- background?

I have been working in the field of data scientist a couple of years. I have a lot of industry experience in performing data analytics, data modeling, data engineering, natural language processing in production, marketing and advertising area by taking advantage of statistical tools, machine learning, neural networks and other advanced tools.

- what type of analysis did that involve at your last project?

Sure, I have done a couple of meaningful industry projects for several big companies in these years to help them to catch up on machine learning and big data for achieving their organization goal. For example, I just finished a predictive analytics project for Shell Oil company in Houston, Texas. That project was supposed to enhance oil field production and cuts cost by finding optimal well settings and forecasting equipment failures and some other potential problems.

- data engineer or DS, which side are you more on?

I usually do both data engineering and data science work, a lot. But I work more on data science.

- confident in python?

Yes. I have worked with python for many years.

- what's featuring engineering is? Importance?

Feature engineering is an art. It is the process of using domain acknowledge of data or business sense to make new features from existing data. The result of our predictive model can increase a lot by creating simple variable. We can also go ahead and find out more interesting characters inside the data, but these could be starters.

- typically only try one model or many different types of models?

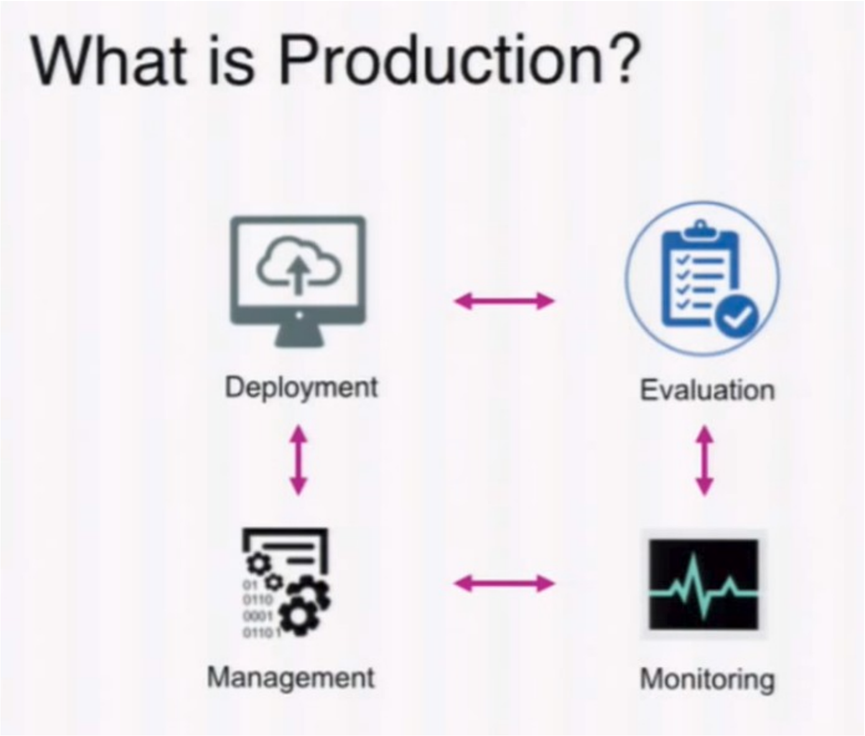
I have built linear models, random forests, decision trees, linear and logistic regression, support vector machine, clustering, neural networks, principle component analysis and I also have a good knowledge on Recommender Systems.

- how do you productionize models?

Productionize basically means sharing, sharing your models and predictions available to everyone, its about measuring, reviewing and improving the quality of these predictions overtime. Its a closed loop including 4 parts:

  - **Deployment**: make predictions, models to everyone.

- **Evaluation**: measure the quality of prediction
- **monitoring**: track the quality over time, collect metrics, feedback
- **management** : improve your already deployed model with feedback



What is Production?

First, we’ll consider what it means to productionize data science code. Data scientists, business analysts, and developers often work on their own laptop or desktop machines during the initial stages of the data science workflow. At some intermediate or later stages in their workflow, they want to encapsulate and deploy a portion or all of their analysis in the form of libraries, applications, dashboards, or API endpoints that other members of the data science team can leverage to further extend, disseminate, or collaborate on their results.

The process of productionizing data science assets can mean different workflows for different roles or organizations, and it depends on the asset that they want to productionize. Getting code ready for production usually involves code cleanup, profiling, optimization, testing, refactoring, and reorganizing the code into modular scripts or libraries that can be reused in other notebooks, models, or applications.

Once the code is ready for production, a number of new considerations must be made to ensure that the deployed projects and applications are robust, performant, reliably accessible, secure, and scalable, among other factors.

Let’s consider an example use case in the deployment stage of the data science workflow. A data scientist has created an end-to-end analysis in an interactive notebook environment that imports and cleans data, then classifies the data using various clustering algorithms. Now, they want to transform this analysis into a report for business analysts that gets updated daily, as well as an interactive web application for all of the users within their organization to interact with.

Another example use case for data science deployments is one that involves a data engineer who wants to transform a notebook used for exploratory analysis into a library that can be used by other data scientists and developers within the organization as well as a web application and REST API for data scientists to export and consume the post-processed, cleaned data in their own analyses. In the following section, we'll explore the details of these data science deployment use cases, describe different types of data science assets, and describe how to deploy and scale data science projects beyond the scope of the original developer's machine.

- what experiance do you have on the data engineering (not feature engineering) end?

I have some experience of building the pipeline to collect the data, clean the data and optimize the pipeline in order to give the result to the users in a faster way. What about on Cloudera and Hortonworks? Aws? Azure? Cloudera?

I have used AWS and Google Cloud Platform these providers before.

- what experiance do you have do CD/CI? Have you done that with Jenkins?

In software engineering, continuous integration (CI) is the practice of merging all developer working copies to a shared mainline several times a day. Grady Booch first proposed the term CI in his 1991 method, although he did not advocate integrating several times a day. Extreme programming (XP) adopted the concept of CI and did advocate integrating more than once per day – perhaps as many as tens of times per day.

Jenkins is an open source automation server written in Java. Jenkins helps to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery.

- any project on DS analytic project that involve something on the financial side? Any experiance with risk analysis?

I have worked in fleetcor, a global company leading in corporation lending and trading. I spent a lot of time on this transaction information happended all over the world. We was supposed to quantify risk of the customer and setting appropriate price for the risk. These transactions happens all over the world, different time zone, and are mainly indexed by the time. And it includes a lot of behavior data of companies like repayment history, transaction style for different financial product.

- how to prevent overfitting of risk analysis? What about in general? are you mostly a hands on person in code, or more of a manager?

perform very well in training data, but poor in testing data
Overfitting occurs when the machine learning algorithm learns a model that fits the training data too well by incorporating details and noise specific to the training data.
The problem of overfitting is usually solved by regularization or early stopping,

- what experience do you have in financial analysis? I see capital one / sun trust can you explain that experience?

I have worked in fleetcor, a global company leading in corporation lending and trading. I spent a lot of time on this transaction information happended all over the world. We was supposed to quantify risk of the customer and setting appropriate price for the risk. These transactions happens all over the world, different time zone, and are mainly indexed by the time. And it includes a lot of behavior data of companies like repayment history, transaction style for different financial product.