MLOps – NLP – Case Study – Vaibhav Jain

Problem Statement

You are working as a data scientist for a healthtech company called BeHealthy. It is a late-stage startup.

Be Healthy provides a web-based platform for doctors to list their services and manage patient interactions. It provides various services for patients, such as booking interactions with doctors and ordering medicines online. (Similar to 1mg, PharmEasy)

Here, doctors can easily manage appointments, track past medical records and reports and give e-prescriptions.

We can assume that BeHealthy has enrolled thousands of doctors across the major cities in India. You can also assume that millions of patients are using BeHealthy's services; hence, a lot of free-text clinical notes would be present in the e-prescriptions.

BeHealthy would want to convert these free-text clinical notes to structured information for analytics purposes. You have seen such a problem in your previous courses on NLP. You had created a CRF model to recognise the Named Entity in the medical data set.

For example:

Clinical Note: "The patient was a 62-year-old man with squamous cell lung cancer, which was first successfully treated by a combination of radiation therapy and chemotherapy."

Disease: Lung Cancer

Treatment: Radiation Therapy and Chemotherapy

Before solving any business problem, we must always keep in mind why a business needs to resolve the problem. In many cases, data scientists directly jump to the solution, using data sets like Kaggle competitions. But in the real world, you should be able to justify the business need, define KPIs and then design an optimal solution.

Now, let's take a look at a **business goal**:

Currently, if you need to extract diseases and treatments from free text, a trained person with clinical knowledge must manually look at the clinical notes and then pull this information.

A data entry team would manually look at the clinical text and extract information about diseases and their treatment data. A data validation team would validate the extracted information. This process is prone to errors, and as the data increases, the data-entry team's size would need to be scaled up.

Automating this process would result in reduced man-hours. The data-entry team would not be required. The data validation team would still need to perform validation on extracted data. It would also significantly reduce manual errors.

Q1. **System design**: Based on the above information, describe the KPI that the business should track.

KPIs for automated system would be:

1. Eliminate the need of data entry team (trained person with clinical knowledge for data entry), Labeling disease and treatments would be automated.
2. Significantly reduce manual errors – Validation process should go as smooth as possible with minimum errors, so that disease and possible treatment can be identified with higher accuracy.
3. Reduce man hours/ manual work/ cost – ultimate goal to reduce the cost for running a large platform by reduing the manual work.

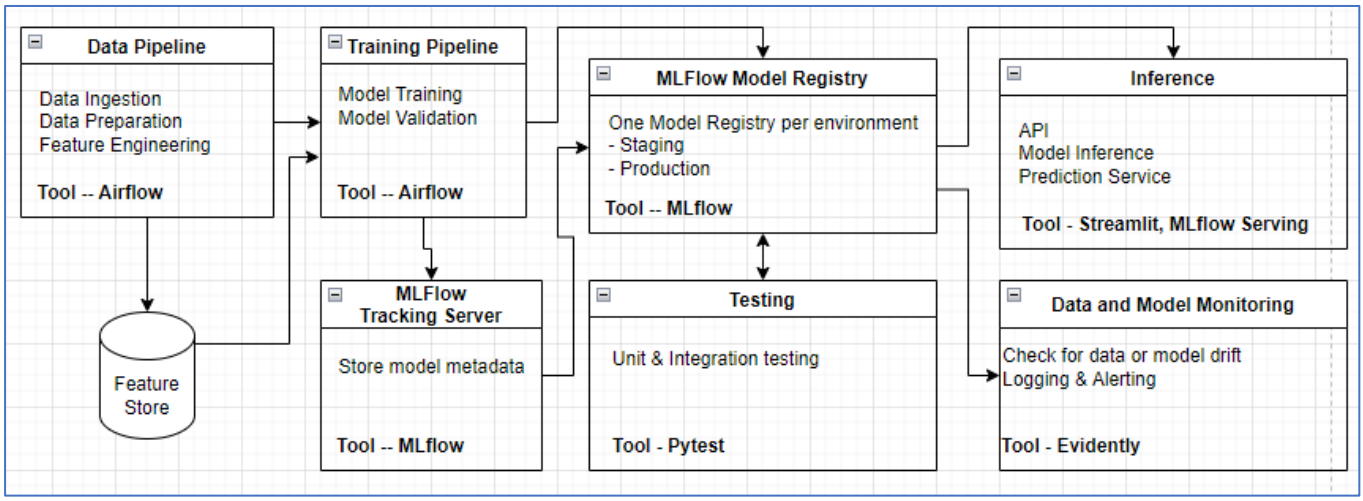
Q2. **System Design**: Your company has decided to build an MLOps system. What advantages would you get by opting to build an MLOps system?

MLOps system provides big advantages, it automates full end to end solution from data processing, building models to inferring outcomes.
Whole team can work with high productivity, they can try and test various models quickly. Model generation can be repeated with great reliability. Every single tuned parameter can be saved.
All experiments can be tracked and referred back to at any given time.

Management can look at things from top level and see whats going on everyone becomes accountable. They can also intervene and provide points to improve particular parameter like accuracy vs precision.

Q3. **System design**: You must create an ML system that has the features of a complete production stack, from experiment tracking to automated model deployment and monitoring. For the given problem, create an ML system design (diagram). The MLOps tools that you want to use are up to your judgement. You can use open-source tools, managed service tools or a hybrid.

Production Stack ML system:



Airflow can ingest data from BeHealthy data source\ can run on any schedule task, and inference can be made available via API endpoints, which can be integrated in BeHealthy aplication or web service.

Q4. **System design**: After creating the architecture, please specify your reasons for choosing the specific tools you chose for the use case.

I have chosen most of open source tools in my system design. Setting all these up is fairly simple and serves basic functionalities in line with tools provided by managed services. Whole system can be integrated with company app or web service via API endpoints.

My system design uses below tools:

1. **Airflow** – This is used to schedule and monitor workflows. Define DAGs and interdependencies \ sequencing of various tasks. Tasks are used for all minor steps like data ingestion, cleaning, preprocessing, feature engineering. And later on to run the actual model training, validation steps.
2. **Mlflow** – This is another tool used to manage machine learning workflow. I have used this to track all the model experiments. With this tool we can centrally manage all the models and complete lifecycle, reproduce experiment results, Store all model parameteres, promote model to staging and production.
3. **Pycaret –** This is python machine learning library, very simple to use, and quickly and efficiently build and deploy machine learning pipelines. It can train 100s of various ml models with single command and we can run various experiments fast and efficiently. With only few lines of code, we can run all the various algorithms\ ml models and know which one can work best on our data.
4. **Pytest –** This is used to run test cases\ test our pipeline code. This make sure we have a working reliable code, and we can detect issues on making any changes to code in future.
5. **Evidently –** This is used to monitor ml models, and analyze we our data or model performance is drifting with new data\ time. Various steps and thresholds can be defined here for specific scenarios to retrain the model or adjust model parameters if performance starts degrading with time.

Q5. **Workflow of the solution**:
You must specify the steps to be taken to build such a system end to end.

The steps should mention the tools used in each component and how they are connected with one another to solve the problem.
Broadly, the workflow should include the following. Be more comprehensive of each step that is involved here.

- Data and model experimentation
- Automation of data pipeline
- Automation of the training pipeline
- Automation of inference pipeline
- Continuous monitoring pipeline

The workflow should ALSO explain the actions to be taken under the following conditions.
After you deployed the model, you noticed that there was a sudden increase in the drift due to a shift in data.

What component/pipeline will be triggered if there is any drift detected? What if the drift detected is beyond an acceptable threshold?
What component/pipeline will be triggered if you have additional annotated data?
How will you ensure the new data you are getting is in the correct format that the inference pipeline takes?

Data and model experimentation – Data exploration, Preprocessing, and model experimentaion would be carried out as part of EDA, & feature engineering by ML Engineer manually in a notebook. These steps will then be coded as individual steps in utils.py file, which will be run by automated tasks pipeline.Various tools used would be python libraries like pandas, sklearn.

Automation of data pipeline – Data can be brought into system on a schedule using service endpoints. Airflow steps\ tasks can run on a schedule and bring data into pipeline. This data can be from BeHealthy app or any webservice or user interface BeHealthy has built for its clients.

Automation of training pileine – Combination of tools like Airflow, Mlflow integrated into MLOps pipelines will carry out the tasks of running preprocessing on data, train the models. A manual step of choosing best model by looking at MLFlow metrices and then promoting the best model to production is required at the end.

Automation of inference pipeline – Once model is trained and available in production, it is made availble via API endpoint to accept input and reply with model inference. This endpoint should be integrated with BeHealthy App or webservice by which clients interact with the platform. Live stream or batch mode – both modes can be implemented to run the model outcome.

Continuous monitoring pipeline – Evidently can monitor both input data and model performance and can detect the drift using various metrices. We can define thresholds to notify ML Ops team, and at the same time trigger retrainig the model on new data.

When we notice there was a sudden increase in the drift due to a shift in data, If a minor drift is detected, model can be retuned using hyperparameters, and a newer model can be promoted to production.

Where as if data drift is beyond a threshold value, whole model needs to be retrained. We can even get a different model with new set of data (for e.g. previously XGBoost might have best accuracy, but now adaBoost might have best accuracy). This new trained model needs to be promoted to production.

For additional annotated data, we can fit the model to new dataset, and retrain it as we get new data daily.

We need to put checks in place in order to detect data format. Strict controls around inference pipelines, via API endpoints so that it does not accept random data.