

# Designing and Creating Testing Datasets for Medical AI/ML End to End Testing

Nancy McCormack

Mayo Clinic, CDH AI Engineering

[McCormack.Weir@Mayo.edu](mailto:McCormack.Weir@Mayo.edu) (working), or [nhou\\_wei@yahoo.com](mailto:nhou_wei@yahoo.com) (personal)

## Abstract

Testing datasets are essential for ensuring that models work correctly in the real production environment. They can be used to test a wide range of functionality, including API, application, and data pipelines. End-to-end testing is particularly important, as it tests the entire model from start to finish.

Designing testing datasets can be challenging, as it requires a deep understanding of the model and the production environment. It is also important to consider the protection of PHI (Protected Health Information) when designing testing datasets.

This paper describes the following items:

- What the Mayo Clinic CDH AI Engineering team does
- What the AI Engineering Verification and Validation Engineers do
- What end-to-end testing is (based on what the Mayo Clinic CDH AI Engineering team does)
- How to design testing datasets with the protection of PHI (Protected Health Information) for end-to-end testing based on a real Mayo Clinic AI/ML project.
- How to create testing datasets for end-to-end testing
- Issues and challenges during the creation of the testing dataset

## Biography

I am a Sr. Full Stack Verification and Validation Engineer for the Mayo Clinic CDH AI Engineering organization, who takes responsibility for verifying and validating models in real production environments to ensure they are safe and effective by collaborating with project team members and cross-function teams. I also lead an offshore V&V team that creates automation frameworks and builds internal tools to help improve the AI Engineering team's productivity.

Before joining Mayo Clinic, I had 8 years of functional management experience and 14 years of hands-on testing engineering experience in the Semiconductor, Networking, and IT industries. Accomplished at managing and leading diverse quality teams using both waterfall and agile methodologies, I am adept in all aspects of the SDLC but excel in test strategy, planning, process, standards, and metrics.

During my free time, I enjoy traveling, watching sports, and tasting different types of foods with my husband, Simon McCormack. I also enjoy dancing and watching real mystery shows.

# 1. Introduction

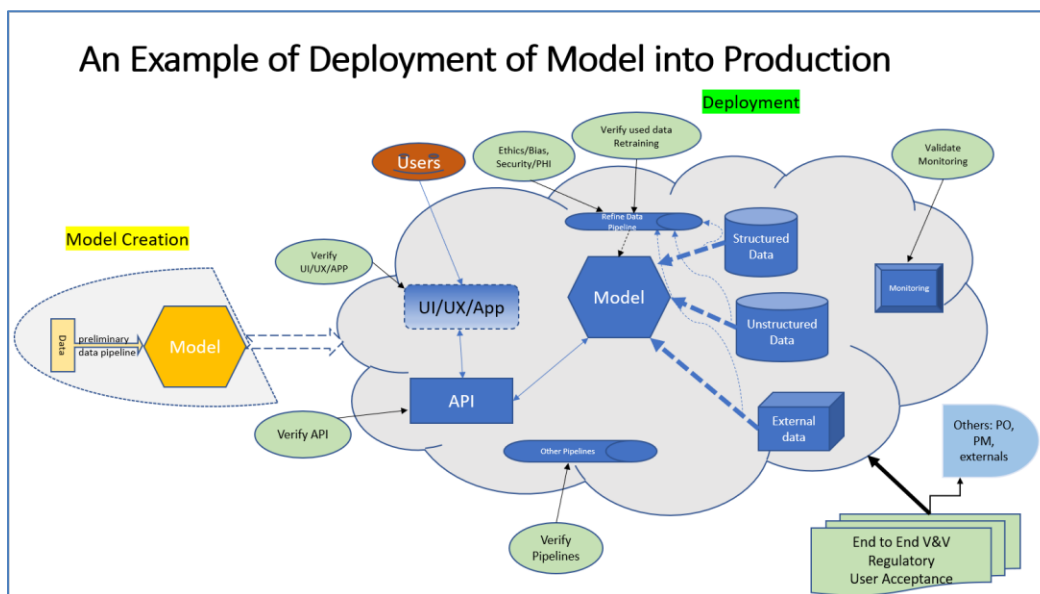
Mayo Clinic has been ranked the No. 1 hospital overall in the nation and top-ranked in fourteen specialties by U.S News and World Report for seven consecutive years.

The Center for Digital Health (CDH) creates innovative digital experiences, aligned with Mayo Clinic's mission to improve the health and well-being of people worldwide, CDH collaborates with partners to establish innovative ways to yield greater value, agility, and effectiveness.

CDH AI Engineering team takes responsibility for deploying models into real production environments. Those models can dramatically reduce the time spent by clinic users who are reviewing patients' history data and information. They also help clinic users make predictions and plan for patients more efficiently and properly. The CDH AI Engineering team is making a significant contribution to improving the quality of care at Mayo Clinic. Our work is helping to save lives and improve the lives of patients.

As a Verification and Validation Engineer on the AI Engineering team at Mayo Clinic, I work closely with MLOps engineers, data scientists, data engineers, IT engineers, and medical informatics specialists to ensure that the models that are deployed in production are safe and effective. I work across the entire lifecycle of a model, from development to deployment to monitoring, to ensure that the model meets the needs of the users and the requirements of the production environment. My work includes developing and implementing verification and validation plans, collaborating with stakeholders to understand the requirements, verifying and validating the API, UI/UX, applications, pipelines, bias, compliance, and other items that support model running in production, and focusing on end-to-end testing, monitoring, reporting, and retraining.

## 1.1 What exactly do AI Engineering and V&V do for deploying models into production environments?



**A Basic Model Deployment**

## Model Creation

When data scientists build models, they take care of data validation with data engineers, model creation, training, and testing to finalize a model that is ready for deployment, and then pass it to our team, the AI Engineering team.

## Model Deployment

To deploy the model into the real production environment, our MLOps Engineers build the following items:

- Data pipelines/API: get proper data from different databases including clinic data from the clinic database, clinic notes, documents, and lab results from the Clinic document database, and external data from external databases.
- Other pipelines: Integration pipeline, deployment pipeline, CI/CD pipeline that are necessary for deployment work.
- UI/UX applications: The end clinic users can use the final product on a web browser, and they are normally built by UI engineer with UX designer.

## Verification and Validation

V&V Engineers start with simple API and UI testing and finish with end-to-end testing and monitoring.

Here are some examples:

API Testing	UI Testing	End to End (Full Stack) V&V
Functional	Functional	Data Flow
Data formatting and defaults (based on data cards)	Usability	Data bias
API Performance	PHI	Use cases
Negative/Error handling	Security	Multi-users
PHI	UI/UX standards	End to end performance
Scalability	Basic UI Performance	Differential
Stability	Negative/error handling/messages	Compliance
Simple Integration	Testable and automatable	Model retaining and monitoring

Monitoring, reporting, and retraining will continue in production after we are done with deployment work.

## 1.2 How to design verification and validation testing datasets for end-to-end testing?

Those are some important items to consider when designing testing datasets for Verification and Validation (V&V) testing, especially for end-to-end testing.

- How the model works: This includes understanding the model's input data, its output data, and how the model makes predictions.
- Data that the model needs: This includes the type of data, the format of the data, and the amount of data that the model needs to make accurate predictions.
- Data range: This includes the minimum and maximum values of the data, as well as the distribution of the data.
- Data bias: This includes understanding what data can cause model bias to help with creating data that can reduce and remove those bias.
- The relationship between all data: This includes understanding how the different pieces of data are related to each other, and how they affect the model's predictions.
- How MLOps Engineers make the model live in production: This includes understanding the production environment, the infrastructure that the model will run on, and the monitoring tools that will be used to track the model's performance.
- Data source: This includes understanding where the data comes from, how it is collected, and how it is cleaned and prepared for use in the model.
- Data flow: This includes understanding how the data flows through the system, from the data source to the model to the production environment.

By considering these items, you can design testing datasets that will help to ensure that the model is working correctly and that it is meeting the needs of the users.

Verification and Validation (V&V) testing datasets need to be designed to cover all necessary combinations of data points and different data flows. This is to ensure that the model can handle all possible scenarios and that it is not biased toward any particular set of data.

Here are a couple of specific examples of how V&V testing datasets can be designed to cover all combinations of data points and different data flows:

- The testing datasets can be designed to include all possible combinations of input data. For example, if the model is a medical diagnosis model, the testing datasets could include all possible combinations of symptoms, test results, and patient demographics.
- The testing datasets can be designed to include different data flows. For example, the testing datasets could include data that is collected from different sources, such as electronic health records, patient surveys, and laboratory tests.

For example, one of projects that I worked on is to develop a patient-focused and physician-driven diagnostic model and visualization framework for Acute Leukemia or MDS (Myelodysplastic Syndromes) patients.

Myelodysplastic Syndromes (MDS) are a group of diverse bone marrow disorders in which the bone marrow does not produce enough healthy blood cells. MDS is often referred to as a “bone marrow failure disorder”. MDS is primarily a disease of the elderly (most patients are older than age 65), but MDS can affect younger patients as well.

The model uses structured data: the patient's most recent bone marrow test data, cytogenetic test data, and genomic test data to calculate IPSS-M and IPSS-R risk scores and levels. These risk scores and levels can be used to help physicians make decisions about treatment options for patients with MDS.

Clinic users can also modify the data based on unstructured data: clinic documents and CBC (Complete Blood Count) test results. This allows for a more comprehensive assessment of the patient's condition and can help to improve the accuracy of the risk scores and levels.

The data, clinic documents, and CBC test results can be internal or external.

*Note 1: The model is an open-source model and is used in the Medicare industry popularly.*

*Note 2: The model is used for adult patients, who are older than 17 years old.*

*Note 3: Risk score and levels are not impacted by gender.*

**IPSS-M (Molecular International Prognostic Scoring System) Levels <-> Scores:**

IPSS-M Level	IPSS-M Score
VL	$\leq -1.5$
L	$>-1.5$ to $-0.5$
ML	$>-0.5$ to $0$
MH	$>0$ to $0.5$
H	$>0.5$ to $1.5$
VH	$>1.5$

**IPSS-R (Revised International Prognostic System) Levels <-> Scores:**

IPSS-R Level	IPSS-R Score
Very Low	$\leq 1.5$
Low	$>1.5$ to $3$
Intermediate	$>3$ to $4.5$
High	$>4.5$ to $6$
Very High	$>6$

**Structure data is used for calculating IPSS-M & IPSS-R scores and levels:**

Data type	Data	Data values	IPSS-M R(Required)/ O(Optional)	IPSS-R R(Required)/ O(Optional)
Bone Marrow	Bone Marrow Blasts	0-30%	R	R
	Hemoglobin	4-20 g/dL	R	R
	Platelet count	0-2000 $1e9/L$	R	R

	Absolute Neutrophil Count	0-15 le9/L	O	O
	Age	18 to 120 years	O	O
Cytogenetics	Presence of del(5q)	Yes/No	R	N/A
	-7/del(7q)	Yes/No	R	N/A
	-17/del(17p)	Yes/No	R	N/A
	Complex Karyotype	Yes/No	R	N/A
	Cytogenetics Category	Very good, Good, Intermediate, Poor, Very Poor	R	R
Genomic	Number of TP53 mutations	0, 1, 2+	R	N/A
	Loss of heterozygosity at TP53 locus	No, Yes, N/A	R	N/A
	FLT3 Mutations includes MLL PTD	No, Yes, Not Assessed	R	N/A
	FLT3s ITD or TKD	No, Yes, Not Assessed	R	N/A
	13 Genes (Individual weights)	Non-mutated, Mutated, Not Assessed	R	N/A
	15 Genes (number of residual mutations)	Non-mutated, Mutated, Not Assessed	R	N/A

### Examples of Testing Datasets

The testing datasets need to be comprehensive enough to test all different risk scores and levels, and different data flows. This means that the datasets need to include patients with all different types of data, including patients with complete data and patients with incomplete data.

Including patients with incomplete data is important because it allows the project to test how the model handles imperfect data. This is important because in the real world, patients will not always have complete data. The project needs to be able to handle imperfect data correctly in order to be accurate and reliable.

Here are some specific examples of how the testing datasets could be used to test different risk scores and levels, and different data flows:

- The datasets could be used to test how the model calculates risk scores for patients with different types of data. For example, the datasets could be used to test how the model calculates risk scores for patients with complete data, patients with incomplete data, and patients with no data.
- The datasets could be used to test how the model handles different data flows. For example, the datasets could be used to test how the model handles data that is entered in different ways, data that is entered in different formats, and data that is entered with errors.

- The datasets could be used to test risk level categorization using the risk scores, especially using the end points of each score ranges.
- The datasets could be used to test bias by using different ages and genders that impact risk scores and levels and possible races, locations, which should not impact risk scores and levels)
- The datasets could be used to test data pipelines by having data from different data sources including internal/external structured and unstructured data.

By including patients with all different types of data, the testing datasets can be used to test the model's accuracy and reliability in a variety of situations. This will help to ensure that the model is able to provide accurate and reliable risk scores and levels for patients with MDS.

Some examples of testing datasets not including entries intentionally missing some data or no data that is a part of testing datasets.

Age	IPSS-R Risk Level   Score	IPSS-M Risk Level   Score	BM% Hemoglobin (g/dL)	Platelet	ANC (1e9/L)	del(5q)	del(7q)	del(17p)	Complex Karyotype	Cyto Category	TP53 Mutation Count	Max TP53 VAF	TP53 locus LOH	Mutated Genes	Missing Genes	Most Recent BoneMarrow Test	Most Recent Cytogenetic Test	Most Recent Genomic Test	CBC Result
47	Intermediate   4	Very High   1.87	11%	12	222	5	false	false	false	Good	2 or more	4%	true	TP53, TP53loh, MLL, PTD	None	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None
18	High   6	Very High   4.26	0%	4	0	0	false	True	True	Poor	2 or more	4%	true	TP53, TP53loh, MLL, PTD	None	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None
24	Intermediate   4	Moderate Low   -0.30%	13	1001	9.8		false	false	false	Good	1	23%	false	TP53	None	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None
51	Low   3	High   1.22	30%	12	222	5	false	false	false	Very Good	2 or more	80%	true	TP53, TP53loh	None	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None
30	Low   3	N/A   -0.86	16%	12	1000	6	false	false	false	Very Good	1	2%	N/A	TP53loh	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None	
72	Very High   7	Very High   7.1	15%	12	1000	6	false	false	true	Very poor	1	31%	N/A	TP53, BCCOR1L, CEBPA, GATA2, IDH1, NFL, PHF6, PPM1D, PTPN11, STAG2	TP53loh	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None
120	Very Low   -3.23	Very Low   -3.23	3	18.2	1073	8.4	false	false	true	Very Good	0	0	false	NRAS	None	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None
17	Intermediate   4	Moderate Low   -0.30%	13	1001	9.8		false	false	false	Good	1	23%	false	TP53	None	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None
121	High   6	Very High   4.26	0%	4	0	0	false	True	True	Poor	2 or more	4%	true	TP53, TP53loh, MLL, PTD	None	Internal/External/None	Internal/External/None	Internal/External/None	Internal/External/None

## Tools that help with designing testing datasets

- **Public Calculators**

For this project, I used public IPSS-M and IPSS-R calculators to help me design testing datasets that use different data combinations to generate different risk scores and levels.

This was a time-consuming process, as I had to manually figure out the data combinations based on risk level and score. However, the effort was worth it, as it allowed me to create a comprehensive testing dataset that will help to ensure the accuracy and reliability of the model.

- **ChatGPT**

Two methods are used to generate testing datasets using ChatGPT:

- **Generated Knowledge Prompting**

We ask to create an artificial data set with corresponding IPSS-M scores and then IPSS-R scores by using the following example of prompts sequence with not filling any real patient data:

- Do you know what an IPSS-M score is? • Return: Factors, description of necessary data
- Can you create a data set containing these factors that correspond with varying IPSS-M scores from very low to very high? • Return: Factors are then compiled into a data set with corresponding IPSS-M score.
- Can you then calculate IPSS-M scores given relevant data? • Return: required data needed to calculate.
- Finally, create data for those values so that the corresponding IPSS-M scores cover the range of possible values. • Return: completed data table.

Findings: ChatGPT has limited knowledge of IPSS-M and R scores, and more focus on bone marrow test data.

Issue: scores sometimes are not right. For example: IPSS-M score should be -0.38. but get 1 from ChatGPT. It probably due to aggressive rounding or a simplified score calculation method.

Here is an example of a data set that ChatGPT created:

Patient	Blasts (%)	Cytogenetic Abnormalities	Hemoglobin (g/dL)	Platelet Count (x10 <sup>9</sup> /L)	Absolute Neutrophil Count (x10 <sup>9</sup> /L)	del(5q)	del(7q)	del(17p)	Complex Karyotype	Cytogenetic Category	Mutated Genes	Missing Gene Variants	IPSS-M Qualification	IPSS-R Qualification	IPSS-R Score	IPSS-M Score
1	18	2	8	120	0.6	TRUE	FALSE	FALSE	TRUE	Intermediate	TP53, FLT3	IKZF1	H	Intermediate	2	0.5
2	7	1	15	500	1.2	FALSE	TRUE	TRUE	FALSE	Very Poor	NRAS, JAK2	-	ML	Poor	3	1
3	2	0	4	800	0.2	TRUE	TRUE	FALSE	TRUE	Very Poor	CEBPA, FLT3	-	VH	Very Poor	4	0.5
4	12	0	18	300	0.8	FALSE	FALSE	TRUE	FALSE	Very Good	-	NPM1, DNMT3A	ML	Good	0	-1
5	4	1	11	1000	0.4	TRUE	FALSE	FALSE	TRUE	Good	-	ASXL1	H	Good	0	-1
6	30	2	5	1800	0.7	FALSE	TRUE	TRUE	TRUE	Very Poor	TP53, FLT3	RUNX1	VH	Very Poor	3	1.5
7	15	0	9	1500	1.5	TRUE	TRUE	FALSE	TRUE	Intermediate	-	ETV6, TET2	H	Intermediate	2	0.5

o Few-Shot Prompting

We ask ChatGPT to replicate existing patterns in pre-existing data that is fake data. We label the original data set as a validated pattern. The pattern is then analyzed by the model and replicated first by generating random data for each column (within given ranges) and second by calculating IPSS-M and IPSS-R risk score for the created data.

Could you analyze the original data and then create random data that adheres to that pattern?

Return: Data with corresponding IPSS-M scores.



Findings: We need to ask to generate randomized data to avoid the exact same dataset as the original.

Issues: Risk level data is always off by one full qualification step. For examples, Low should be Moderately Low and High should be Very High. It might because data is generated only based on 10 original datasets. Believe that the larger number of datasets in the original datasets will increase the accuracy.

Here is an example of the original dataset:

IPSS-R Score	IPSS-M Score	BMB	Hemoglobin (g/dL) (date)	Platelet (date)	ANC [1e9/L] (date)	del(5q)	del(7q)	del(17p)	Complex Karyotype	Cyto Category	TP53 Mutation Count	Max TP53 VAF	TP53 locus LOH	Mutated Genes	Missing Genes
Intermediate   4	Very High   1.87	11%	12 31-Mar-2023	222 31-Mar-2023	5 31-Mar-2023	false	false	false	false	Good	2 or more	4%	true	TP53, TP53loh, MLL_PTD	None
High   6	Very High   4.26	0%	4	0	0	false	True	True	false	Poor	2 or more	4%	true	TP53, TP53loh, MLL_PTD	None
Intermediate   4	Moderate Low   -0.41	30%	13	1001	9.8	false	false	false	false	Good	1	23%	false	TP53	None
Low   3	High   1.22	30%	12 31-Mar-2023	222 31-Mar-2023	5 31-Mar-2023	false	false	false	false	Very Good	2 or more	80%	true	TP53, TP53loh	None
Low   3	N/A   -0.86	16%	12	1000	6	false	false	false	false	Very Good	1	2%	N/A	TP53	TP53loh
Very High   7	Very High   7.1	15%	12	1000	6	false	false	true	false	Very poor	1	31%	N/A	TP53, FLT3, CBL, DNMT3A, EZH2, KRAS, NPM1, NRAS, SF3B1, U2AF1, BCR, BCORL1, CEBPA, GATA2, IDH1, NF1, PHF6, PPM1D, PTPN11, STAG2	TP53loh
Very Low   -3.23	Very Low   -3.23	3	18.2	1073	8.4	false	false	true	false	Very Good	0	0	false	NRAS	None
Intermediate   4	Moderate Low   -0.41	30%	13	1001	9.8	false	false	false	false	Good	1	23%	false	TP53	None
High   6	Very High   4.26	0%	4	0	0	false	True	True	false	Poor	2 or more	4%	true	TP53, TP53loh, MLL_PTD	None

An example of a generated data set:

Patient	Bone Marrow Blast (%)	Hemoglobin (g/dL)	Platelet Count (nanoliters)	ANC	del(5q)	del(7q)	del(17p)	Complex Karyotype	Cyto Category	TP53 Mutation Count	Max TP53 VAF	TP53 locus LOH	Mutated Genes	Missing Genes	IPSS-M Qualification	IPSS-R Score
1	19%	14	1900	3	FALSE	TRUE	FALSE	FALSE	Good	>=2	6%	TRUE	TP53, TP53loh, RUNX1	None	H	High Risk
2	4%	10	1500	0	FALSE	FALSE	FALSE	FALSE	Poor	0	0%	FALSE	None	None	VL	Very Low Risk
3	27%	12	1000	6.8	FALSE	FALSE	FALSE	FALSE	Good	1	15%	FALSE	TP53	None	L	Low Risk
4	22%	11	1800	7	FALSE	FALSE	FALSE	FALSE	Very Good	2 or more	85%	TRUE	TP53, TP53loh, MLL_PTD	None	ML	Intermediate Risk
5	13%	12.5	1200	6.5	FALSE	FALSE	TRUE	FALSE	Very Good	1	5%N/A	N/A	FLT3, CBL, DNMT3A, EZH2, KRAS, NPM1, TP53	TP53loh	ML	Intermediate Risk
6	16%	13	1000	5.5	FALSE	FALSE	FALSE	FALSE	Very Poor	1	25%N/A	N/A	TP53, FLT3, CBL, DNMT3A, KRAS, NPM1, RUNX1	TP53loh	H	High Risk
7	2%	16	900	9.2	FALSE	TRUE	TRUE	FALSE	Very Good	0	0	FALSE	NRAS, TP53	None	ML	Intermediate Risk
8	28%	14	1100	8.5	FALSE	FALSE	FALSE	FALSE	Good	1	18%	FALSE	TP53	None	L	Low Risk

Overall, Generated Knowledge Prompting: creating model generated context information improves prompt efficiency and Few-Shot Prompting: Providing expected output improves output format accuracy. The combination of those two methods can lead to better testing datasets.

ChatGPT can help to create testing datasets much quicker. However, due to ChatGPT's limited knowledge and the fact that it does not have the latest data, we still need to use the public calculator to verify and confirm the datasets. Hopefully, when ChatGPT has the latest data, the accuracy of creating datasets can be improved.

### 1.3 How to create testing datasets?

There are different ways to create testing datasets with advantages and disadvantages.

#### Creating fake data in a local database

Advantages:

- Does not require dependency on other teams.
- Can be easily modified when needed.
- Can be used to design and create common data sets for future similar projects.

Disadvantages:

- Takes a significant amount of time to understand the format and content of clinic notes, documents, medications, and lab results in order to create realistic fake data.
- The data flows in a local database may not be the same as the data flows in the real production environment, which can impact end-to-end testing.

Overall, creating fake data in the local database can be a quick way to create testing datasets to start V&V, but the disadvantages of creating fake data in a local database should be carefully considered before using this approach.

#### Creating test patients in Mayo databases

To protect patient health information (PHI), using real patients in testing and automation is not allowed that is important for compliance with HIPAA and other regulations, so we need test patients based on the testing dataset we design and to meet verification and validation purposes.

Test patients are fake patients with fake data but created in Mayo real testing systems/databases. It required to follow the specific process and work with multiple teams.

For special reasons and special cases, test patients can be created in production with going through much more difficult approval process.

Creating test patients while following the Mayo process in Mayo databases is important.

Advantages:

- Protect PHI.
- Use real data flow.
- Get accurate data created by experts that includes clinic documents, medications, and lab results.

Disadvantages:

- Requiring the cooperation of other teams.
- May be difficult to get priority for test patient creation from other teams.

Overall, creating test patients in Mayo Databases can be a valuable tool for testing models. However, it is important to be aware of the advantages and disadvantages of this approach before using it.

## 1.4 Challenges I had during test patient creation

### Finding test patients

Before creating test patients, we always need to find out if there are any existing test patients in systems to avoid unnecessary work for all teams, but finding test patients with characteristics that align with the functionality under test can be challenging.

The challenges are:

- Creating queries to find data is challenging for those unfamiliar with data structures and without a medical background.
- Finding test patients takes time due to the large size of our database and the time it takes to run each query.
- Finding common test patients that have all mandatory data is time-consuming since some of them may not have all of the mandatory data.

Overall, finding test patients can be a challenging task, especially in large databases. However, there are a few things that can be done to make the process easier. These include:

- Working with a data engineer to create efficient queries.
- Indexing the database to improve the performance of the queries.
- Using tools that can help to search through the database more efficiently.

By taking these steps, it is possible to make the process of finding test patients more efficient and less time-consuming.

### Creating test patients

During test patient creation, there are some challenges too:

- Incomplete understanding of the data used by the model can lead to missing data for test patients.
- Limited knowledge of multiple data sources can slow down the process of test patient creation.
- Inconsistency in lab test results from multiple CIS can make test patient creation challenging.

Overall, creating test patients can be a challenging task. However, there are a few things that can be done to make the process easier. These include:

- Use a data card to document the data that is used by the model. This will help ensure that the test patient creation tool has access to all of the necessary data.
- Learn from database experts in the organization to understand the data schema. This will help you work with the right resources to create realistic and accurate patient data.
- Develop a data mapping process to translate data from one format to another. This will help ensure that the data from the different data sources can be used by the test patient creation tool.

Note: All test patients created based on testing datasets are stored and managed by Mayo Epic testing systems and environments. To modify test patients for good reasons, we will have to submit requests to work with multiple teams to get requests completed.

## 2. Conclusion

Our end-to-end testing can ensure that the model is deployed into production correctly, can help to ensure that the model works correctly in the real world, as it is exposed to different data and different environments, and help to ensure that the model is secure and compliant with regulations. So, the proper testing datasets are the key to end-to-end testing.

To design the correct testing datasets, we need to define the scope of the testing, need to understand how the model works, how the model lives in production environments, what data the model uses, the relationship between data, how data flows, and how MLOps engineers deploy the model into production environments. Bias, security, regulations, retraining, and monitoring must be considered as well.

Use a variety of data sources to create the testing datasets. This will help to ensure that the model is not overfitting to any particular dataset.

Use a variety of data ranges to create the testing datasets. This will help to ensure that the model is not biased towards any particular range of values.

Use a variety of data relationships to create the testing datasets. This will help to ensure that the model is able to handle different types of relationships between the data.

Use a variety of data flows to create the testing datasets. This will help to ensure that the model is able to handle different types of data flows.

Choosing the proper way to build testing datasets is based on different types of tests. Building testing datasets correctly from the beginning of the project with appropriate resources would improve the effectiveness of the testing process and the quality of the project.

Necessary help from medical experts is needed and collaboration is critical to the successful creation of testing datasets.

### 3. References

- IPSS-M Risk Calculator: <https://mds-risk-model.com/>
- IPSS-R Basic Calculator: <https://www.mds-foundation.org/ipss-r-calculator/>
- ChatGPT Model: GPT-4 <https://openai.com/research/gpt-4>
- GPT-4 Technical Report:  
<chrome-extension://efaidnbmnnnibpcajpcgqlclefindmkaj/https://cdn.openai.com/papers/gpt-4.pdf>

### 4. Acknowledgements

- John Skiffington** Mayo Clinic CDH AI Engineering Director
- Eric Will** Mayo Clinic AI/ML Software Architect and project technical lead
- Max Comfere** Mayo Clinic AI Engineering Summer Intern (ChatGPT researcher)