Machine Learning and Pattern Recognition

A High Level Overview

Prof. Anderson Rocha
(Main bulk of slides kindly provided by **Prof. Sandra Avila**)
Institute of Computing (IC/Unicamp)

MC886/MO444

Types of Machine Learning Systems

Types of Machine Learning Systems

Trained with human supervision (or not)

Supervised *vs.*
Unsupervised *vs.*
Reinforcement learning

Can learn incrementally on the fly (or not)

Online *vs.*
Batch Learning

How they generalize

Instance based *vs.*
Model based learning

Types of Machine Learning Systems

Trained with human supervision (or not)

Supervised *vs.*
Unsupervised *vs.*
Reinforcement learning

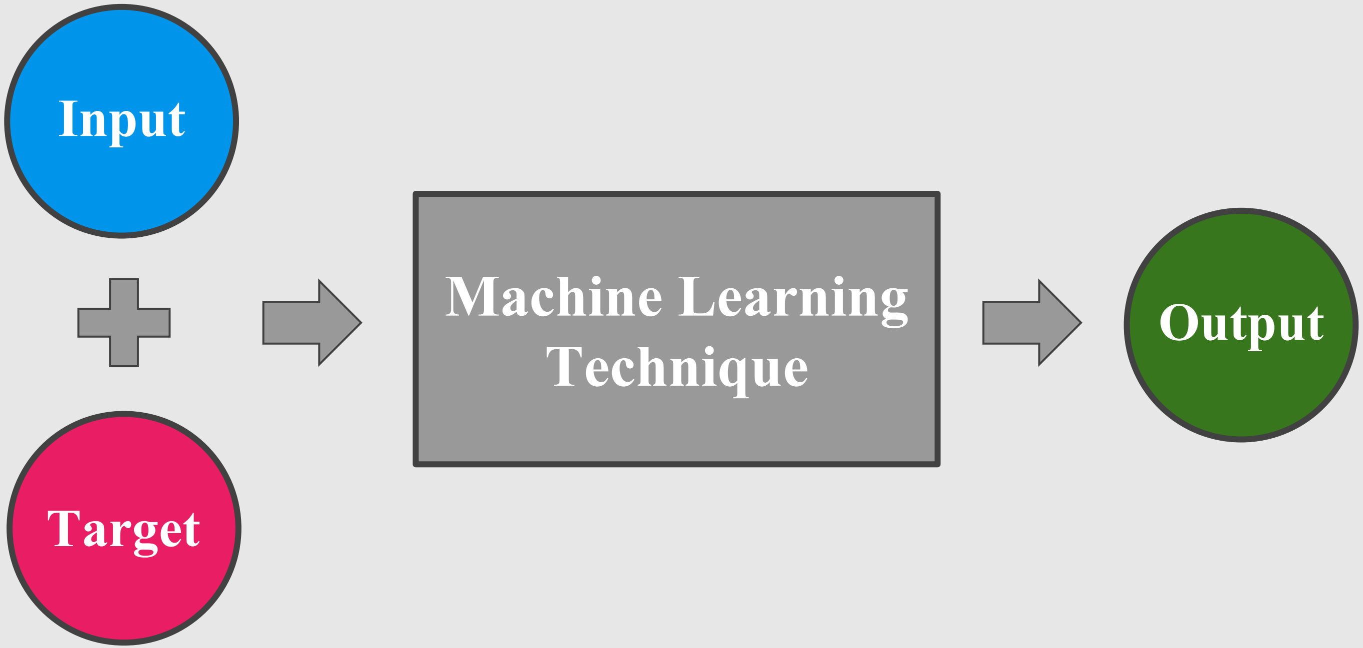
Can learn incrementally on the fly (or not)

Online *vs.*
Batch Learning

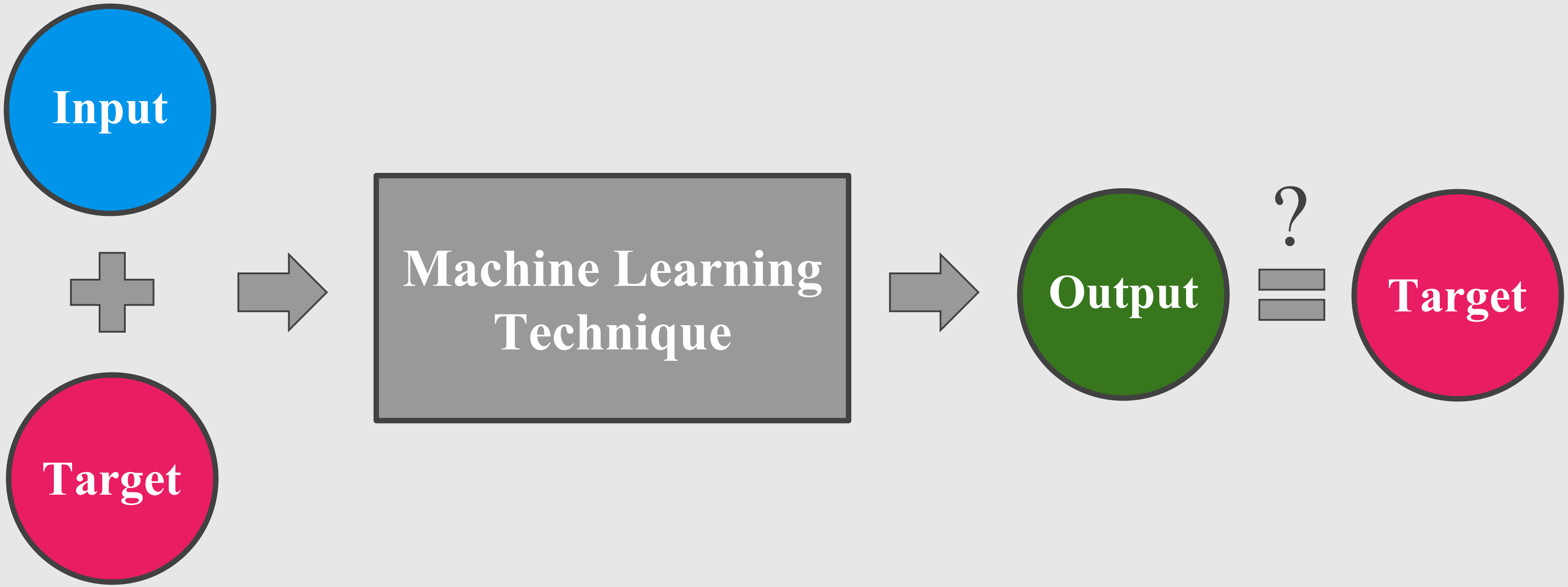
How they generalize

Instance based *vs.*
Model based learning

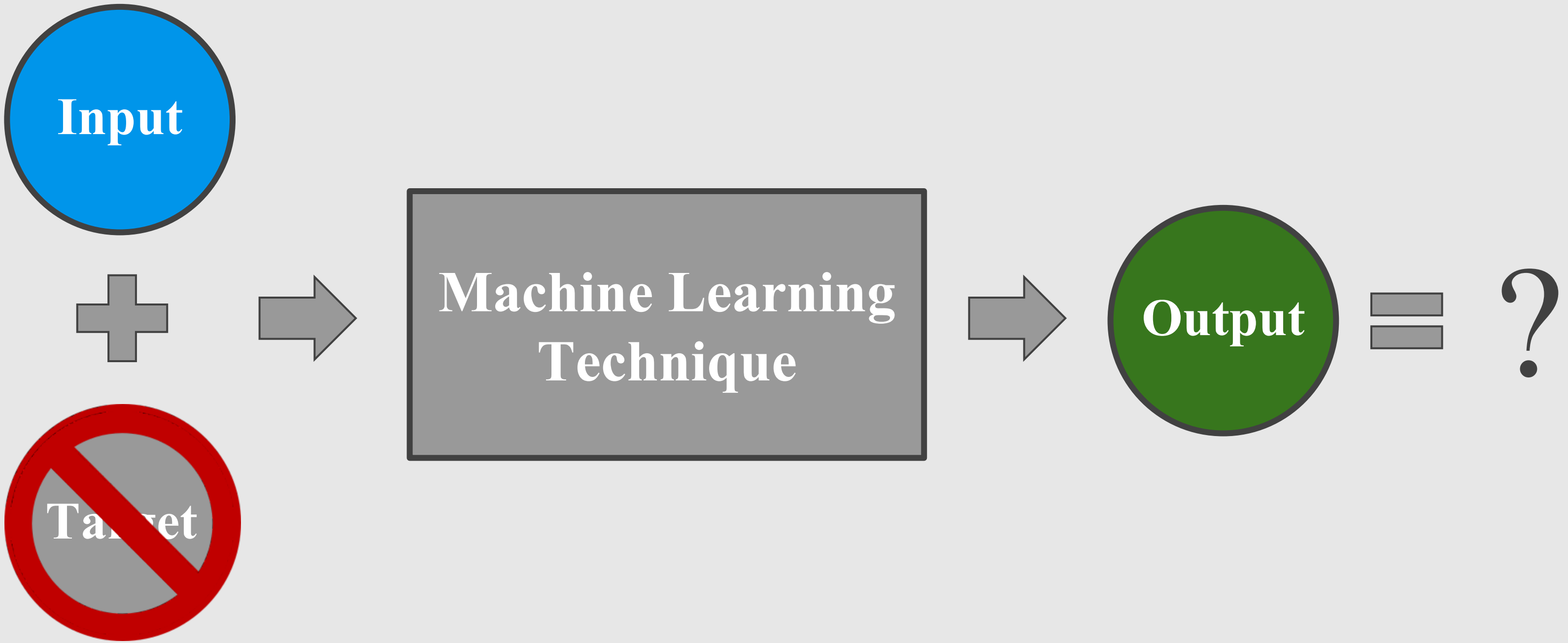
Supervised Learning



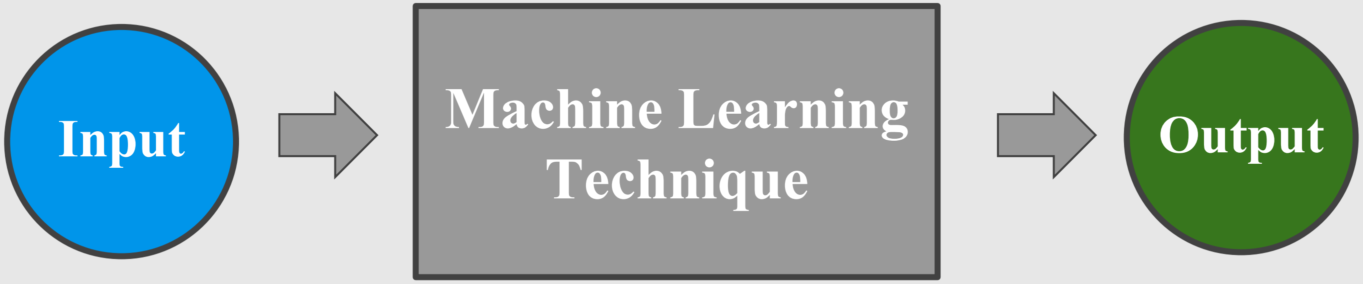
Supervised Learning



Unsupervised Learning



Unsupervised Learning



The goal of unsupervised learning is **to find patterns** in the data, and build new and useful representations of it.

Unsupervised Learning

Clustering algorithm tries to detect similar groups.

Dimensionality reduction tries to simplify the data without loosing too much information.

Applications

- Social network analysis
- Market segmentation
- Information compression
- Information retrieval
- ...

Today’s Agenda

- Clustering
  - k-Means Algorithm
  - Optimization Objective
  - Random Initialization

Clustering

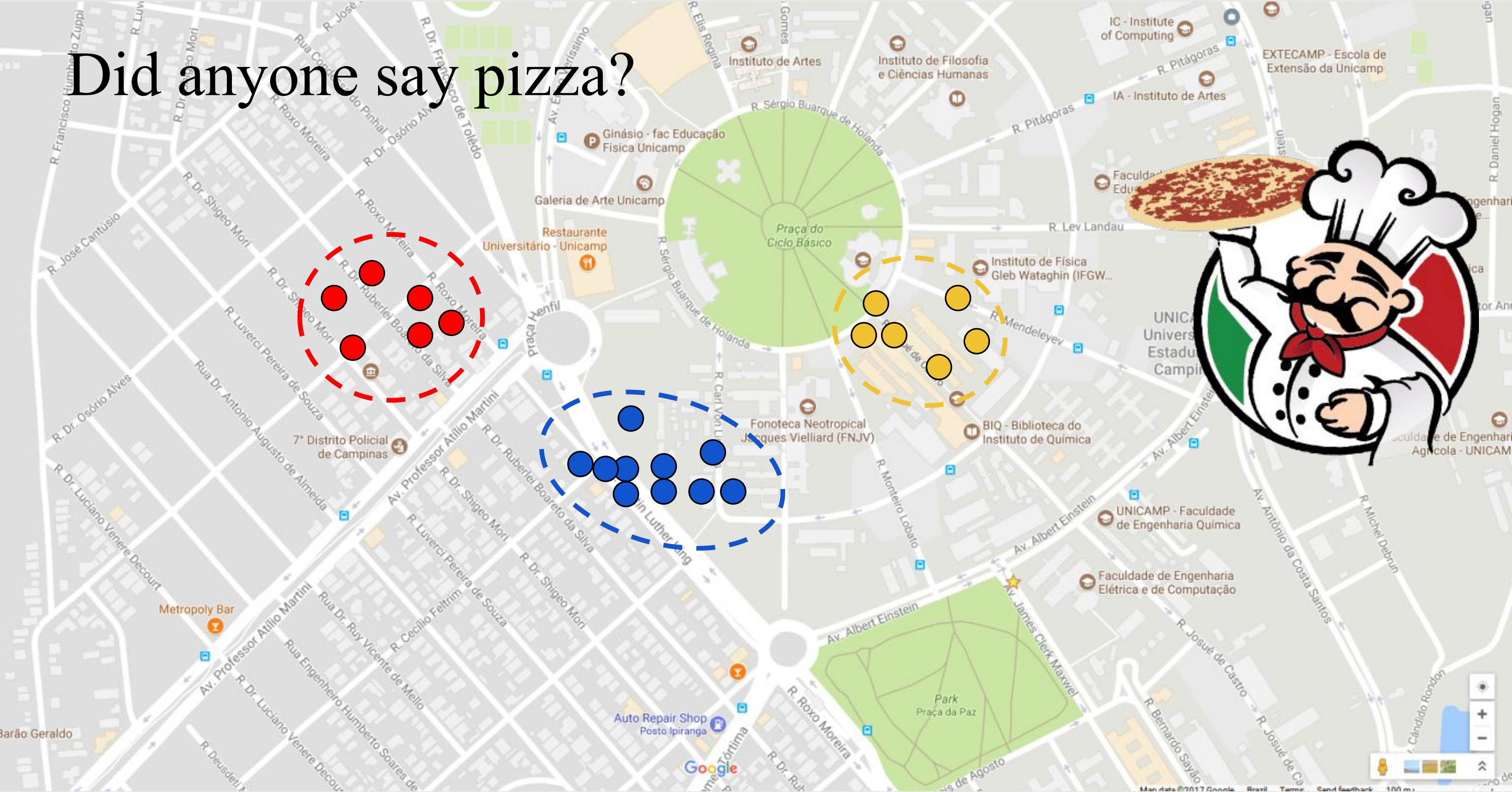
k-Means Algorithm

Did anyone say pizza?

https://www.youtube.com/watch?v=IpGxLWOIZy4

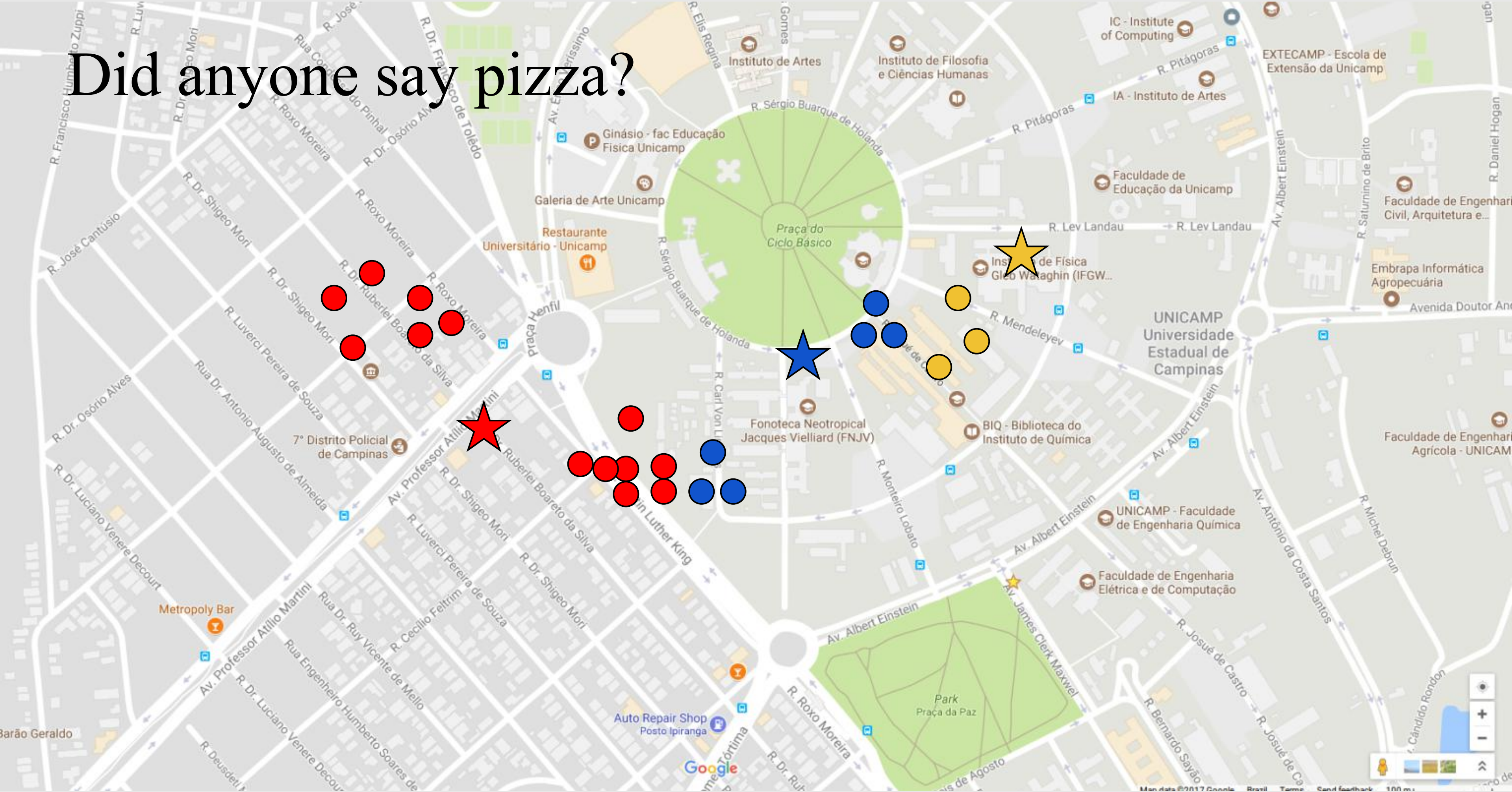
Did anyone say pizza?

Did anyone say pizza?



Did anyone say pizza?

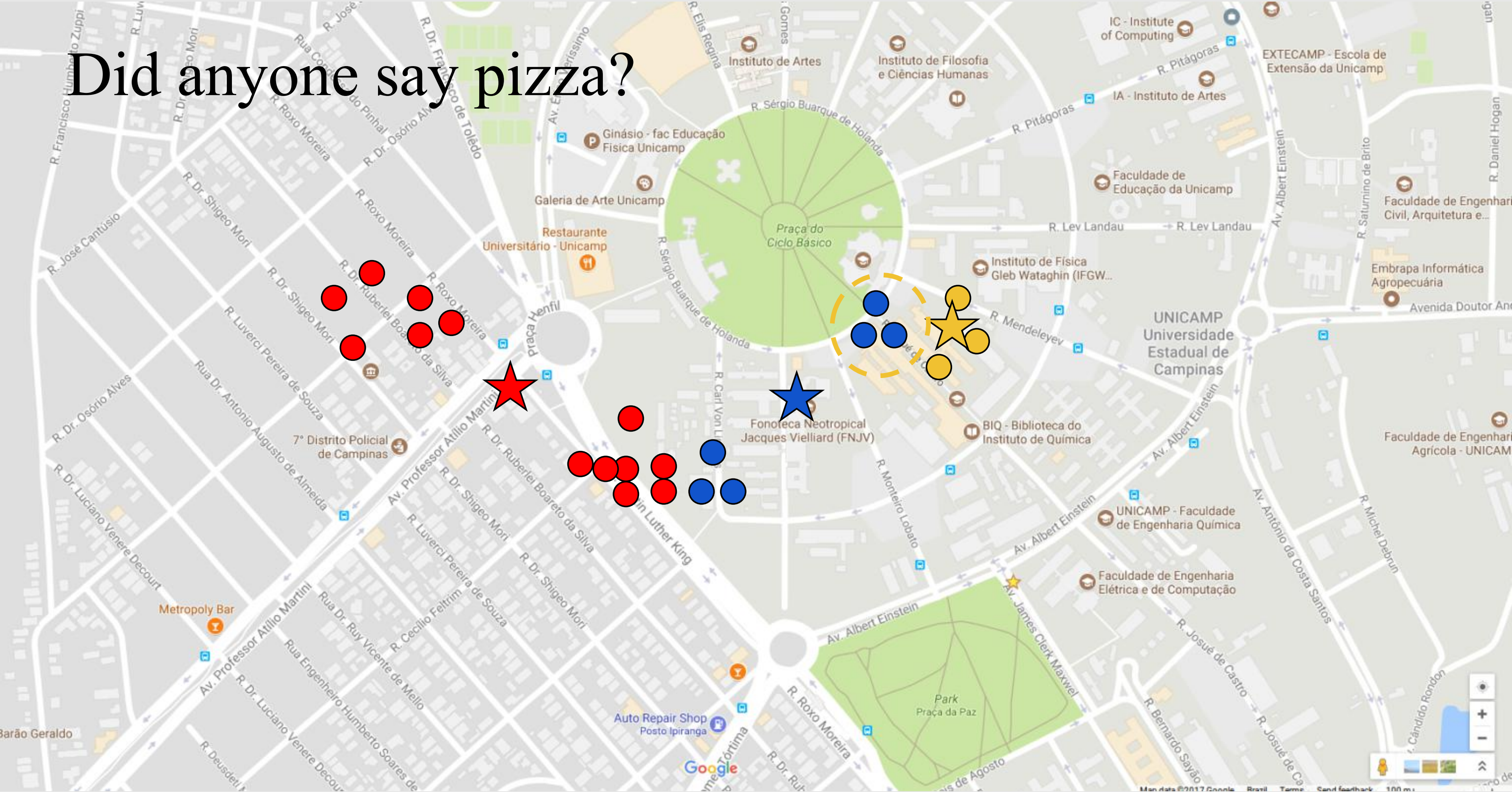
Did anyone say pizza?



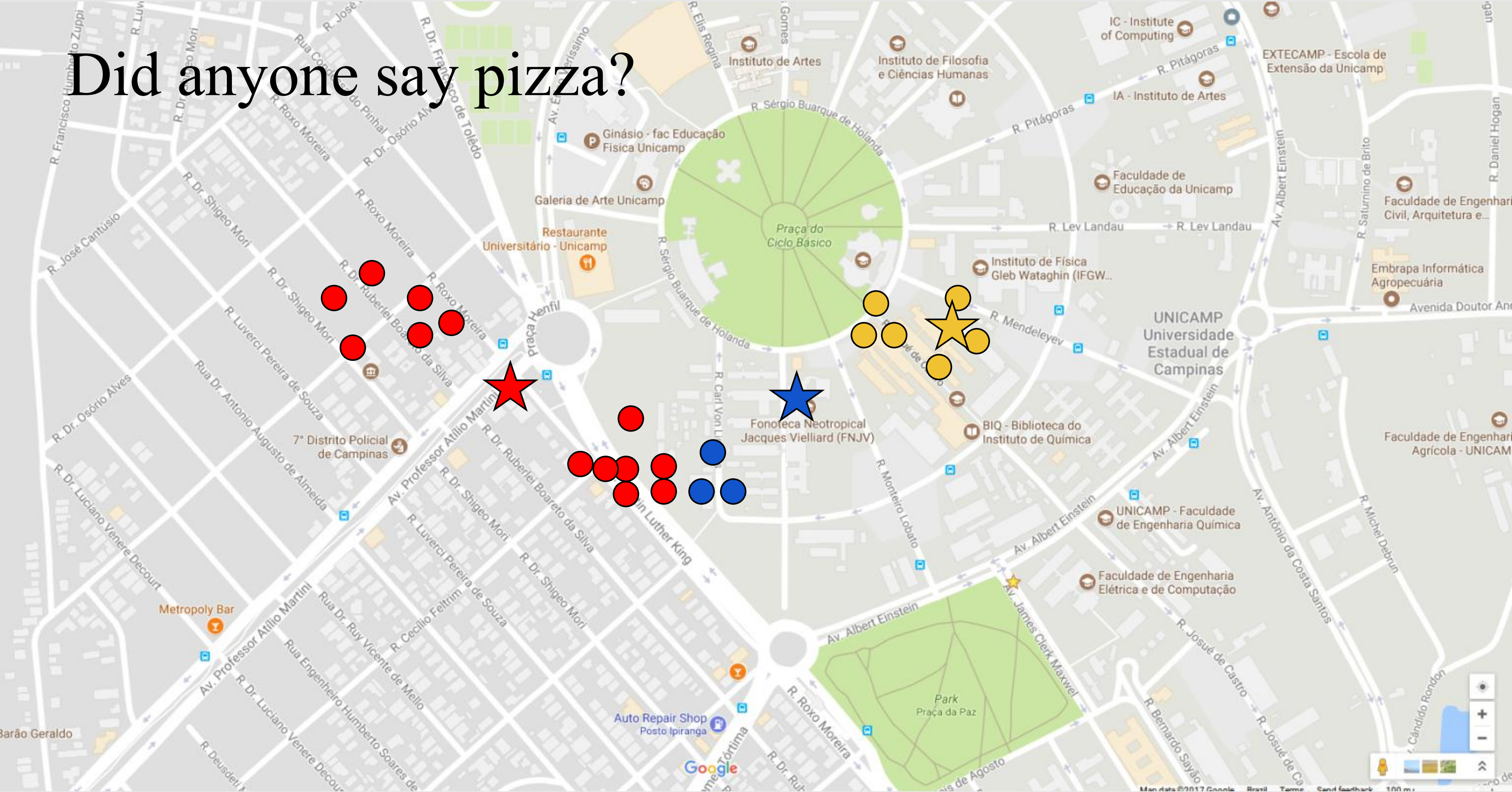
Did anyone say pizza?

Did anyone say pizza?

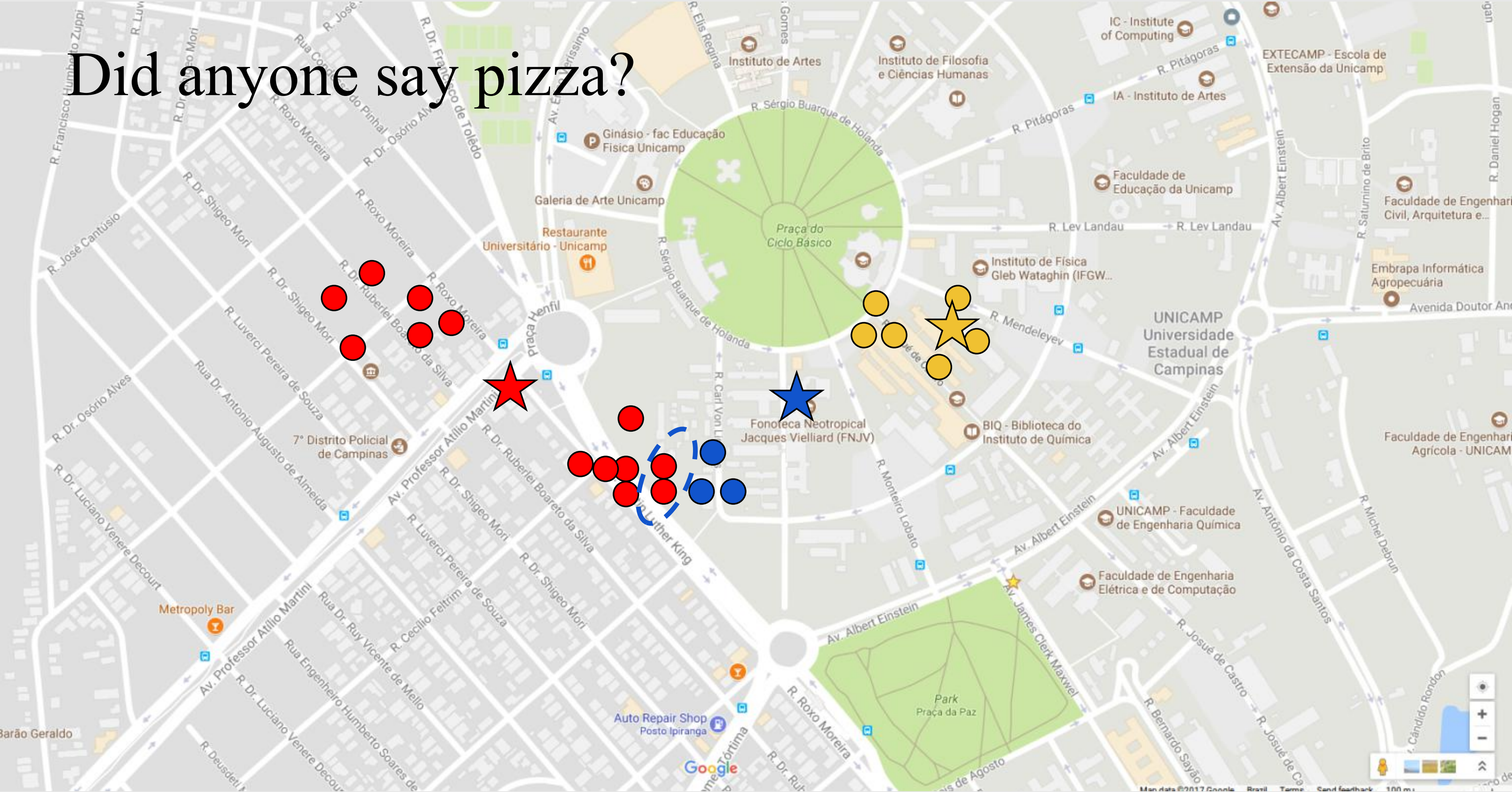
Did anyone say pizza?



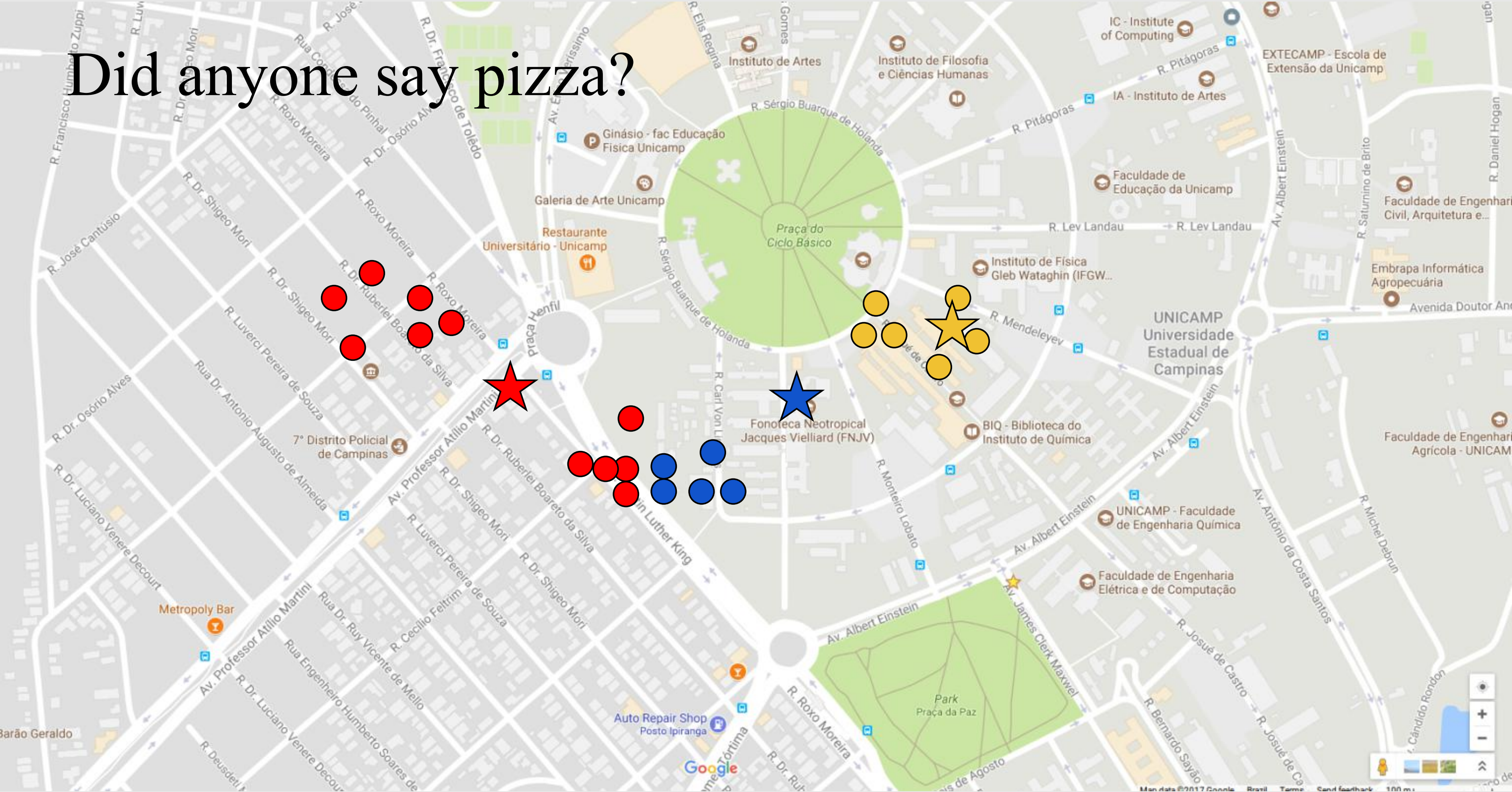
Did anyone say pizza?



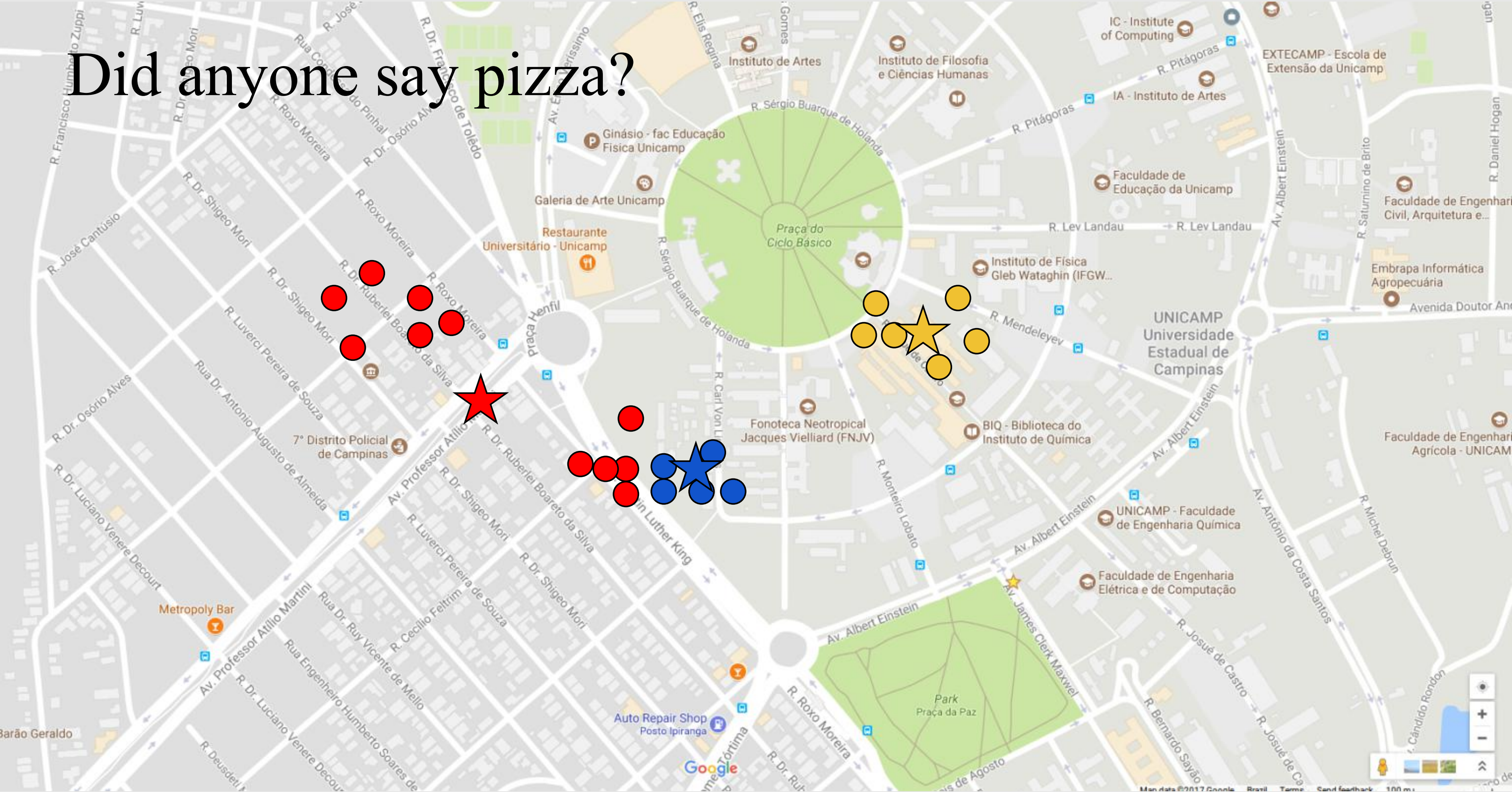
Did anyone say pizza?



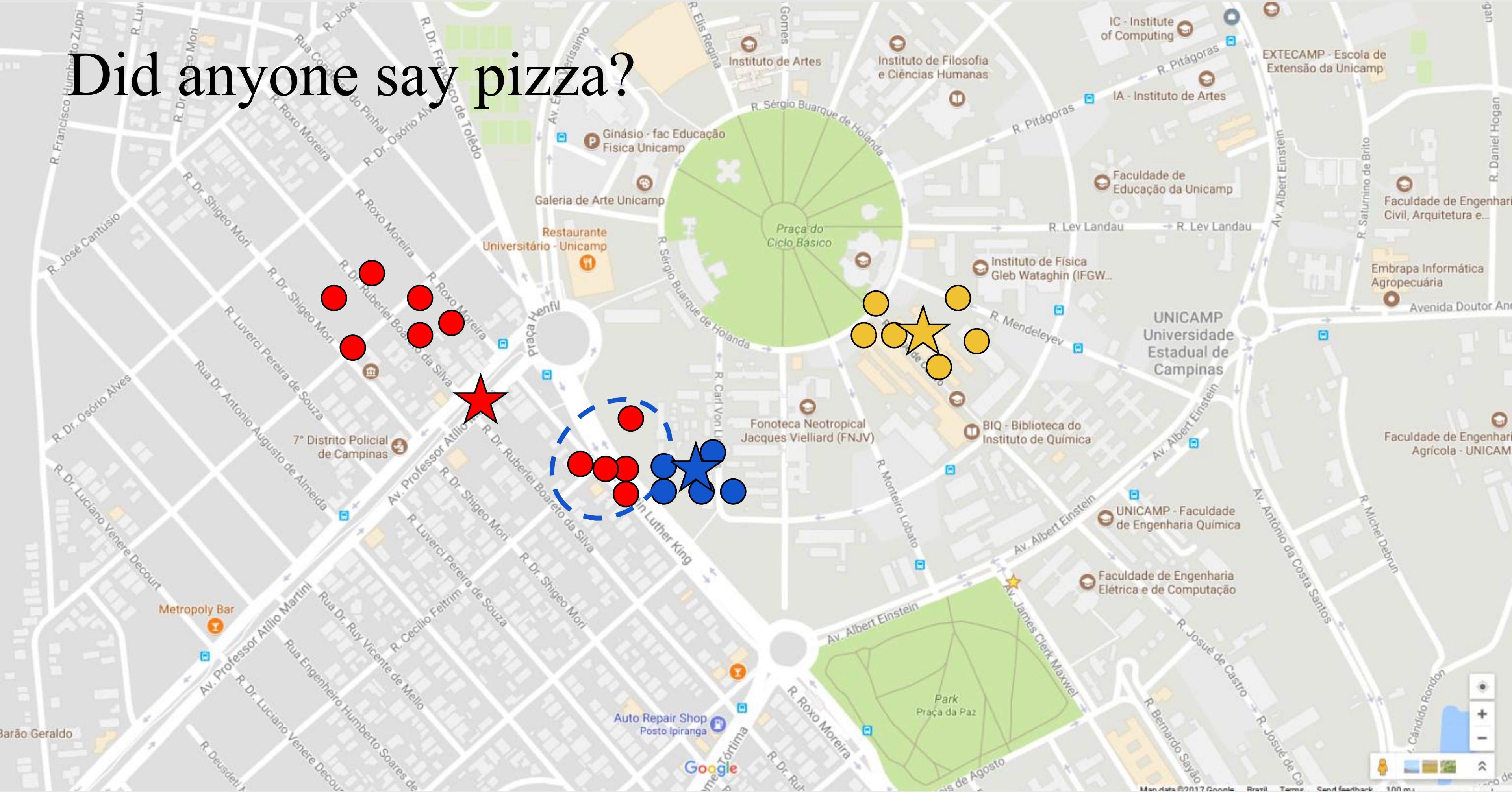
Did anyone say pizza?



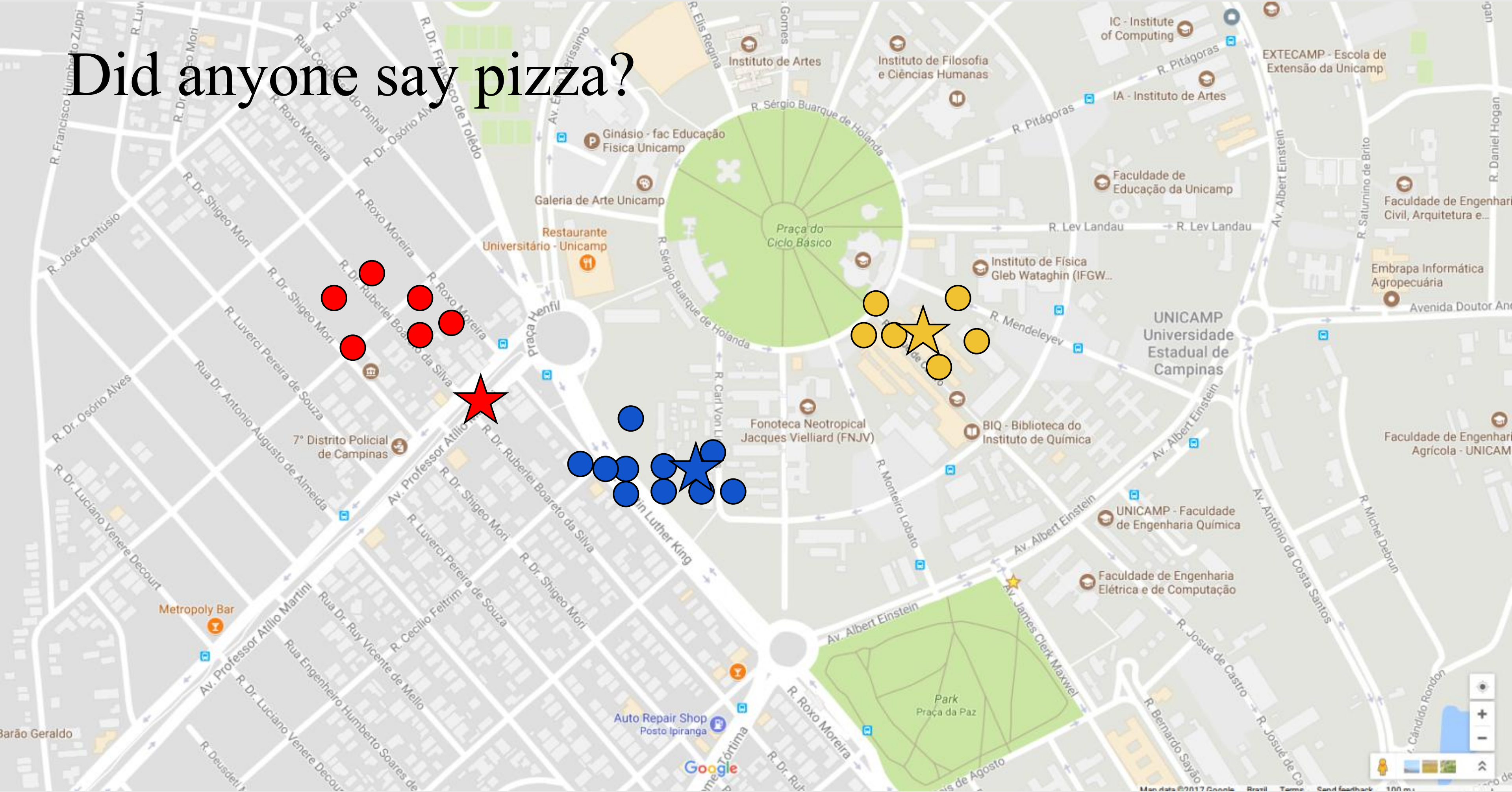
Did anyone say pizza?



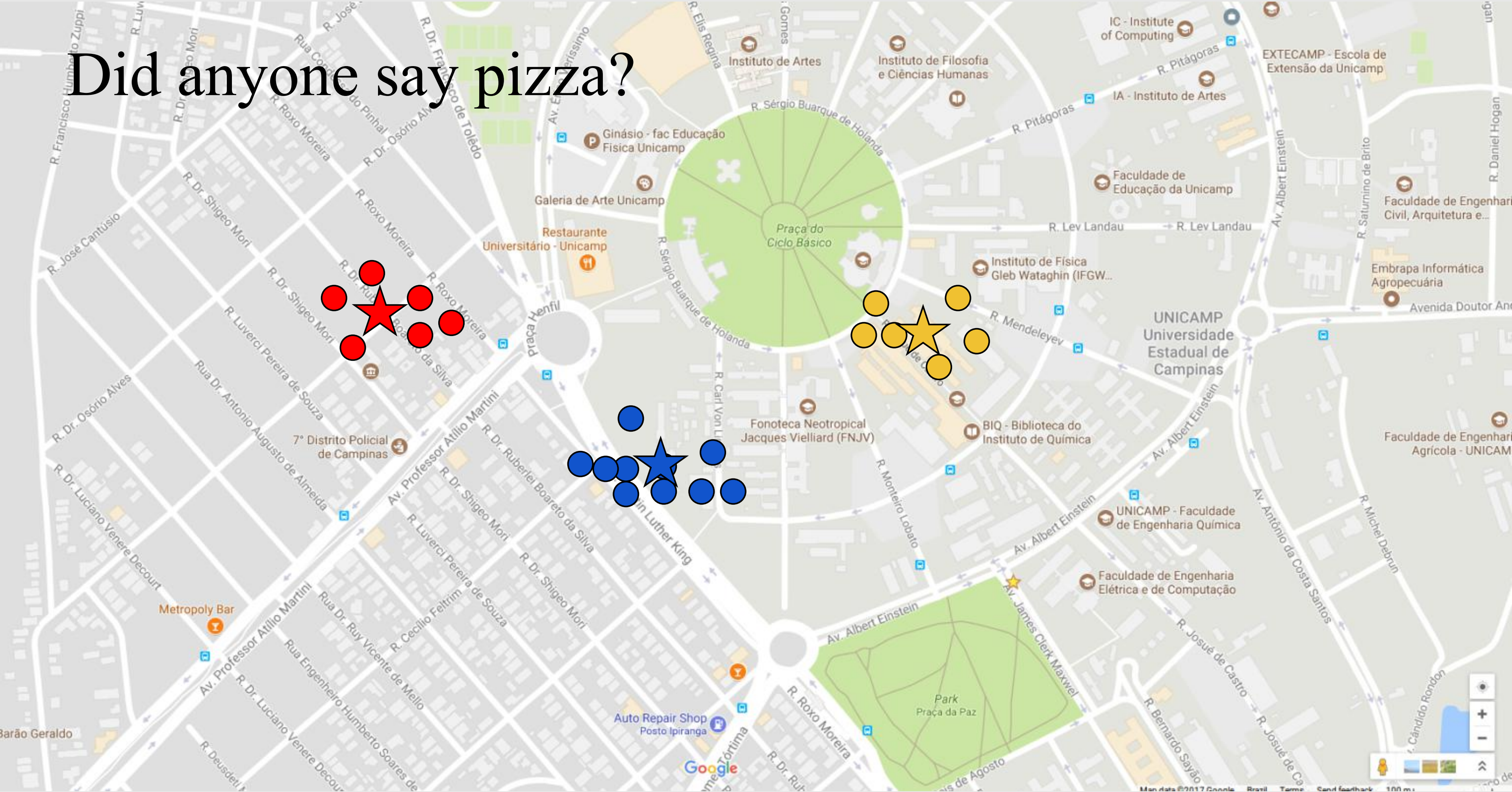
Did anyone say pizza?



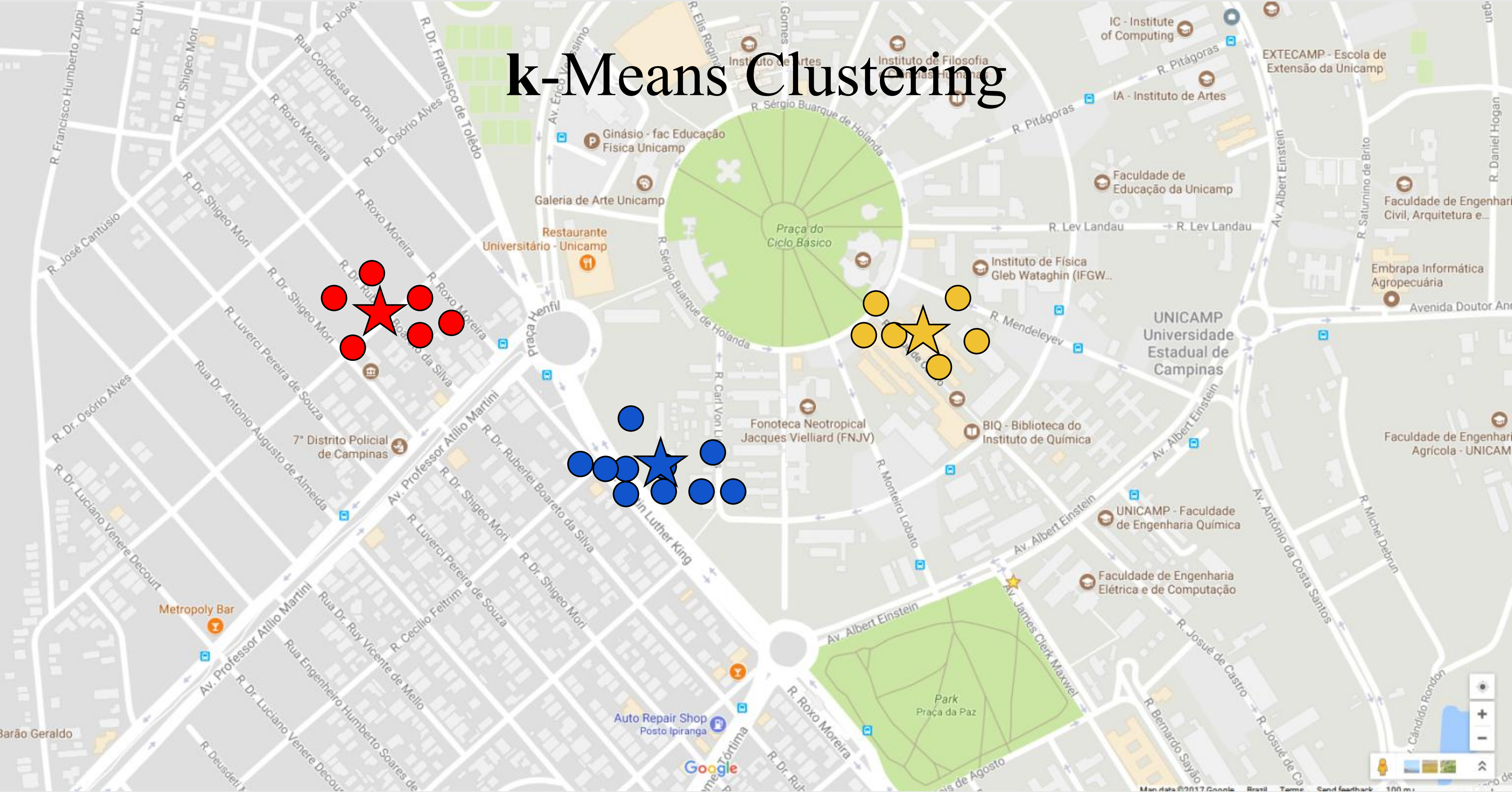
Did anyone say pizza?



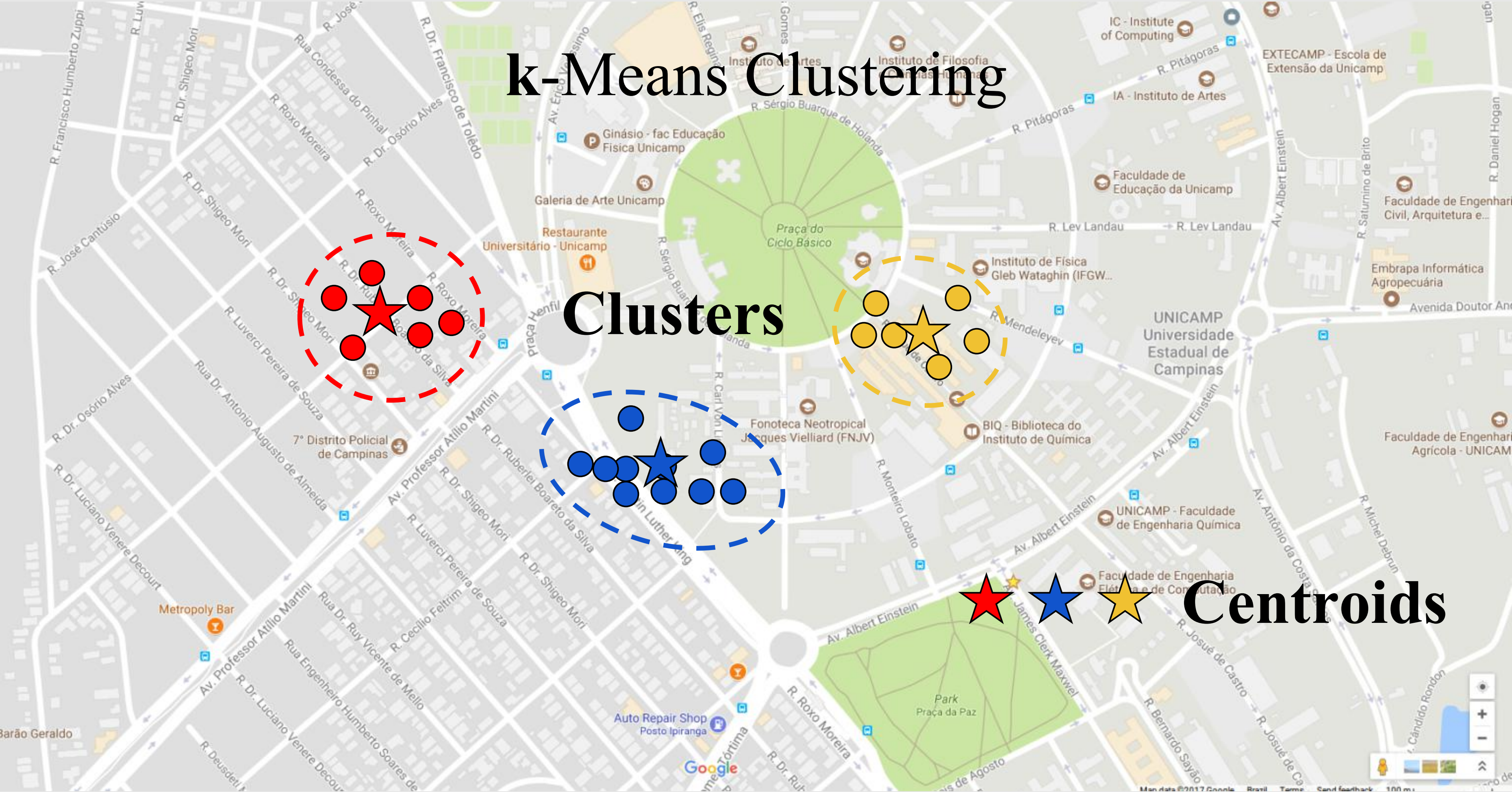
Did anyone say pizza?



k-Means Clustering



k-Means Clustering



k-Means: Image Segmentation

Credit: Christopher Bishop

k-Means: Image Segmentation

Original | K = 10 | K = 3 | K = 2

Credit: Christopher Bishop

k-Means: Image Segmentation

Original

K = 10

K = 3

K = 2

Credit: Christopher Bishop

k-Means: Image Segmentation

Original

K = 10

K = 3

K = 2

Credit: Christopher Bishop

k-Means: Image Segmentation

Original | K = 10 | K = 3 | K = 2

Credit: Christopher Bishop

k-Means Algorithm

1. **Define the *k* centroids.**
2. **Find the closest centroid & update cluster assignments.**
3. **Move the centroids to the center of their clusters.**
4. **Repeat steps 2 and 3** until the centroid stop moving a lot at each iteration.

k-Means Algorithm

1. **Define the *k* centroids**. Initialize these at random.

k-Means Algorithm

1. **Define the *k* centroids.**
2. **Find the closest centroid & update cluster assignments.** Assign each data point to one of the *k* clusters. Each data point is assigned to the nearest centroid's cluster (Euclidean distance).

k-Means Algorithm

1. **Define the *k* centroids.**
2. **Find the closest centroid & update cluster assignments.**
3. **Move the centroids to the center of their clusters.** The new position of each centroid is calculated as the average position of all the points in its cluster.

k-Means Algorithm

1. **Define the *k* centroids**.
2. **Find the closest centroid & update cluster assignments.**
3. **Move the centroids to the center of their clusters.**
4. **Repeat steps 2 and 3** until the centroid stop moving a lot at each iteration (i.e., until the algorithm converges).

k-Means Algorithm

Input:

- ➔ K (number of clusters)
- ➔ Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

k-Means Algorithm

Randomly initialize K cluster centroids $\mu_1,\ \mu_2,\ \ldots,\ \mu_K \in \mathbb{R}^n$

k-Means Algorithm

Randomly initialize K cluster centroids $\mu_1, \ \mu_2, \ \ldots, \ \mu_K \in \mathbb{R}^n$

repeat {

}

k-Means Algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

}

k-Means Algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

repeat {

$$\min_k \|x^{(i)} - \mu_k\|$$

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

}

k-Means Algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

repeat {

Cluster assignment step

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

}

k-Means Algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

for $k = 1$ to K

μ_k := mean of points assigned to cluster k

}

k-Means Algorithm

Randomly initialize K cluster centroids $\mu_1, \ \mu_2, \ \ldots, \ \mu_K \in \mathbb{R}^n$

repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

for $k = 1$ to K

μ_k := mean of points assigned to cluster k

} **Move centroid step**

k-Means Algorithm

Q: What if a cluster doesn't have any element?

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

for $k = 1$ to K

μ_k := mean of points assigned to cluster k

}

k-Means Algorithm

Q: What happens when we don't have very well separated clusters?

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

for $k = 1$ to K

μ_k := mean of points assigned to cluster k

}

Clustering

Optimization Objective

k-Means Optimization Objective

$c^{(i)}$ = index of cluster (from 1 to K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

k-Means Optimization Objective

$c^{(i)}$ = index of cluster (from 1 to K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

$$x^{(i)} = 2, \quad c^{(i)} = 2, \quad \mu_{c^{(i)}} = 2$$

k-Means Optimization Objective

$c^{(i)}$ = index of cluster (from 1 to K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

Optimization objective:

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|$$

$$\min_{\substack{c^{(1)}, \dots, c^{(m)} \\ \mu_1, \dots, \mu_K}} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$$

k-Means Optimization Objective

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid **closest** to $x^{(i)}$

for $k = 1$ to K

μ_k := mean of points assigned to cluster k

}

Clustering

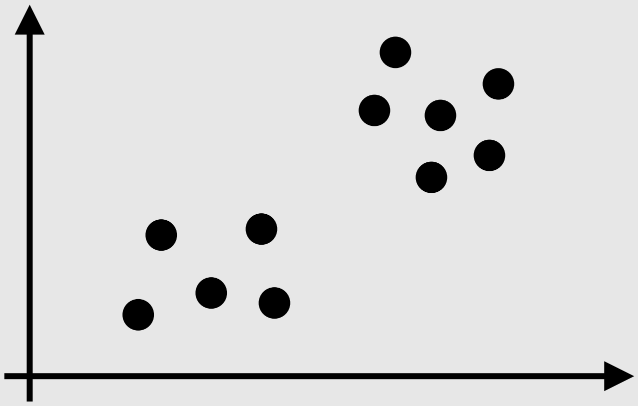
Random Initialization

Random Initialization

Should have $K < m$.

Randomly pick K training examples.

Set $\mu_1, \ldots, \mu_K$ equal to these K examples.



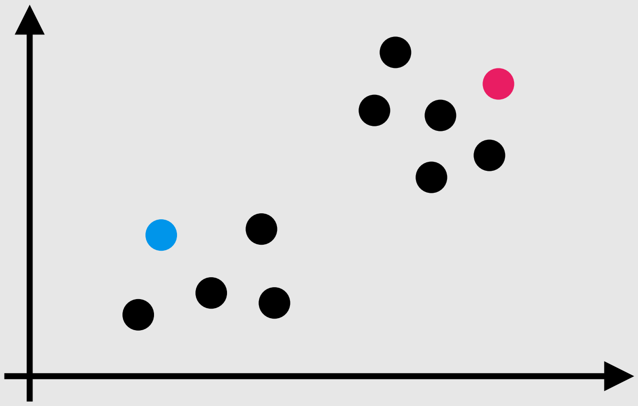
Random Initialization

Should have $K < m$.

Randomly pick K training examples.

Set $\mu_1, \ldots, \mu_K$ equal to these K examples.

$K = 2$



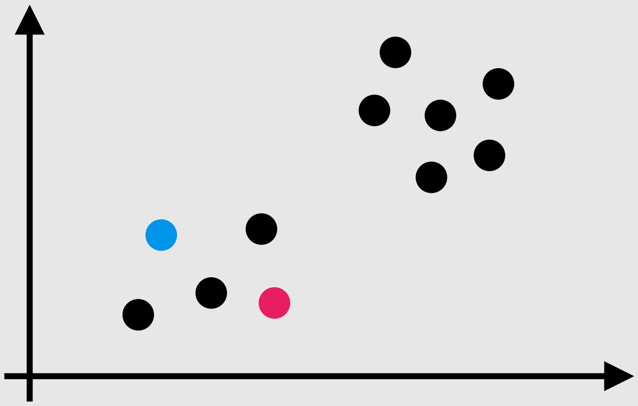
Random Initialization

Should have $K < m$.

Randomly pick K training examples.

Set $\mu_1, \ldots, \mu_K$ equal to these K examples.

$K = 2$



Random Initialization

for $i = 1$ to 100 {

Randomly initialize k-Means.

Run k-Means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$

Compute cost function J.

}

Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$.

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

naftali harris

Blog | About | Contact | I'm Feeling Lucky

Visualizing K-Means Clustering

January 19, 2014

Suppose you plotted the screen width and height of all the devices accessing this website. You'd probably find that the points form three clumps: one clump with small dimensions, (smartphones), one with moderate dimensions, (tablets), and one with large dimensions, (laptops and desktops). Getting an algorithm to recognize these clumps of points without help is called *clustering*. To gain insight into how common clustering techniques work (and don't work), I've been making some visualizations that illustrate three fundamentally different approaches. This post, the first in this series of three, covers the k-means algorithm. To begin, click an initialization strategy below:

Mini-batch **k**-Means

- Uses mini-batches to reduce the computation time, while still attempting to optimize the same objective function.

- Converges faster than [k-Means], but the quality of the results is reduced.

References

Machine Learning Books

- Pattern Recognition and Machine Learning, Chap. 9 “Mixture Models and EM”
- Pattern Classification, Chap. 10 “Unsupervised Learning and Clustering”

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 8