# MLOps – NLP – Case Study

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

KPI, or Key Performance Indicator, is a measurable value that tracks the performance of a business in achieving its objectives. In the context of the BeHealthy problem statement, the business goal is to automate the process of extracting diseases and treatments from free text, which would result in reduced man-hours, eliminate the need for a data-entry team, reduce manual errors, and scale up the size of the data without increasing the size of the data-entry team.

The KPI that the business should track is the accuracy of the automated process in extracting diseases and treatments from free text. This KPI would measure how well the automated process is performing in correctly identifying the diseases and treatments from the clinical notes compared to the manually extracted information. The accuracy of the automated process should be high enough to ensure that there are no errors in the extracted data, and it should be continually monitored and improved. By tracking this KPI, BeHealthy can ensure that their automated process is meeting the desired business goals of reducing man-hours, eliminating the need for a data-entry team, reducing manual errors, and scaling up the size of the data without increasing the size of the data-entry team.

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

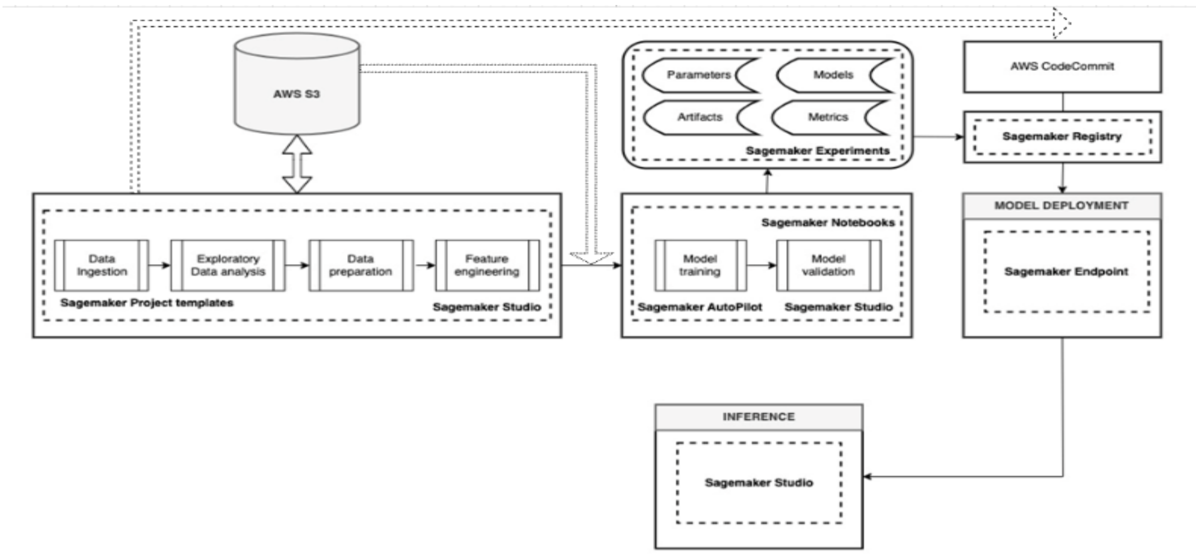
Building an MLOps system, or Machine Learning Operations system, provides several advantages for companies that are developing and deploying machine learning models. Some of the advantages are:

- Improved model performance: MLOps systems ensure that machine learning models are built and deployed efficiently and effectively, resulting in improved model performance.
- Faster model deployment: With MLOps, models can be deployed faster and more efficiently, reducing the time it takes to bring a model from development to production.
- Better model governance and version control: MLOps systems provide robust version control and governance, ensuring that models are developed, deployed, and maintained in a structured and compliant manner.
- Increased collaboration: MLOps systems enable better collaboration between data scientists, developers, and IT operations teams, resulting in improved communication and teamwork.

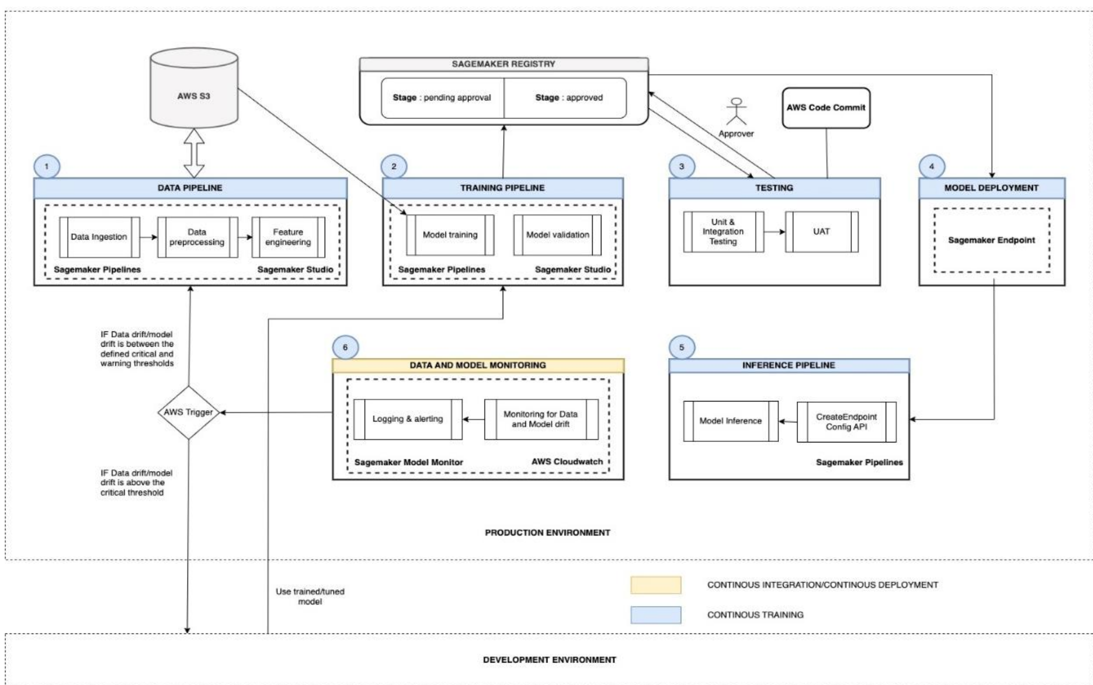
- Improved scalability and flexibility: MLOps systems can easily scale and adapt to the changing needs of a business, allowing companies to quickly and efficiently deploy new models as needed.
- Reduced operational costs: With MLOps, companies can automate many aspects of machine learning model development, deployment, and maintenance, resulting in reduced operational costs.
- Increased model accuracy and reliability: MLOps systems help ensure that machine learning models are developed, deployed, and maintained consistently, resulting in increased model accuracy and reliability. This can help businesses make better decisions based on the insights generated from the models.
- Enhanced data security and compliance: MLOps systems provide robust security and compliance measures, ensuring that data is handled and stored appropriately. This can help businesses mitigate data privacy and security risks.
- Improved customer experience: By developing and deploying machine learning models more efficiently, businesses can deliver better customer experiences, such as faster response times, more accurate predictions, and personalized recommendations.
- Competitive advantage: MLOps systems can help businesses gain a competitive advantage by enabling them to quickly and efficiently develop and deploy machine learning models. This can help businesses stay ahead of the competition and improve their market position.
- Better resource management: MLOps systems can help businesses optimize their resources by automating many aspects of model development and deployment. This can help businesses reduce costs and increase efficiency.

In summary, building an MLOps system can provide businesses with a range of benefits, including increased model accuracy and reliability, enhanced data security and compliance, improved customer experience, competitive advantage, and better resource management.

## 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).




***Development Pipeline for experimentation and Trials***




***Production Pipeline for MLOps system for the***

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

Based on the use case and the fact that the BeHealthy is a late-stage startup, we have chosen managed cloud services (AWS PageMaker) as our primary tool for this system. Here are some of the reasons why:

- Easy deployment and management: Managed cloud services like AWS SageMaker provide easy deployment and management of the entire system. This is especially beneficial for a startup like BeHealthy, as it reduces the need for dedicated DevOps personnel and saves time and resources.
- Cost-effective: Managed cloud services offer a pay-as-you-go pricing model, which can be more cost-effective than traditional on-premises infrastructure. This is important for a startup like BeHealthy, which may have limited resources.
- Scalability: Managed cloud services offer scalability, allowing the system to scale up or down based on the volume of data and user traffic. This ensures that BeHealthy can handle an increase in demand for its services without experiencing downtime or performance issues.
- Availability and reliability: Managed cloud services provide high availability and reliability, ensuring that the system is always up and running. This is essential for a healthcare platform like BeHealthy, which needs to provide uninterrupted services to its users.
- Security: Managed cloud services like AWS SageMaker offer robust security features, including encryption, access control, and compliance with industry standards. This ensures that user data is secure and that BeHealthy meets its regulatory requirements.

Overall, we believe that managed cloud services like AWS SageMaker provide the right combination of ease of deployment, cost-effectiveness, scalability, availability and reliability, and security to meet the needs of BeHealthy's system.

Here are some of the key tools and services offered by AWS SageMaker which can be leveraged by BeHealthy:

- Jupyter Notebook: SageMaker offers Jupyter Notebook, an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. This tool is particularly useful for exploratory data analysis and rapid prototyping of machine learning models.
- Data Wrangler: This tool helps data scientists to clean, transform and aggregate data quickly and easily. It has built-in data cleaning and feature engineering capabilities and can handle a variety of data formats, including CSV, JSON, and Parquet.
- Amazon SageMaker Studio: This is an integrated development environment (IDE) that provides a web-based interface to build, train, and deploy machine learning models. It offers a range of tools and services, including Jupyter Notebook, AWS Glue DataBrew, and built-in machine learning algorithms.
- Built-in algorithms: SageMaker offers a range of built-in machine learning algorithms for common tasks such as classification, regression, and clustering. These algorithms are pre-optimized and can be used with large datasets.

- Custom algorithms: SageMaker also allows users to develop and deploy custom machine learning algorithms using frameworks such as TensorFlow and PyTorch.
- Automatic Model Tuning: This tool automates the process of hyperparameter tuning by running multiple training jobs in parallel, selecting the best-performing model based on the user-specified objective metric.
- Model hosting and deployment: SageMaker makes it easy to deploy trained machine learning models as web services, which can be accessed by other applications through a REST API.

## 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.

1. DATA AND MODEL EXPERIMENTATION
2. AUTOMATION OF DATA PIPELINE
3. AUTOMATION OF THE TRAINING PIPELINE
4. AUTOMATION OF INFERENCE PIPELINE
5. 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?

---

The system architecture chosen for this case study is divided into multiple layers for the different stages of ML system life cycle. The ML system has been designed to considering all steps involved in a software Lifecyle with emphasis on keeping KPI high for entire Lifecyle with all benefits of MLOps.

The system has been divided into 2 environments for separation between development and client facing system:

- **Development environment**: The Development environment has been designed to facilitate the evaluation of various models and determine the optimal solution for the given problem statement and dataset. It offers an ideal platform for swift experimentation with both the data and models at hand.
- **Production environment**: This environment where the best model is deployed after testing and contains pipeline for training, testing, monitoring and inference which fulfills their role to allow system to operate in best manner.

Broadly steps are same as any workflow of MLOPS which have been explained below:

**1. DATA AND MODEL EXPERIMENTATION**

This is done completely in the Development environment which it was designed for. In Development environment Data scientist can collaborate and perform experimentation and trails to find the best model with best features which can be used in future as well in case of any drift.

In the Structure we advised the feature engineering is done in Jupyter notebook given in SageMaker studio which contains template which can be used to speed up the process. Here data scientist does the experimentation and prepare final data which is then used by SageMaker Autopilot to do experimentation across optimized models and algorithms to choose best performing model based on a metric.

All theses are tracked by the SageMaker experiments and trials where it stores all artifacts as well as results, which once approved can be pushed to endpoint from SageMaker model registry for testing purposes using SageMaker endpoint.

Once result is obtained as per threshold model is moved to production.

## 2. CAPABILITIES OF SAGEMAKER PIPELINES

- **Build ML workflows:** Using Python SDK, we can build ML workflows comprising parameters, different steps and data dependencies. We can also orchestrate SageMaker jobs such as the processing job and the training job and can also trigger the execution of these pipelines.

- **Troubleshoot ML workflows:** We can visualize the execution of the pipeline and the status of each step in the pipeline in real time in SageMaker Studio. We can also view additional information about each of the steps in SageMaker Studio.

- Manage models: We can manage different versions of models using the Model Registry. We also have the capability to approve/reject models in the model registry. The model registry consists of different model packages, and each model package consists of multiple versions of the model.

- **Scaling MLOps**: We can create a project in SageMaker Studio and get a code repository, seed code and the MLOps infrastructure set up for we. We are provided with MLOps templates published by SageMaker for building, deploying and establishing end-to-end workflows.

- **Track lineage:** With in-built lineage tracking for SageMaker pipelines, we can track data, models and artefacts. Also, support is provided for tracking custom entities.

## 3. AUTOMATED DATA AND TRAINING PIPELINE

This component focuses on automating model training by converting the code developed in notebooks to Python scripts. With automation coupled with using a feature store, the data and training pipelines can run whenever there is any change in the live data. This helps in the continuous delivery of existing deployed models after it is re-trained on the newly transformed data stored in the feature store. The tool used for automation in this stage is SageMaker Studio and SageMaker Pipelines.

## 4. TESTING

In this step, we will test the different methods used in data preparation, feature extraction and model validation to effectively track whether all the components are working in the desired manner. The tests applied here are unit tests, integration tests, and user acceptance testing (UAT). If the model passes all these tests, it can be moved to production, that is, it can be used for making inferences/predictions. Therefore, testing helps in the continuous integration of models trained on new data.

## 5. INFERENCE PIPELINE

In this stage, once the model/code passes all the tests, we will go ahead and deploy the model for serving predictions. The tool used in this stage is SageMaker Endpoint for deployment and providing end-service.

## 6. DATA AND MODEL MONITORING

Keeping a continuous check on the deployed model is essential for tracking the model performance and ensuring that the model doesn't go stale. It signals what action needs to be performed based on any changes in the live data. The 'trigger' connected to this component decides what action to take: model experimentation or model retraining. The tool used in this stage for monitoring any data drifts is called SageMaker Model Monitor and AWS CloudWatch. The Amazon SageMaker Model Monitor continuously monitors the quality of Amazon SageMaker ML models in production. The Model Monitor provides the following types of monitoring:

- Monitor data quality: Monitor drift in data quality
- Monitor model quality: Monitor drift in model quality metric, such as accuracy
- Monitor bias drift for models in production: Monitor bias in your model's predictions

We can set alerts using AWS CloudWatch when in the case of deviations in the model quality.

- **What component/pipeline will be triggered if there is any drift detected? What if the drift detected is beyond an acceptable threshold?**
  If the drift falls within the warning and critical thresholds, it suggests that the inference data differs from the training data model and may result in inaccurate outcomes. Therefore, the model should be retrained using new data to optimize its parameters.
  However, if the data drift exceeds the critical threshold, model retraining becomes ineffective, and the best approach is to revisit the development environment, conduct experimentation to determine the optimal ML model, subject it to UAT, and finally deploy it for inference purposes.
- **What component/pipeline will be triggered if you have additional annotated data?**
  To ensure the new patterns in the additional data are captured, the entire pipeline starting from the data ingestion stage must be triggered. This involves ingesting the new data into the pipeline, followed by training, validating, testing, and deploying the model again. By doing so, the model will be updated to incorporate the new patterns in the additional data, resulting in more accurate outcomes.
- **How will you ensure the new data you are getting is in the correct format that the inference pipeline takes?**
  In order to prepare the new or test data for accurate predictions by the model, it needs to undergo the data pipeline, which involves preprocessing and formatting the data to match the requirements of the model. As a result, the data pipeline is typically triggered on the test

dataset. Once the data has been processed, the inference pipeline can be used to generate predictions on the processed test dataset or dataframe.