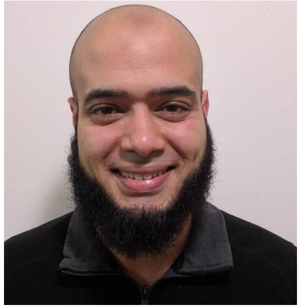
# Machine Learning
# Toward ML Seniority

**Mostafa S. Ibrahim**
*Teaching, Training and Coaching for more than a decade!*

*Artificial Intelligence & Computer Vision Researcher*
*PhD* from Simon Fraser University - Canada
*Bachelor / MSc* from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



# Juniors vs Seniors

- In practice, juniors might receive simple ML projects or a small task
- However, seniors in big companies might take a big and challenging cycle
- The senior sometimes need to initiate the project idea itself
- This means you find and idea and sell it to the manager
- This also involves a great deal of ambiguity and uncertainty

# What: The idea

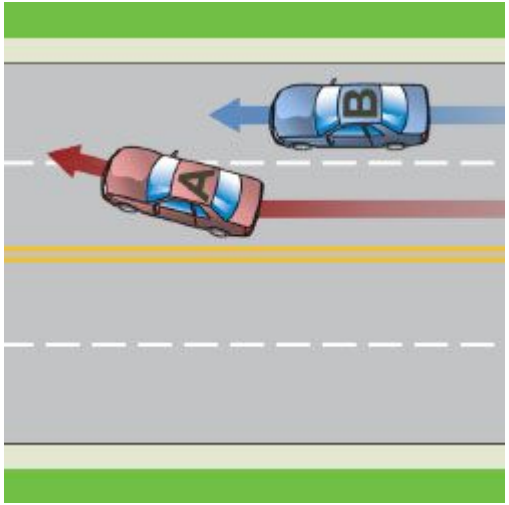
- A good senior doesn’t wait till the end of the current (ML) project and then asks what is the next
  - If there is no next, then there is no need to hire you!
- If it seems nothing is a next project, work on finding 2-3 high level ML ideas that can bring value to the company
  - Tool helps an internal team
  - New feature for the customer
- This requires:
  - Understanding the business
  - Innovative thinking **skills**
  - Critical thinking **skills**
  - Presentation and Communication **skills**
  - Ability to deal with ambiguity

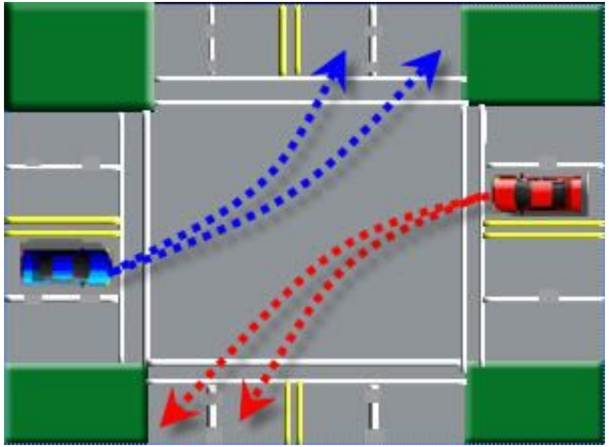
# Why: The justifications

- Managers will aggressively ask many WHY questions
- We need to justify strongly the importance of this question
- If you can, be ready with numbers/evidence that support your claims about the importance of this features
- No one wants 4-6 months with no useful outcome

# What: The Scenarios

- You might need to state several **scenarios** where the feature is useful
- This is also useful for the **demos** we present higher management
- Example, in autonomous driving, we introduce feature that will reduce the **collisions** between the cars. We show details about that and demo it





# What: The literature review

- Is this a new problem? Or there are relevant work out there?
- What kind of low-level challenges in this problem?
- What are the data challenges in this problem?
- What are the different pros and cons of current approaches?
- Can we just use an available approach?
  - What is the licence of the project? Do we need to buy the code?
- What are the publicly available datasets?
  - Diverse enough? Do we need to enhance it more?
  - What is the licence of the data? Do we need to buy the data?

# What: the data?

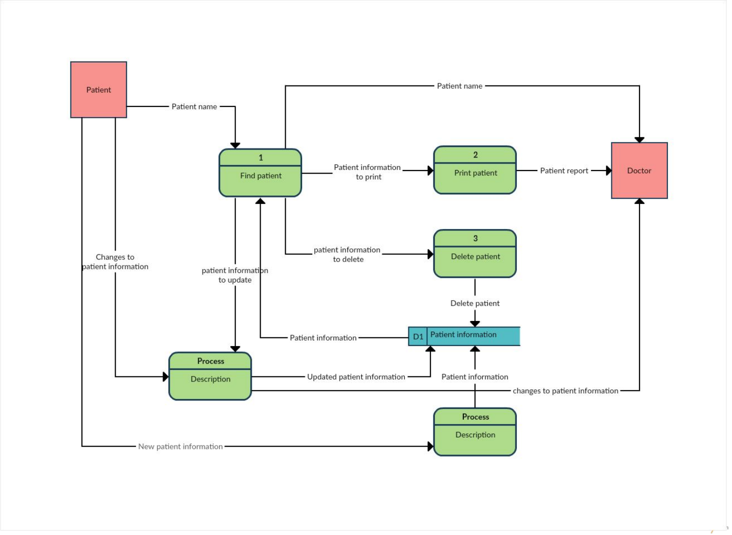
- If we need to collect our own data:
- How many examples?
- How/Who will collect them? How much time/money do we need?
- Do we need annotation? How/Who/Time/Money
    - Is it internal team? Crowdsource for annotation like Amazon Mechanical Turk?
- Will we generate **synthetic** data?
    - Will your team build it?
    - Do you want to depend on another team?
    - Will data be generated at once?
      Or a tool to generate more examples?
    - What are the requirements of the generation? The tool?

# How: The approach

- What are the potential wayS to tackle the problem?
- What are the pros and the cons?

# How: System Flow

- Sometimes we develop a simple module. Sometimes, it is a system of several interacting module
- Which modules are yours?
- Which modules you will use/integrate with?



# What and When: Resources

- What are the machine resources do you need? Local or cloud? Storage?
- How many human resources do you need?
- **When** can you deliver the project? What are your milestones? Your deliveries?

# Conflict of interests

- Typically the engineer would like to expand the timeline/resources to give them space to finish the task
- Managers on the other side push toward shrinking time/resources
  - Manager: Why do you need 6 months? 4 are not enough?
    - Why 5 persons? A Similar project needed 3 only?
    - Can you annotate by yourself instead of hiring an external team?
- Be ready to convince others and push on the opposite side

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."