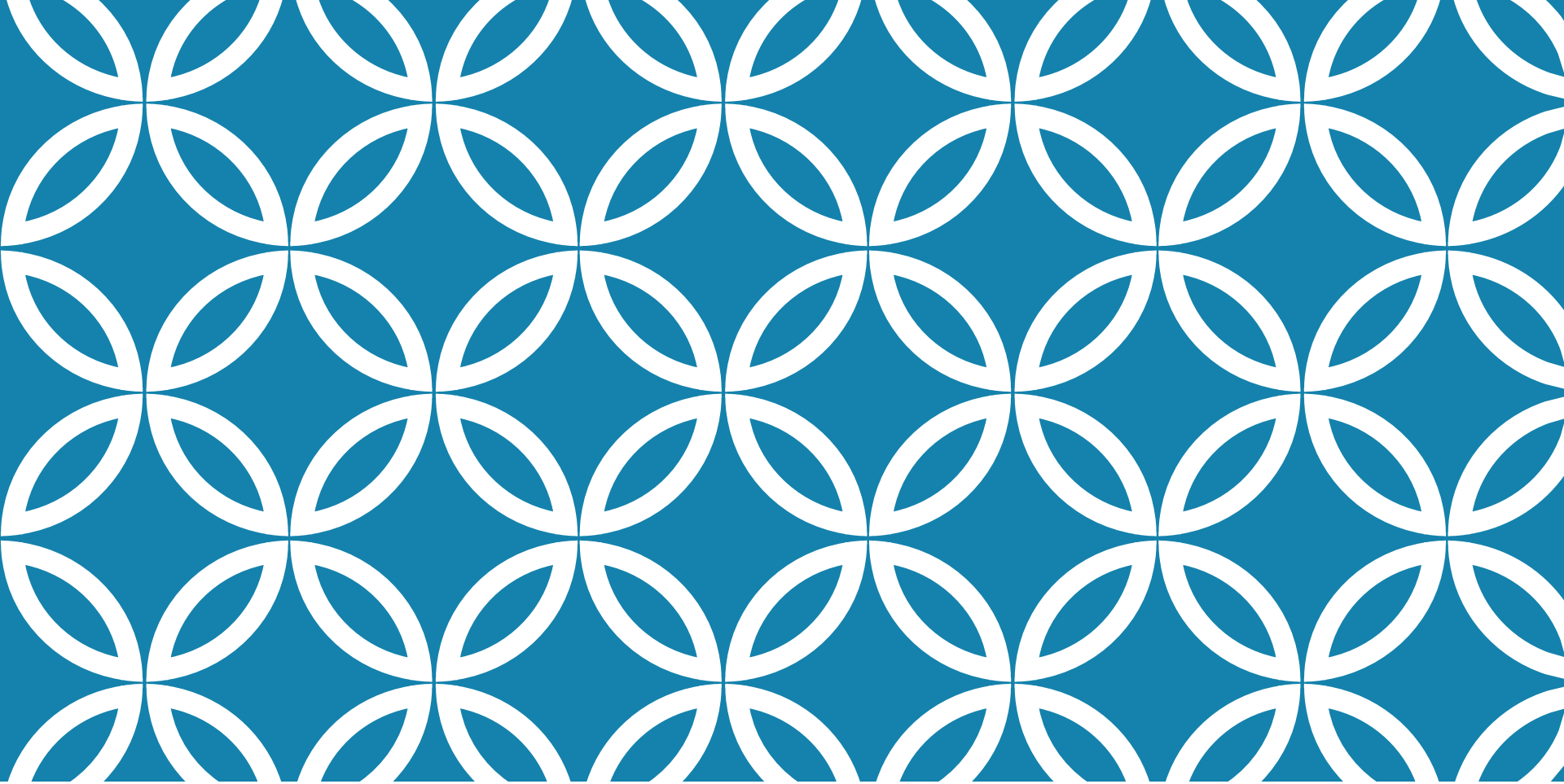


MOBILE APPLICATION

Fabio Pulvirenti
Data Scientist

FIRST USE CASE - APPLICATION NEW FEATURE

- Let's image we want to test the impact on the loading time of a new feature and we have just 24h to test it
- Study its impact on the performance of the social feed loading times (SFLT)
- Anticipate the usage of this feature to be X user / Day.
- Monitor the performance of the social feed loading times for the next 24 hours to evaluate the effect of the feature

DUMMY DATA

- No real data
  - we should generate our dummy data
- This application has surely a lot of users
  - on Facebook more than 2 Millions like
- Let's assume that half of the users who have liked it on Facebook are using it
  - Every day, half of them train with the app
  - Basis 500K people using it. In the sake of simplicity and of performances we work with 1/10 of the data, 50,000 users

DUMMY DATA

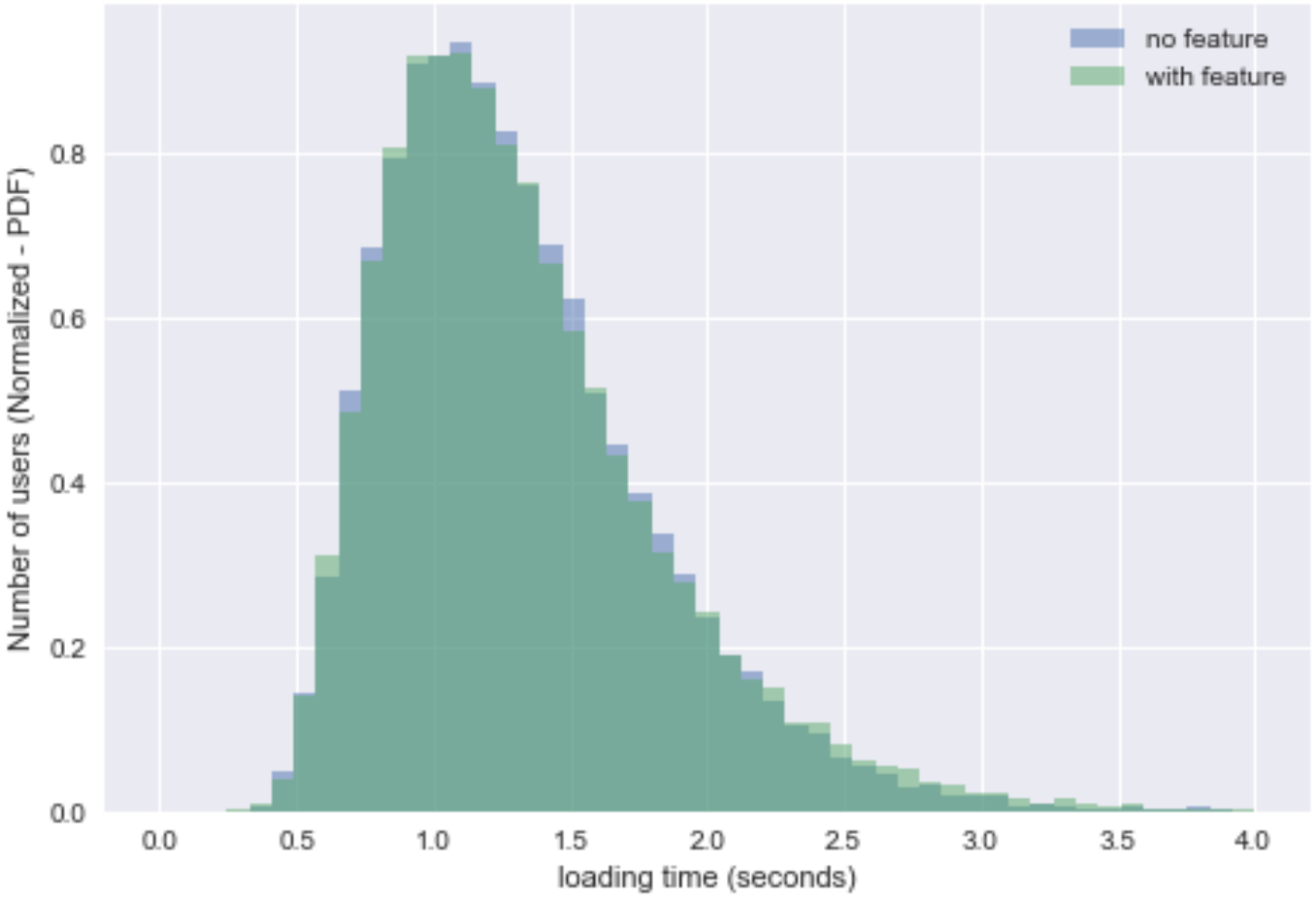
- Assume that the feature is related to a new functionality or content (i.e. not related to the algorithm used to load the social feature)
  - Its use increases loading time
- Since the test is very short (24h), we disclose the new feature for half of the users*
- In this way e are not biasing the analysis with different loading time related to the number of users using the server
  - If we consider just a small amount of users, there is the possibility that the additional loading time is not realistic because the central server is not much loaded

* Suppose the increase in the loading time is estimated in terms of seconds

LET'S START WITH SOMETHING EASY

- Imagine that we have already the loading time distribution among the users.
- We know the feature has been disclosed for half of the users
  - We have two distributions, the first related to the normal users, and the second related to the users trying the new feature
- Model it with a log-normal distribution, which is often used to model server waiting time
  - Based on some tries I have done with my own phone, the social feed is usually loaded in about 1 second
  - Let's assume a variance value up to 0.3s.
- Users with the new feature:
  - increase a bit the loading time average and the variance

SFLT DISTRIBUTIONS



- The distributions are very similar
- The distribution of the SFLT of the users with the feature is a bit more spread after the zero
- The users without the new feature are experience a more steady waiting time in the social feed

DISTRIBUTIONS

- Means
  - Reference (or without the feature) : 1.308s
  - With the feature: 1.32133
- They are very close! The distributions are almost the same
- 95% Population between
  - Reference: [0.6093 – 2.4630] (Y-Z question A)
  - Feature: [0.6066 – 2.5983]
- We have a higher mean loading time for the users with the new feature. We know that it has not happened by chance because we have created it different.
- But, in a real scenario, what could we say about it? Is it statistically significant? Which is the probability that this difference just happened by chance?

STATISTICAL HYPOTHESIS TESTING

- In this case, it would be mandatory to statistically test the hypothesis that the loading time is affected by the feature.
- 2-sample permutation-based hypothesis test (used for independent groups like ours)
- Null hypothesis: The feature does not affect the loading time
- Test statistic: mean of the loadings time of the user with the new feature
- Threshold: 95% of confidence
- P-value: is the probability of obtaining a value of our test statistic that is at least as extreme as the observed measure, given the null hypothesis is true..
- In our case: probability to obtain, after a random permutation of the two distributions, a mean value for the loading time with the feature higher than what we got
- In English ☺ : probability that what we have seen has happened by chance

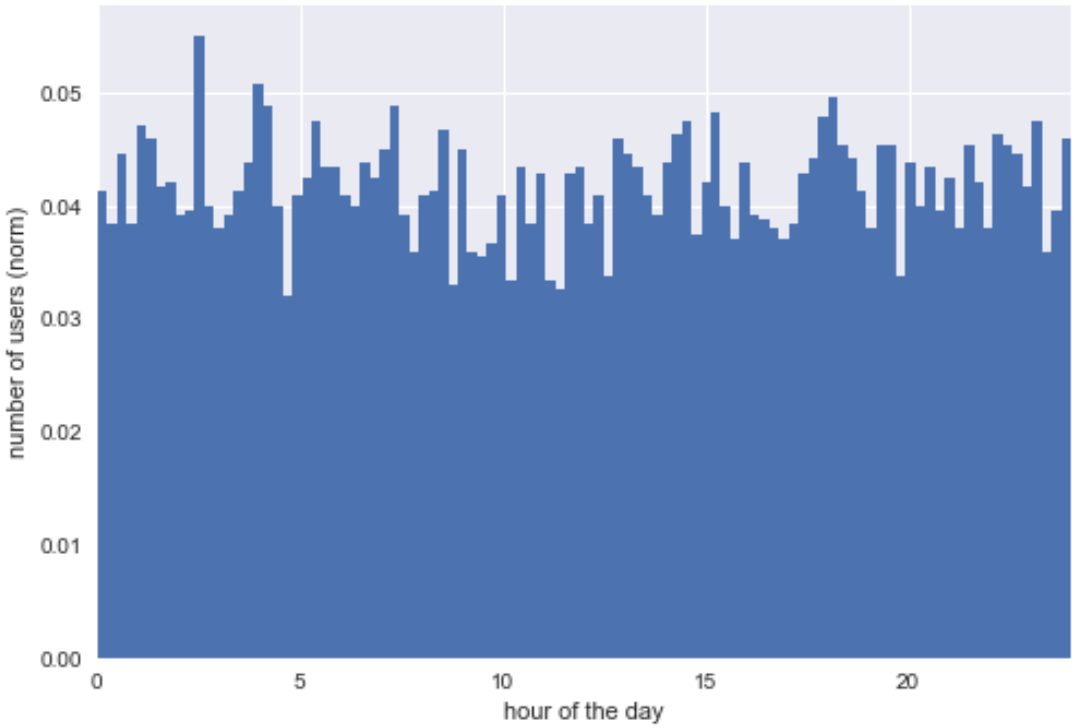
STATISTICAL HYPOTHESIS TESTING

- Create a huge number of permutation of the two distributions
- For each permutated sample of the one related to the new feature, calculate the mean value
- Count how many of them have an even higher value than what we obtained (p-value)
- If p-value is below 0.05 we reject the null hypothesis (it means that it is not covered by the 95% of the population)
- In our case, we obtain 0.0025
  - Data is statistocal significantly different than what we would observe if the null hypothesis was true
  - It is very low (as predictable), we could be very confident about our results
- We reject the null hypothesis, hence the the loading times are affected by the new feature, with a p-value of 0.25%

LET'S HAVE A BIT OF FUN

- Let's try to redo the test taking into account user traffic
- Since the application is world wide spread, we could use a uniform user distribution in the 24h

However, let's do something more interesting and focus on a single time-zone



USER TRAFFIC

Basing on my experience, there are usually 3 moments for doing physical activity:

- Early in the morning
- Lunch break
- After work

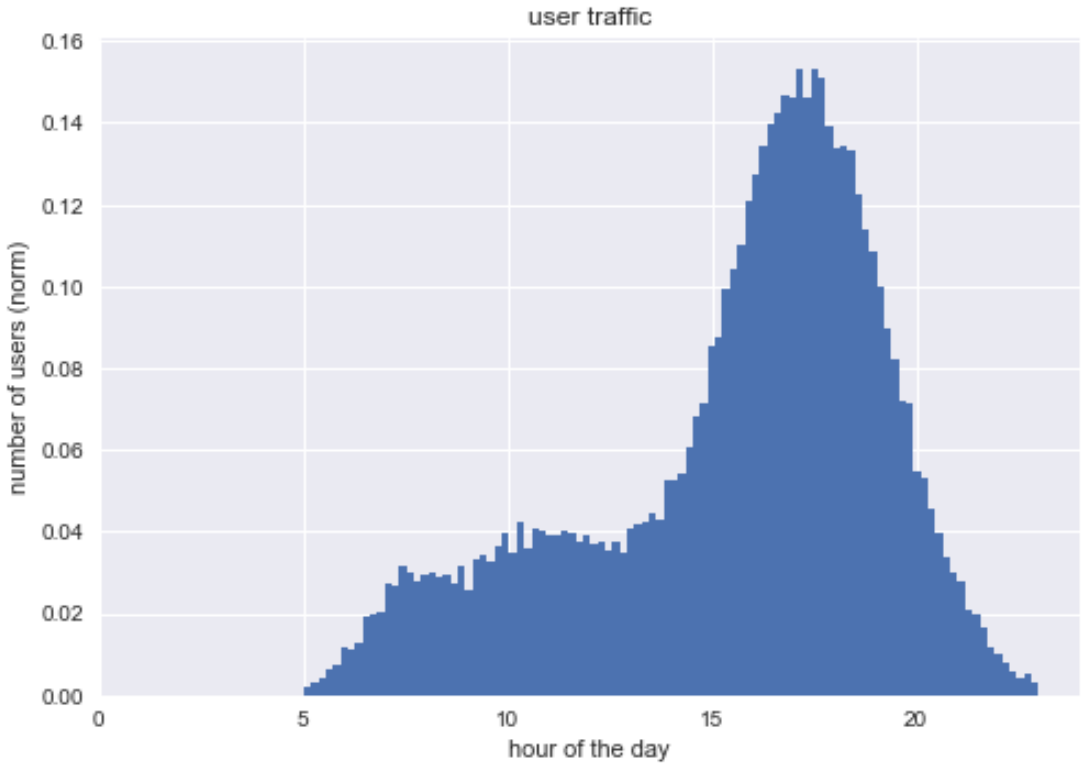
Most of the people play sport after work

(Not many play it early in the morning, especially if we are not considering jogging

USER TRAFFIC

Model the traffic as composed of these 3 main components (for each one we assume a Normal distribution)

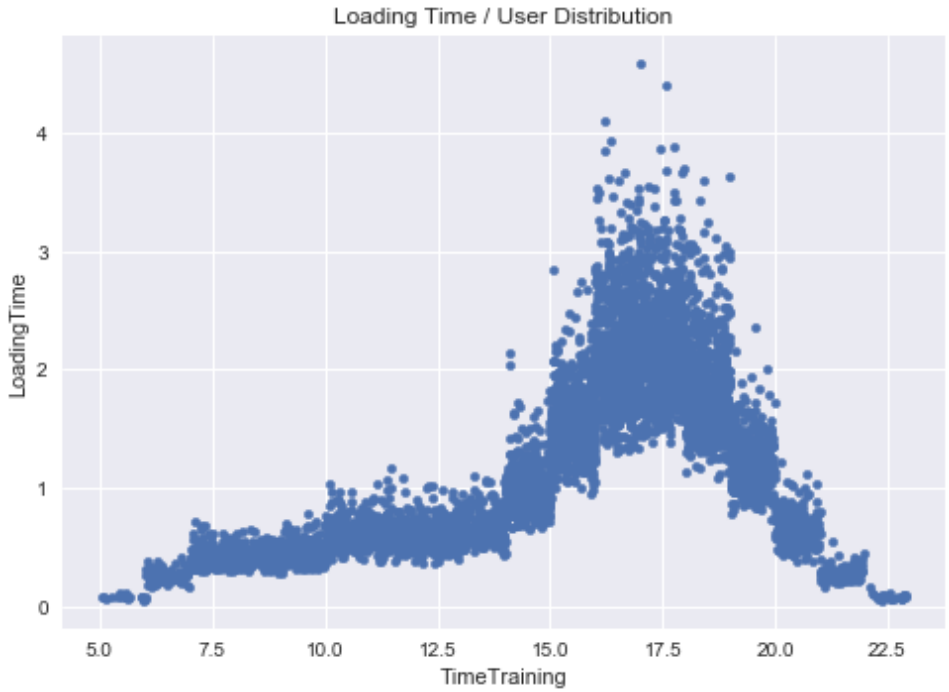
We can still regognize the 3 normal distributions centered in the early morning, at 11.00 and at 17.30

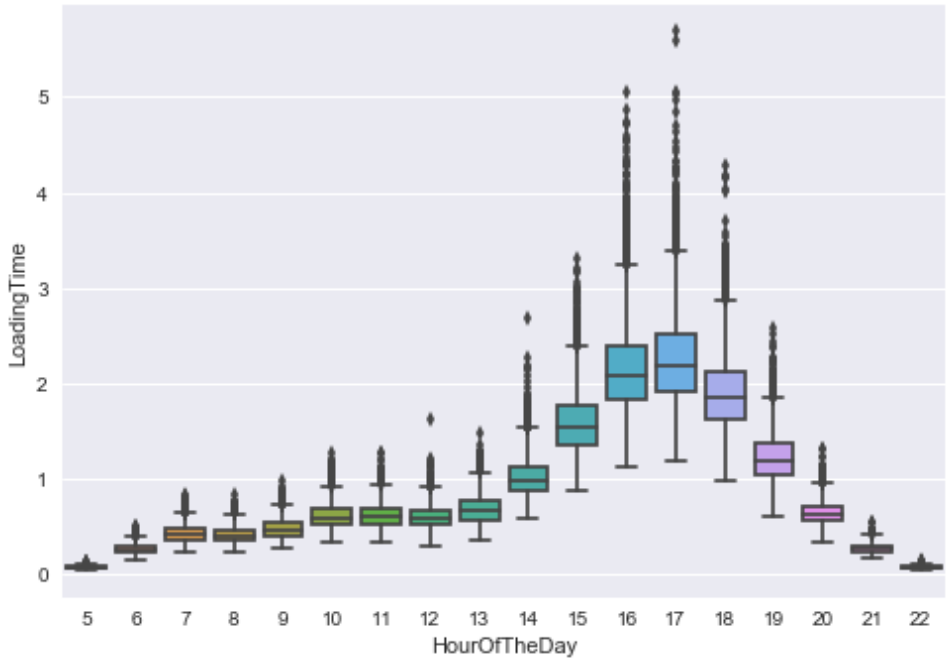


USER TRAFFIC

Model the loading time as composed of two components

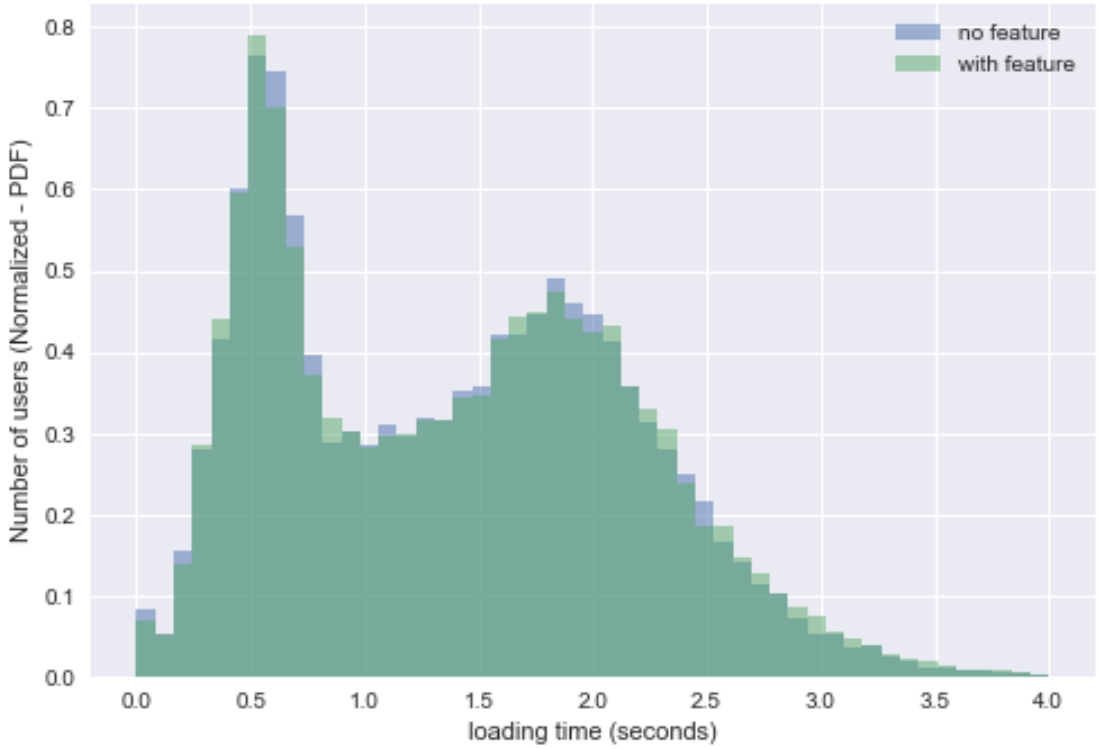
- A fixed component proportional to the number of active users
- A random component following a lognormal distribution





GO BACK TO OUR FEATURE

- Let's regenerate the dummy data from this model
  - Reference 1.3846 s
  - Feature: 1.3964 s
- Very close!
- Let's repeat the hypothesis testing with the same conditions as before
- P-value = 0.0095
- The feature does affect the loading times

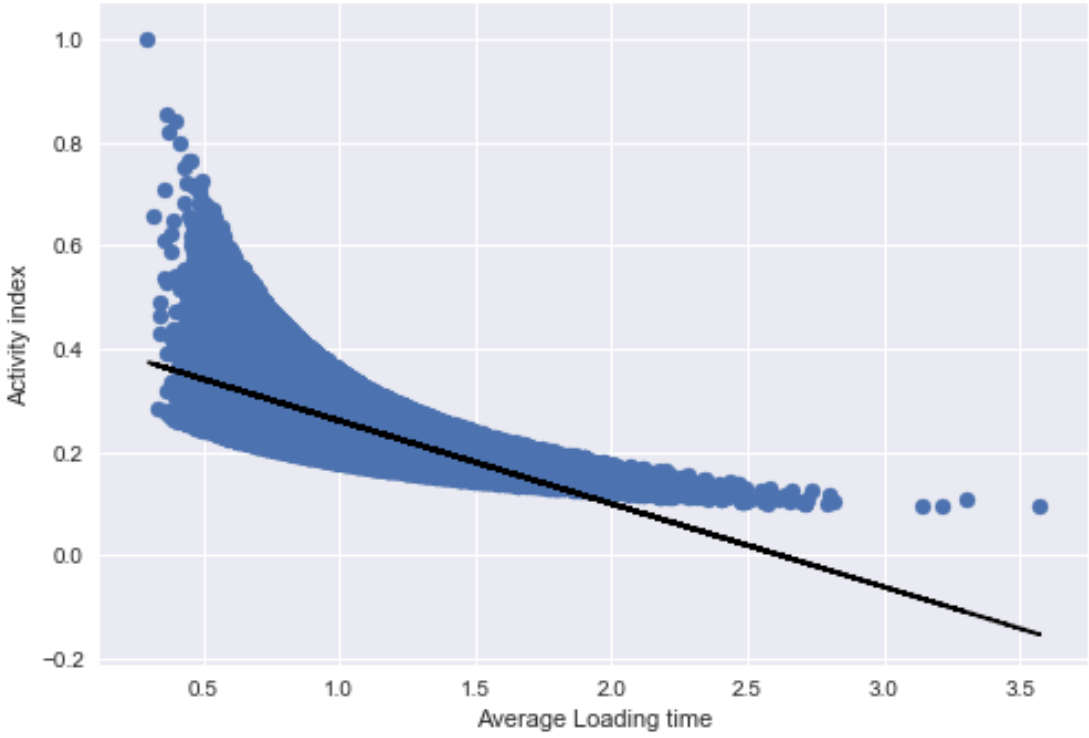


USER TRAFFIC AND LOADING TIME

- In the model, we have assumed that the loading time is affected by the traffic.
- At the same time, in real world, loading time, and in general, the performance of an application influences the users.
  - An increasing loading time would push some users to try competitors.
- Testing and measuring the impact is not easy.
- I would introduce an index measuring the activity of the users in the last month.
- Then I would look for a correlation between this value and the average loading time in the last month.

USER TRAFFIC AND LOADING TIME

- I would expect something like this
- People are more likely to use the app if the loading time is short
- In this scenario, we can also somehow predict the activity of the user given his loading time
  - If, somehow, it increases, the user will probably decrease the activity until quitting



USER TRAFFIC AND LOADING TIME

- Another way to be sure about the results, is again, statistical hypothesis testing.
- We should compare the average loading time (per user) of the users that quit and the users that are still active.

ROLLBACK THE FEATURE?

- It affects the Social feed loading time
- We have also mentioned the possible effects of an increasing loading time, which can decrease (in our synthetic data) the activity of users
- On the other end, this feature could be appreciated by the users. The appreciation could mitigate the increased loading time

USERS SATISFACTION

Model the users satisfaction or happines as composed of components

- Satisfaction to use the application (S_0)
- Penalty related to loading time P(t)

$S = S_0 - P(T)$

If the satisfaction is over a threshold α, the user keeps using the the app

$S = S_0 - P(T) > \alpha$

USERS SATISFACTION

If the loading time increases, it should be mitigated by the satisfaction delivered by the new feature S_F

$$S = S_0 + S_F - P(T') > \alpha$$

In the end, not to lose the user, it must hold:

$$S_F > P(T') - P(T)$$

ROLLBACK THE FEATURE?

In a more concrete way, we can measure the activity index already introduced.

- If, despite the increased loading time, people activity index is increased for the users with the new feature, the feature is appreciated.

Very dependent on the revenue model:

- Focus on few users but more satisfied and willing to spend.
- If the new feature makes the loading time of the social feed just too slow for some users, they will quit.
- Are they using old devices?
  - These people do not spend much on mobile applications

SECOND USE CASE – ARTIFICIAL INTELLIGENCE TRAINER

Study the impact of a change within the AI feature

Useful metrics?

- The same activity index already introduced in the first use case
- Absolute or based on the users behavior (customized for each user)

- Requirements:
  - Historical data
  - AI-coach assignments

- Assign the new feature to a shard of the users and measure the retention in the 4° week of training

SECOND USE CASE – ARTIFICIAL INTELLIGENCE TRAINER

- Assume a baseline retention of 90%
- Assign the feature to less than 2,000 people should be enough
- We could use the activity index to understand who is still using the coach
- Or, more simply, a boolean value stating if the user is still using it. Let's do this!
- Generate our dummy data (Binomial (1,0.8/0.82))
- Users still using the coach:
  - Reference 0.80%
  - Feature 0.83%

STATISTICAL HYPOTHESIS TESTING

For binomial data, the variance is directly determined by the mean

- A Z-test should be enough.
- Null hypothesis: no difference in the distributions related to the new feature
- Confidence interval: 95% (if p.value is below 5% we will reject the Null hypothesis)

Results:

- p-value: 0.0333 - The value is below the threshold (3.3%) so we reject the Null hypothesis
- With a confidence of 99%, the confidence interval would have captured the population mean

We would have failed to reject the null hypothesis. This is why it is very important to set the threshold before running the experiment

SUGGESTING A NEW HYPOTHESIS TO TEST?

- Rely on domain-experts to understand which are the factors involved in the change
  - E.g. , does user sex affects how he/she reacts to the change
- Whatever the data suggest, do not test it with the same sample of data!
- Therefore, let's assume the social domain experts list a set of features / parameters which affect or could be affected by the new feature or, more generally, by the AI-coach.
  - E.g., Sex, Age, behavior on the social feed
  - Check if we own data related to these aspects and test if there is something going on there
  - Could we be interested to this possible new relationship?
  - Test it on a new sample!

DOES SEX AFFECT THE RETENTION INDEX?

Let's imagine our experts have suggested the sex could influence the appreciation of the new AI feature

- Maybe, boys are more suggested by this AI-stuff and, for this reason, more positive towards possible wrong recommandations
- Try measure influences and correlations among the retention index and the sex

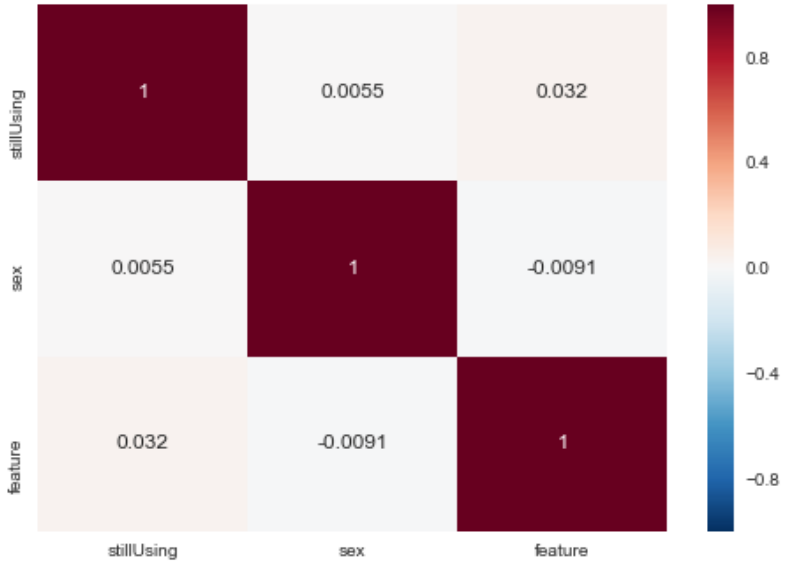
DOES SEX AFFECT THE RETENTION INDEX?

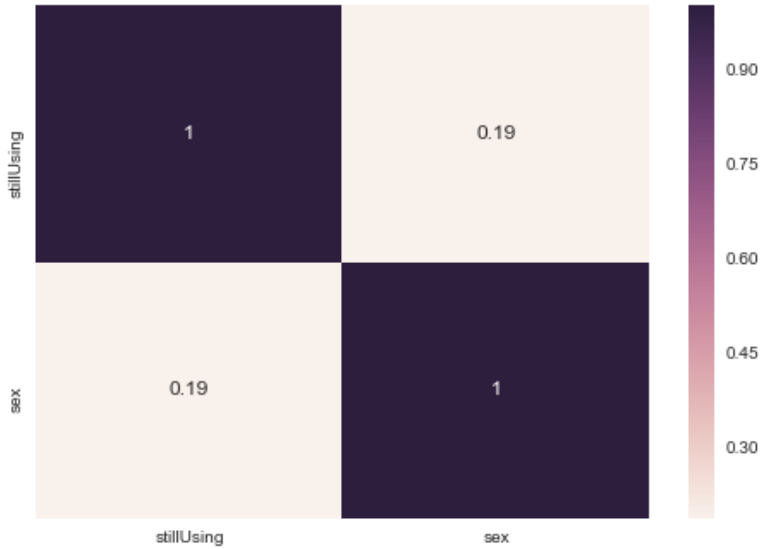
Not much correlated at a first sight (but here the users with the issue is less than 1/10 of the whole population)

However, let's focus on the users with the new features enabled.

Sex and retention are quite correlated for the users with the new feature

So we should recollect data and statistically test this hypothesis.





BAYESIAN APPROACH

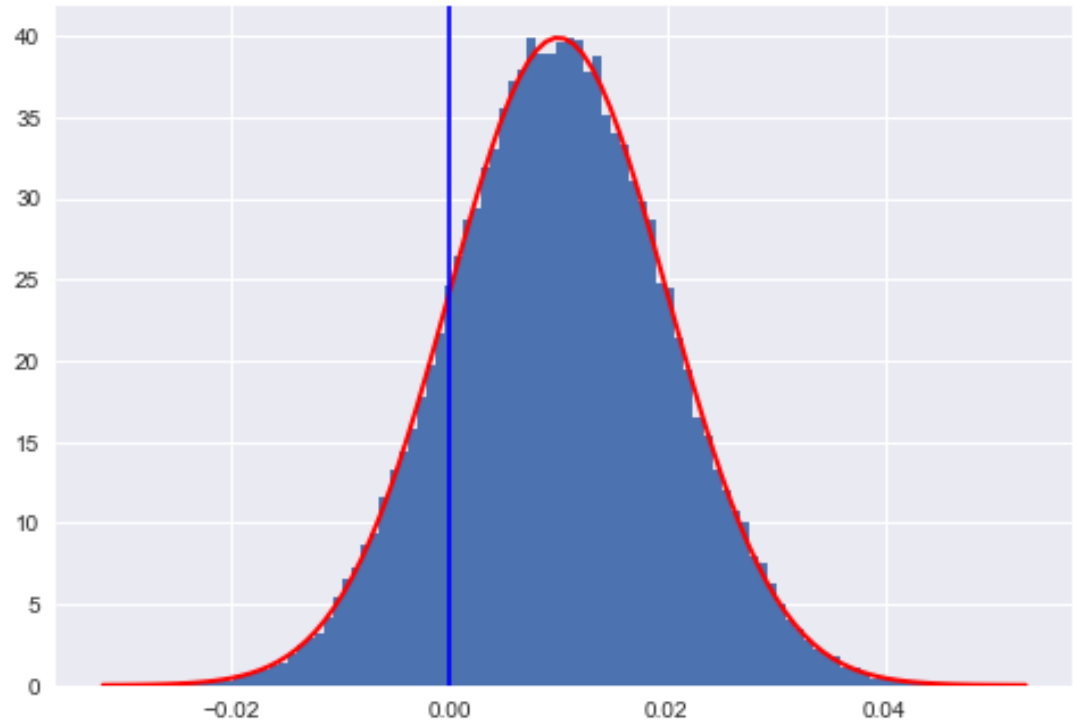
We have used a frequentist approach:

- Making predictions only with data from the current experiment

In a Bayesian approach, it is used a prior probability which is combined with the current experiment data to draw conclusions

Most important pro:

- Managing a real distribution a not a p-value. This is way more interpretable and easy to communicate.



BAYESIAN APPROACH

In our case we could have obtained a shape similar to this. Assuming an historical analysis we could have obtained all the parameters to obtain the distribution $P(\mu_{feature} - \mu_{ref})$

The area under the curve below 0 could have meant the probability and the confidence to say that $\mu_{feature} < \mu_{ref}$

I am not an expert in this environment but I really would like to deepen it!