> Written with StackEdit.

- Based on your past experience was is the best way to deploy machine models in a production environment.

  Ans： I don't think there is clear-cut answer for this question. Before deploying machine models, I may need to make clear some quesitions: What are the data preprocessing needs? How will I evaluate the model's performance in production? How frequently do I plan on re-training my model? Does the model need to be able to run offline?
  There are some good practice at my previous company:

  - Automated model training (every week or month) so the model's always fresh
  - Saving the actual data we use at prediction time to a db so we can train on that (much easier than trying to write complicated SQL queries to duplicate that exact same logic in a different language)
  - Run a set of tests on each new model that we want to deploy (make sure it runs, make sure it gets a certain error score on a pre-defined dataset, etc.)
  - Super important: change our operations based on analytics from the model. Find cases where the model is struggling, and see if we actually need those as a company, or if we can improve our operations there so the model has something it can learn.
  - Monitor the distribution of values being fed into the model at prediction time, to make sure they're consistent with how we trained the model (both to check for errors in our data extractors, as well as changes in our underlying datasets that might indicate a change in behavior the model wasn't trained to recognize).
  - Monitor customer feedback to find cases we can improve upon.

- What been your experience with Docker? Usecase?

  Ans: I have been using it a lot before. To me, Docker is like a **shipping container system** for applications. For an example, when I want to spin up a Database Server (MySQL) quickly on your laptop. Or setup a Redis Server for your team. Docker makes this dead simple. All I need to do is : *docker run -d -p 3306:3306 tutum/mysql* I often find myself giving a demo or two on the weekends to some group or the other. The software stack for these demos varies big time. I am increasingly finding that Docker images as an ideal way to package and demo

these applications. This way I stick to a consistent method to package and demo my software.

- how do you collaborate with others.

  Ans: I used to collaborate with my teammates on github. The general workflow is:

  1. Fork projects on github
  2. Create a topic branch from `master`.
  3. Make some commits or debug to improve the project.
  4. Push this branch to your GitHub project.
  5. Open a Pull Request on GitHub.
  6. Discuss, and optionally continue committing.
  7. The project owner merges or closes the Pull Request.

- how do you train a model for deployment? Once trained how is that model deployed, same architecture as training or different?

  Ans: It usually depends on if the model need to be able to run offline. Offline learning trains models in a separate environment from which it is trained, and then the model is uploaded to the server. In contrast, online learning trains the model continuously from streaming data in realtime. If your dataset isn't changing over time, choose offline learning. It's much cheaper and more reliable, though you should still monitor your dataset for change.

- whats you deployment experience in spark?

  Ans: I used Spark to perform some predictive analysis on food inspection data that was acquired through the City of Chicago data portal.This dataset contains information about food establishment inspections that were conducted in Chicago, including information about each establishment, the violations found (if any), and the results of the inspection. The general progress was
  1 . Create a Spark MLlib machine learning app
  2 . Construct the input dataframe
  3 . Understand the data
  4 . Create a logistic regression model from the input dataframe

5 . Evaluate the model using another dataset
6 . Create a visual representation of the prediction

- what work have you done with sparkml? How does that framework fall short vs tensorflow \ other frameworks?

  Ans: I used Spark to perform some predictive analysis on food inspection data that was acquired through the City of Chicago data portal.This dataset contains information about food establishment inspections that were conducted in Chicago, including information about each establishment, the violations found (if any), and the results of the inspection. The general progress was
  1 . Create a Spark MLlib machine learning app
  2 . Construct the input dataframe
  3 . Understand the data
  4 . Create a logistic regression model from the input dataframe
  5 . Evaluate the model using another dataset
  6 . Create a visual representation of the prediction

  Difference: Apache Spark is data processing framework and Tensorflow is used for custom Deep Learning /Neutral network design. Apache Spark has been wrapped at a level where it could also be used for ML natively but not as sophisticatedly as TensorFlow.

- what kind of preprocessing do you do on your data before you feed that into your model?

  Ans:

  - Learn the background of data and goal
  - Getting enough data
  - Inspecting the data
  - Separate the features and labels
  - Cleaning null values (missing data)
  - Finding duplicate names
  - Outlier detection
  - Encoding data
  - Data engineering
  - Features selection
  - Feature scaling
  - Splitting the data for training and testing
  - Validation data

    - Model selection
    - Verifying consistent distributions across subsets

- what types of data sources have you used in the past?

  Ans: Companies have moved ahead of traditional data sources by shifting their data on the cloud. Cloud storage accommodates structured and unstructured data and provides business with real-time information and on-demand insights. The main attribute of cloud computing is its flexibility and scalability. As big data can be stored and sourced on public or private clouds, via networks and servers, cloud makes for an efficient and economical data source.

- how comfortable are you learning new tools?

  Ans: I feel comfortable with learning any new tools not only because they are useful, but also I usually like the the feeling of learning new knowledge. Data science is so new that there are always some new tools or frameworks coming up. We need to be ready to learn new things and I am ready.

- what languages are you most comfortable with?

  Ans: I am familiar with Python, Scala, and R