	58.768201	43	41.545000	77.197400
tecno	57.171041	321	28.595273	103.281983
infinix	42.257589	85	26.457895	53.987941
bullittgrouplimited	29.926596	22	24.036885	32.065987
hmd	49.270401	81	21.720667	106.658499
lenovo	53.792534	20	18.093044	95.164486
huawei	26.069487	569	15.277778	48.653991
iphone	26.878911	310	15.208034	59.003436

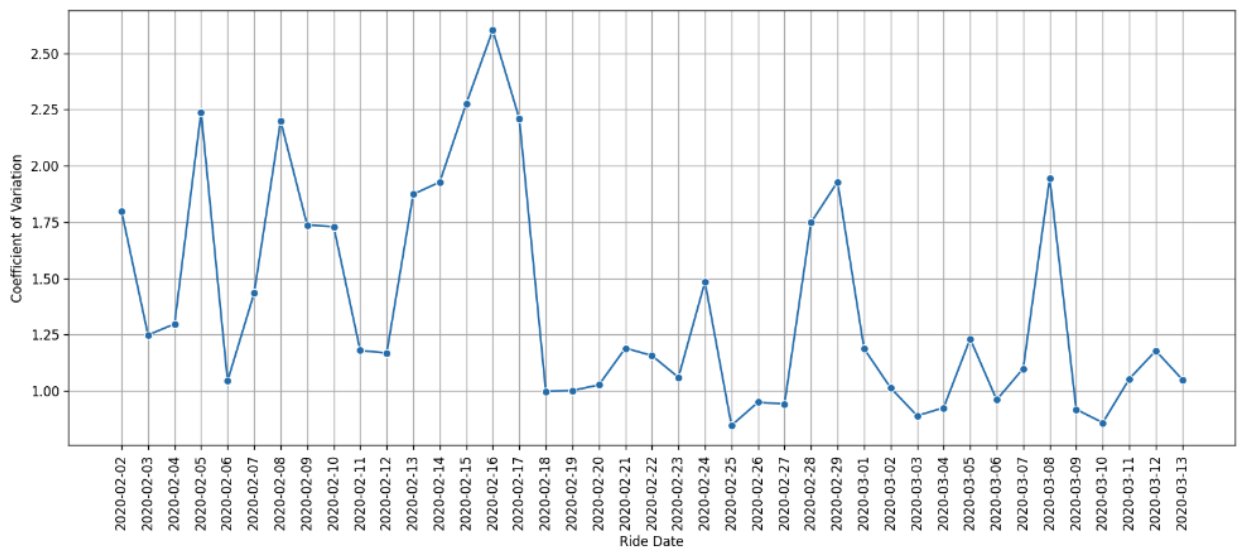
Tecno, Infinix, and Itel are repeat offenders with up to 20 pp increase in the median deviation for Itel. We can circumvent this by predicting a 5-10 pp higher upfront prices for these devices in our upfront pricing model.

Date

Is there any particular day (such as holidays) when prices deviate more?



The mean deviation peaks more than usual on the 4th and 16th of February. However, the median deviation on the 16th remains around the same upon closer inspection of the coefficient of variance (given below). The 16th was disproportionately affected by outliers. The date alone doesn't look like a factor that's impacting prices. Furthermore, there were no global holidays on these days–the data might need to be bifurcated further into countries to understand.



Destination Changes

Are more destination changes leading to higher prices?

dest_change_number	Mean Abs Deviation (%)	count	Median Abs Deviation (%)	Standard Deviation	Coefficient of Variation
1	29.670394	2873	16.024845	56.495670	1.904109
2	36.863870	49	18.873239	44.396102	1.204326
3	65.525390	52	36.763728	85.585217	1.306138
4	156.289365	6	44.790761	219.323125	1.403314
5	2.893103	2	2.893103	0.434017	0.150018
7	78.221525	2	78.221525	49.739775	0.635883

- There is an increase of 2 pp in median deviation with one destination change and an additional 18 pp with three destination changes.
- However, it's difficult to confidently say that this trend will continue as the number of ride changes increases due to the unavailability of sufficient data.
- The increase should lead us to investigate how much the driver deviated from the route after being assigned the new destination, which led to higher pricing.

EU Indicator

How non-EU countries comparing to EU countries in terms of deviation?

eu_indicator	Mean Abs Deviation (%)	count	Median Abs Deviation (%)	Standard Deviation	Coefficient of Variation
0	56.735657	706	25.733732	94.597697	1.546309
1	22.608114	2278	13.647450	36.957138	6.222734

- There is a higher price deviation in both mean and median in non-EU member countries.
- This could be because the road infrastructure is better developed in the EU. Frequent road closures and deviations in distances cause prices to increase.
- It could also be due to regulations that do not allow price deviations in the EU.

Looking at the deviation in distances below:

eu_indicator	Mean Abs Deviation (%)	count	Median Abs Deviation (%)	Standard Deviation	Coefficient of Variation
0	62.515441	706	34.035366	87.730839	1.403347
1	30.817081	2278	12.194187	140.684272	4.565139

It's evident that distances deviate more likely in non-EU countries than in EU ones.

Additional Insights

App Version

Is there a particular rider app version that's buggy?

Although pricing is calculated on the backend, is something inherently wrong with how a particular app version extracts location or duration information?

rider_app_version	Mean Abs Deviation (%)	count	Median Abs Deviation (%)	Standard Deviation	Coefficient of Variation
CA.5.47	34.609615	50	27.749529	37.221572	1.075469
CI.4.11	28.045800	15	23.750000	27.973147	0.997410
CA.5.08	35.812453	13	22.959116	39.579469	1.105187
CA.5.38	29.187596	27	22.333333	30.941768	1.060100
CI.4.04	23.642331	11	22.307692	17.244367	0.729385
CA.5.36	42.568204	95	22.156863	65.684249	1.543035
CA.5.32	37.592955	43	21.428571	82.231247	2.187411
CA.5.13	35.103528	11	21.388889	27.500050	0.783398
CA.5.04	40.726749	15	21.000000	59.132441	1.451931
CI.4.14	35.100988	96	19.532794	64.834100	1.847073
CA.5.42	38.807904	233	19.347826	69.586812	1.793109
CA.5.23	38.504122	18	18.293863	63.346614	1.645190
CA.5.26	34.086285	13	18.260870	59.616094	1.748976
CI.4.17	32.480360	536	17.236074	60.255134	1.855125

Some app versions, such as CA.5.47, typically perform poorly compared to others. However, it's also important to note that the app version is a function of adoption–this means the problem is only apparent as most customers are on this version.

We can normalize this data and see the percentage of rides on app versions with over 20% deviation.

rider_app_version	num_rides	num_rides_with_20_perc_deviation	perc_rides_with_20_perc_deviation
CA.5.13	11	8.0	72.727273
CI.4.11	15	10.0	66.666667
CA.5.47	50	30.0	60.000000
CA.5.38	27	16.0	59.259259
CI.4.04	11	6.0	54.545455
CA.5.08	13	7.0	53.846154
CA.5.04	15	8.0	53.333333
CA.5.36	95	50.0	52.631579
CA.5.32	43	22.0	51.162791
CA.5.23	18	9.0	50.000000
CA.5.42	233	114.0	48.927039
CI.4.22	27	13.0	48.148148
CI.4.17	536	253.0	47.201493
CI.4.19	390	184.0	47.179487
CA.5.27	17	8.0	47.058824

We notice a similar trend here, with versions such as CA.5.47 being repeat offenders that might require additional investigation.

Why are there 0 distances and 0 duration?

There are roughly 35 rides, or 1% of the ride population, with 0 distances throughout the ride. These could be a result of the following:

- GPS was malfunctioning, and the actual distance didn't get recorded.
- The driver took the ride off of the app.

In the second point, we also notice durations that are 0. Roughly 50% of the 0 distance cases. These drivers could have reached the pick-up point and immediately canceled the ride. In India, Uber and Ola drivers cancel the ride and take the commission[2]; customers are indifferent as they're usually charged the same amount.

[2] https://entrackr.com/2021/12/why-do-uber-and-ola-drivers-ask-users-to-cancel-rides/