Track AI/ML

Final Product :

A. Pose detection model

1. Make algorithm and train it with the dataset
2. Make rest API of it
3. Input the video/data from the frontend
4. Make frontend to display it to user

B. Sentimental analysis

1. Make algorithm and train it with the dataset
2. Make rest API of it
3. Input the video/data from the frontend
4. Make frontend to display it to user

Before jumping to the project, make yourself clear with what you are going to build, either you want to build the project on a given dataset or you want to make a project to generate that dataset.
To begin the project you follow the process:-

1. Data Acquisition
   You can get the data from Kaggle or some other resources but make sure your dataset is clean and free from redundancy
   This will help to make accurate predictions
2. Data elimination
   You should remove the unwanted or redundant data.
3. Data processing
   As this is the most crucial step to perform, you can use a heatmap to find the relations among the fields and get the best dataset to train the model

Note-
As there are many algorithms to get the project done, don't fool yourself by overfitting the model.

Good to have a feature

- Slang language detection in your sentimental analysis
- You can provide the real-time pose prediction for the model

For frontend: -
You can choose to be simple design as this is to give your project an edge to show your project to anyone and anywhere.
For learning frontend, you can go through with the basics.

Some basics to began:

1. **Choose a language (Python/R)**: I would recommend Python over R as it is a more general programming language implementing the OOP concepts which are very necessary for making good scalable codes.
   You can also use Python to deploy your ML models directly via Flask services.
   Do install jupyter notebook or Anaconda(It contains all necessary IDE even jupyter notebook).

2. **Dive into the ML algorithm:** Now, you are ready to learn a new ML algorithm that helps you to predict the future

of anything. Before you start with ML algorithms, it’s important to learn basic probability and statistics(Don’t be afraid of Mathematics- you just need to revise a few concepts you had learned in your school). Then just start with supervised learning as your first step and gradually move towards unsupervised learning and finally a peaceful destination deep learning.
For content, you can refer to https://machinelearningmastery.com/start-here/ (I know it is boring for some for you to read but once you get started it becomes your habit and you will really enjoy it.) And further content and resources will be provided to you later if you are really interested.

3. **Get started with any competition:** Once you have the basic understanding of ML algorithms it’s time to use them in real-life applications and this you can experience with ML competitions. To get started with the competition, I will suggest referring to very good content developed by Kaggle Grandmasters on Coursera and the link is given below.

   How to Win a Data Science Competition: Learn from Top ...www.coursera.org › learn › competitive-data-science

   It will give you an idea about how to approach an ML contest. Once you did this, start participating in ML contests on various platforms like
    1. MachineHack (https://www.machinehack.com)
    2. Analytics Vidhya (https://datahack.analyticsvidhya.com/contest/all/)

I suggest you participate in a contest that is already completed and then refer to the notebooks that are topped in the leaderboard and try to learn from them and implement it in the next contest. Check out their GitHub or

In the Linkedin profile, you must find their notebooks. Even Kaggle notebooks are just perfect to learn new and amazing things and some things are too complicated so beware.

4. **Get started with any competition:** Kaggle would be the best place as there are solution kernels for the problem and try replicating the same. In the start you might not understand everything you implement, then use your googling skills to decode the answer you need and that's how you'll learn.
5. Try every kind of problem - Supervised, NLP, Computer Vision, Unsupervised and try to get a hang of everything.
6. Find a problem you think is solvable using your learned skills, (TRUST ME EVERYTHING is solvable/can be optimized using Machine Learning).
7. Check Analytics Vidhya and MachineHack for more cool competitions as they have no solution kernels until the competition gets over, so that gives you an opportunity to test your skills

Some documentations to refer to for pose detection project :

1. https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html
2. https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html
3. https://www.geeksforgeeks.org/posenet-pose-estimation/
4. https://learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/
5. https://developers.google.com/ml-kit/vision/pose-detection/android#kotlin
6. https://www.tensorflow.org/lite/examples/pose_estimation/overview

Some documentations to refer to for the Sentiment analysis project:

1. https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk
2. https://realpython.com/sentiment-analysis-python/
3. https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/
4. https://techvidvan.com/tutorials/python-sentiment-analysis/

Basic steps to learn frontend Tech:

“ The best way to learn is by implementing things, so instead of just reading these, do try implementing each of them.”

You can also choose to build frontend on flutter for mobile app, this is to begin with HTML CSS, and JS frameworks.

Step 0: Basics

If you've never touched frontend technologies before, you'll have to go learn those before. It's going to take time. But you won't be able to move forward without it. Mozilla Developer Networks is your friend here. ***Stay away from w3schools because it usually just covers the syntax instead of explaining the topics. You should always look for concepts in MDN docs.***

- HTML Basics
- CSS Basics
- Javascript Basics
- HTTP Basics

Step 1: Basics Continued...

Now that you know the basics, you'll continue your journey to learn different day-to-day concepts that you'll need while programming. I encourage you to understand these concepts deeply. You'll always need them as long as you work in the frontend domain.

- HTML / Browser

    - Introduction to the DOM
    - Introduction to events
    - Web Storage API (local storage)
    - History API

- CSS
  This is a topic that is always easy to start with but difficult to have a good understanding of. While working with CSS, there are often hacks/forceful styling which seems to work well for us but for building a good application, we should use CSS in such a way that works well with all the different screen sizes. The given articles are a must-read and will provide you a good understanding of important concepts.
    - How to learn CSS
    - Specificity explained
    - A guide to flexbox

- Javascript
    - Book series - You Don't Know JS
    - Book - Eloquent Javascript
    - Understanding hoisting in Javascript
    - What is function composition
    - Closures and scoping
        - Additional reading: How do js closures work? - StackOverflow
    - Map, reduce and filter

- Networking
    - Client-Server overview

    - XMLHttpRequest
    - Working with JSON

Step 2: Specific Interesting topics

This is a collection of some interesting concepts that are often used. You must know what's an event loop in an asynchronous environment. Similarly, module systems, routing, authentications are all interesting and important concepts. This list is nowhere near exhaustive. But it'll give you a good understanding of where you are. Frontend development has evolved so much. Just learning Javascript, HTML, CSS doesn't cut it anymore.

- What the heck is the event loop anyway?
- Javascript Module systems
- Building a basic javascript router.
- Networking
    - What are web sockets?
    - Session VS token-based authentication

Step 3: The next layer

The topics in this section are about the advanced tech which gets mostly used while building web applications. Before proceeding into these, you should have a good understanding of why we need to use them in the first place - their advantages and disadvantages if any. There are many different frameworks that are being used, you can check them all and decide which one suits you best and start with. The following links are concerned with - react, redux, and sass.

- Tutorial: Intro to react
- React: getting started
- React lifecycle methods
- React router
  - You can also check for anything related to the router: React training: react-router
- SASS basics
- Redux

You should have a clear understanding of how data in these frameworks flows between components and why is it done in such a manner.

Step 4: Behind the scene

Frontend space is vast and there is a lot to consume but it is always important to have an idea of what happens behind the scenes and an understanding of how the browser works.

- What happens when the g-key is pressed?
- How do browsers work?
- Critical rendering path
- The secrets of CSSOM and why you should care!
- Javascript engine: how do they even?