



PhUSE Presentation 2023

Intro to ML Models and What to Do When They Become Sassy

www.geninvo.com
North America | UK | Europe | India



Agenda

- 1) Introduction
- 2) What is a Model?
- 3) Examples of ML in Life Sciences
- 4) ML vs. DL Models
- 5) Building/Training a Model
(example)
- 6) Assessing Model Effectiveness
- 7) Fixing a Poorly Behaving Model
- 8) Future of ML in Life Sciences
- 9) Conclusion
- 10) References / Thank-You's

An abstract pattern of white lines and dots on a dark teal background, resembling a circuit board or a network diagram, located on the left side of the slide.

Introduction

An abstract pattern of white lines and dots on a dark teal background, resembling a circuit board or a network diagram, located on the right side of the slide.

What is a Model?



Fig. 1: Image search for "zoolander"; www.themovethemesong.com, Accessed 20 Feb. 2023.

What is a Model?

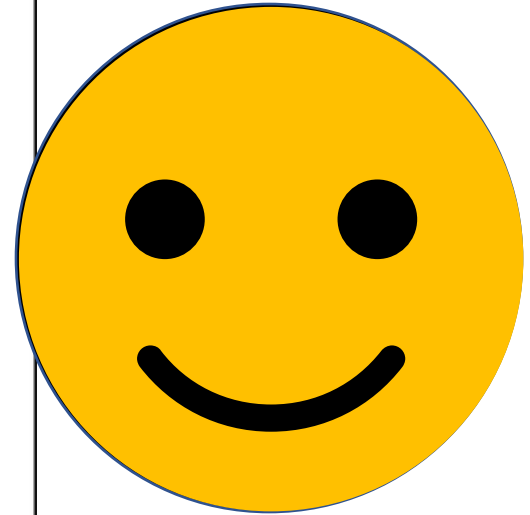
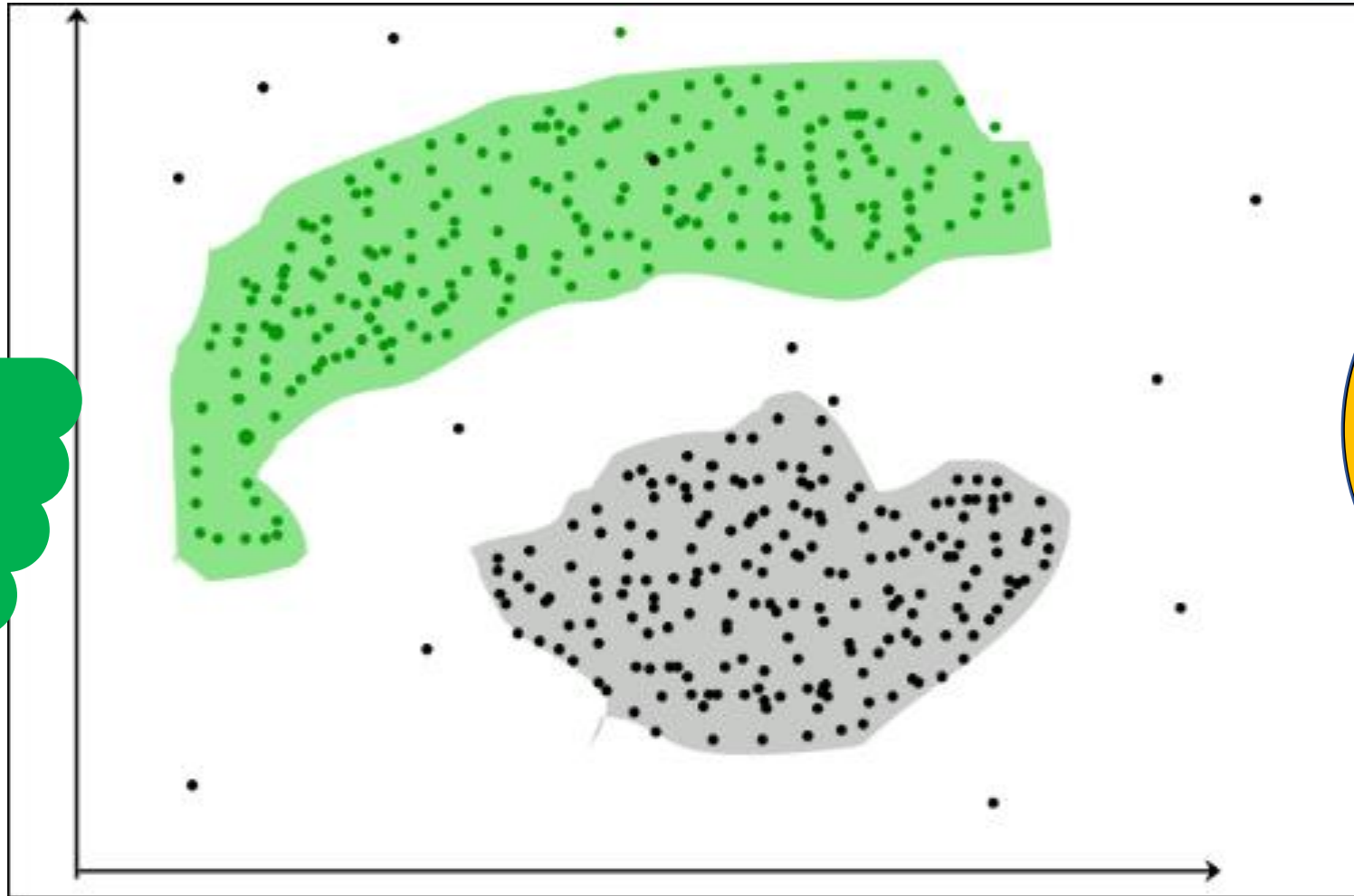


Fig. 2: Image search for "machine learning model images", www.towardsdatascience.com, Accessed 20 Feb. 2023.

What is a Model?

What is a model in the context of AI and machine learning?

- A mathematical function...
- that maps input data...
- via training data that has been annotated to identify desired outputs...
- amongst the surrounding data...
- to better recognize patterns or “learn” how to identify or predict these outputs on its own.
- A KEY component of AI/ML that allows predictions and decisions to be made based on complex, variable data WITHOUT code changes



Examples of ML in Life Sciences

Here are just some examples/use-cases:

1) Drug Discovery:

- To **identify potential new drug candidates** by analyzing large amounts of chemical and biological data.
- Help researchers **identify patterns in the data that may not be immediately apparent** and can be **used to predict the efficacy and safety of new drug candidates**.



Fig. 3: Image search for "Drug Discovery". www.uktechnews.com. Accessed 20 Feb. 2023.

Examples of ML in Life Sciences (cont'd)

Here are just some examples/use-cases:

2) Genomics and Personalized Medicine:

- To analyze genomic data and **identify genetic markers that may be associated with certain diseases or drug responses.**
- To develop **personalized treatment plans** based on a patient's individual genomic profile.



Fig. 4: Image search for "Genomics and Personalized Medicine". www.dnatix.com. Accessed 20 Feb. 2023.

Examples of ML in Life Sciences (cont'd)

Here are just some examples/use-cases:

3) Clinical Trials:

- To design and optimize clinical trials by predicting patient outcomes and identifying patient populations that are most likely to respond to a given treatment.
- To monitor and analyze data from ongoing clinical trials, and to predict the likelihood of adverse events or treatment failures.



Fig. 5: Image search for "Clinical Trial Design Optimization".
www.stanforduniversity.com. Accessed 20 Feb. 2023.

Examples of ML in Life Sciences (cont'd)

Here are just some examples/use-cases:

4) Medical Imaging:

- To analyze medical images, such as X-rays, CT scans, and MRIs, to **identify patterns and anomalies that may indicate disease or injury**.
- To **predict disease progression and treatment outcomes** based on imaging data.



Fig. 6: Image search for "AI in Medical Imaging". www.boldbusiness.com. Accessed 20 Feb. 2023.

Examples of ML in Life Sciences (cont'd)

Here are just some examples/use-cases:

5) Electronic Health Records:

- To analyze electronic health records to identify patterns and trends in patient data, such as disease prevalence, treatment outcomes, and healthcare utilization.
- To predict patient outcomes and to identify patients who may be at risk of developing certain diseases or conditions.



Fig. 7: Image search for "AI with EHR". www.dreamstime.com. Accessed 20 Feb. 2023.

ML Models vs. DL Models

Feature	Machine Learning	Deep Learning
Approach to learning	Uses algorithms to learn patterns and relationships in data	Uses artificial neural networks to learn representations of data
Feature engineering	Requires human experts to engineer relevant features (manual extraction)	Learns features automatically from the data
Complexity of models	Typically, less complex than deep learning models	Typically, more complex than traditional machine learning models
Types of data	Suited for structured data with relatively few features	Suited for unstructured data with large numbers of features
Training data size	Can work well with smaller training datasets	Often requires large amounts of training data to achieve optimal performance
Training time	Can be trained relatively quickly compared to deep learning models	Can require longer training times due to the complexity of the models (requires GPU-power over CPU)
Interpretability	Can be more interpretable than deep learning models	Can be more difficult to interpret due to the complexity of the models

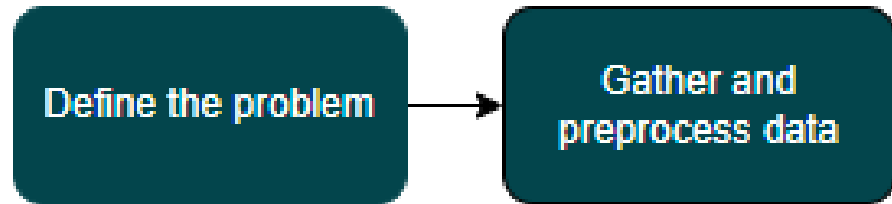
Building/Training a Model (example)



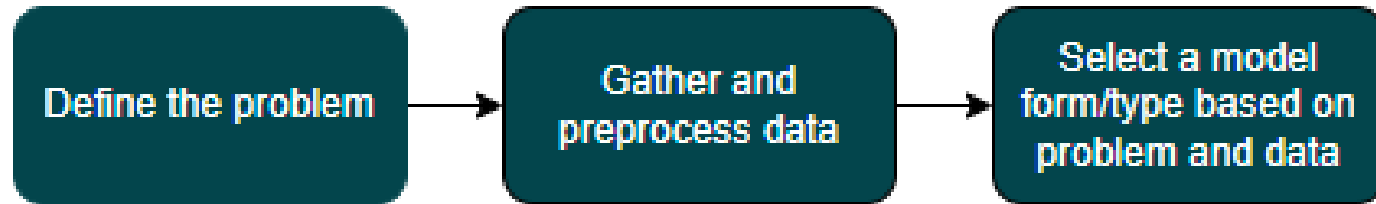
Building/Training a Model (example) (cont'd)

Define the problem

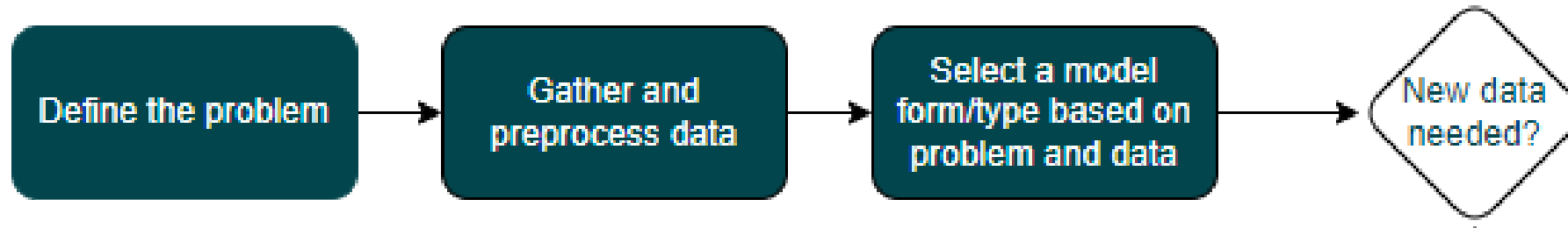
Building/Training a Model (example) (cont'd)



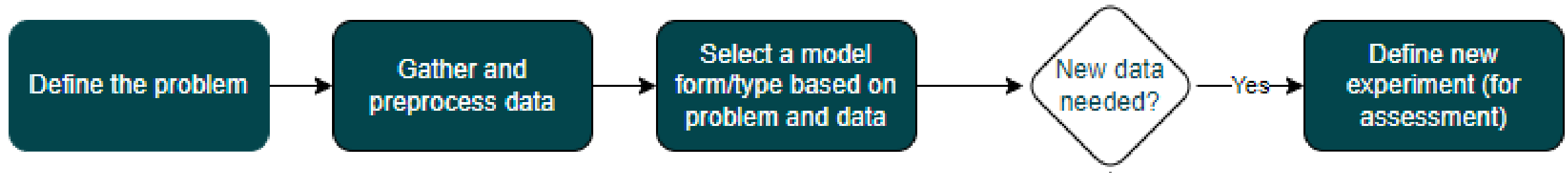
Building/Training a Model (example) (cont'd)



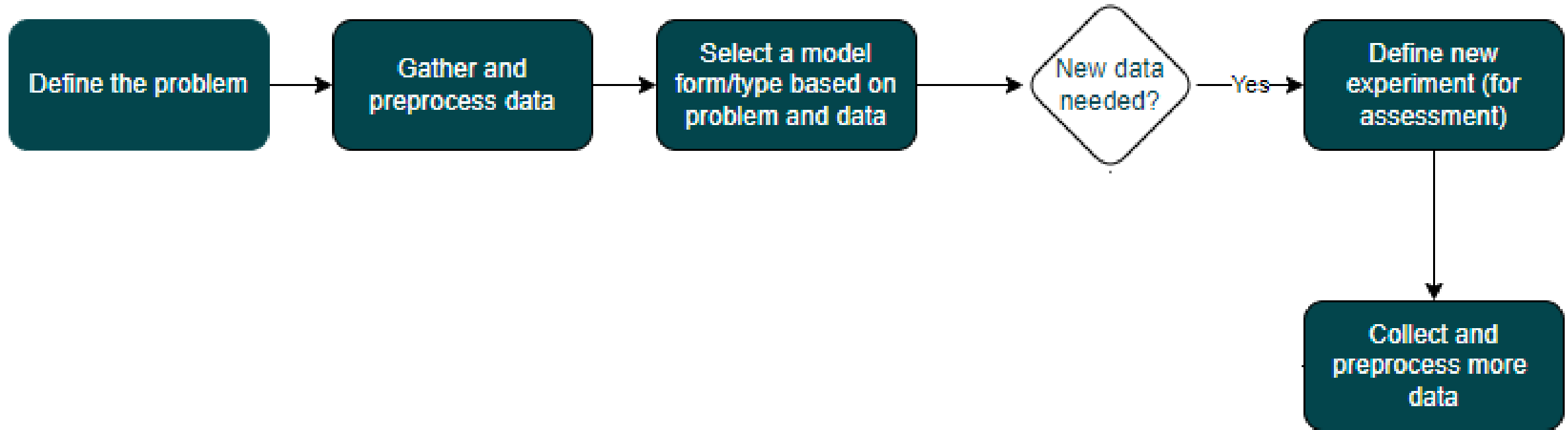
Building/Training a Model (example) (cont'd)



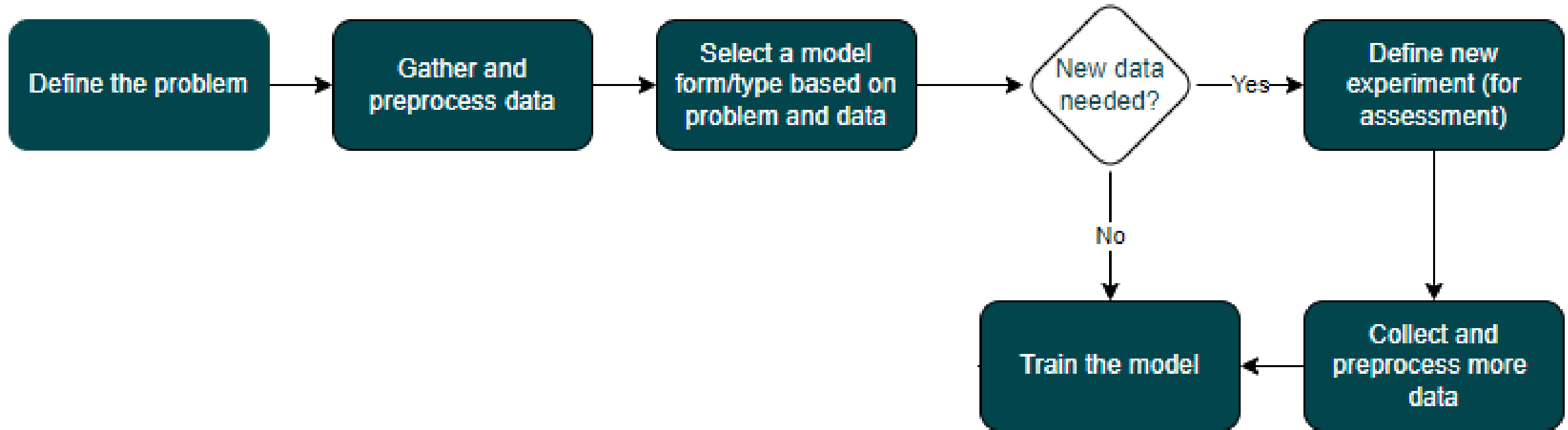
Building/Training a Model (example) (cont'd)



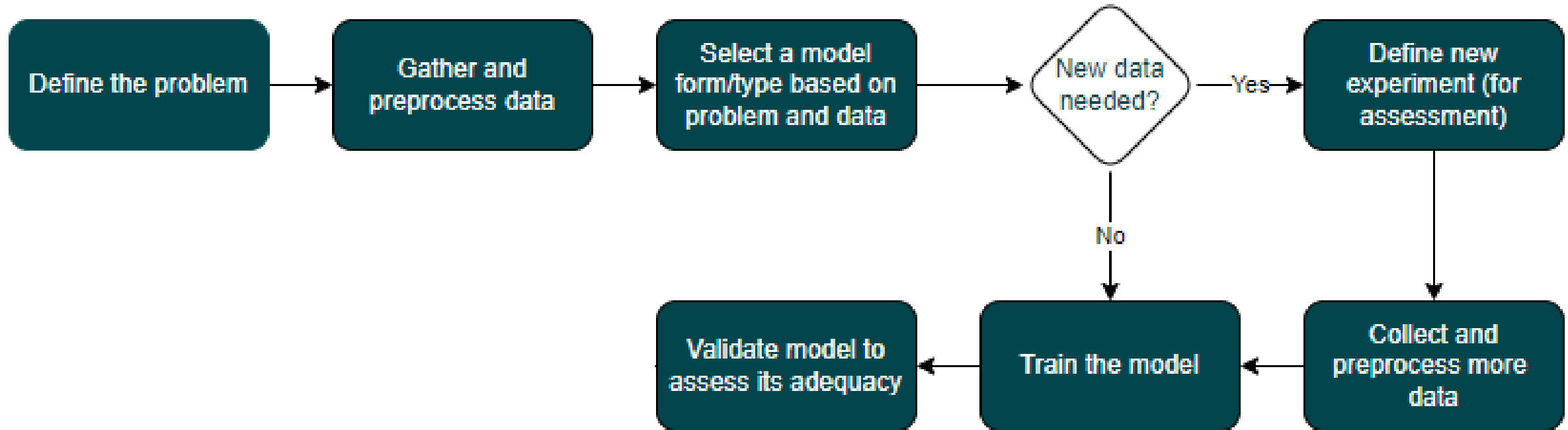
Building/Training a Model (example) (cont'd)



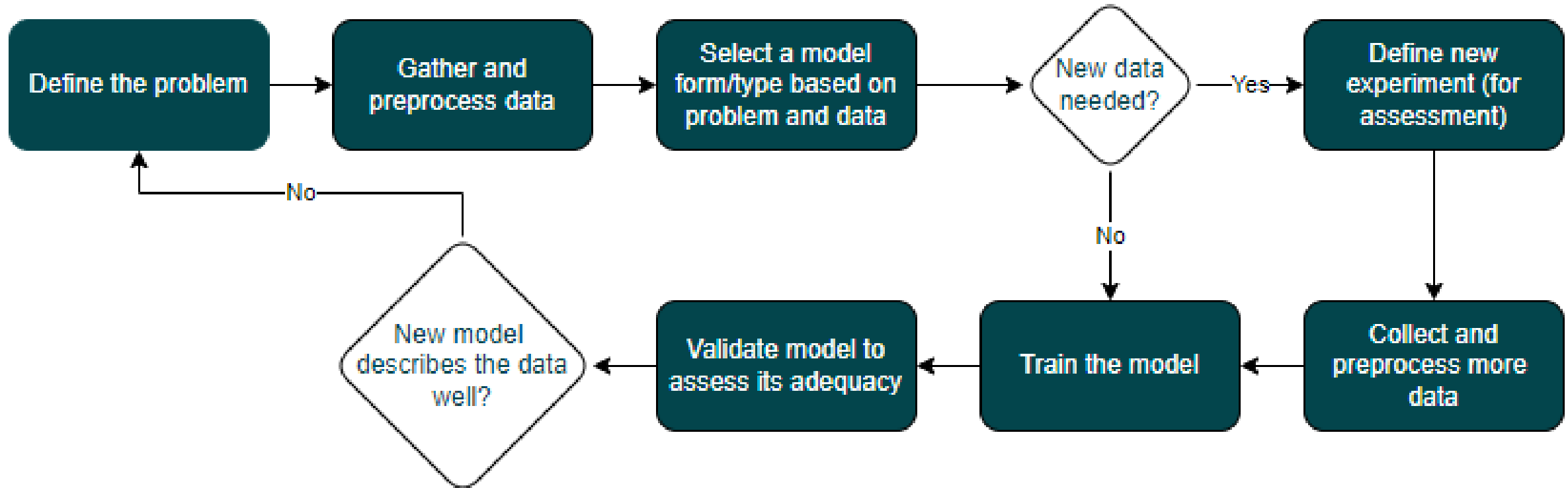
Building/Training a Model (example) (cont'd)



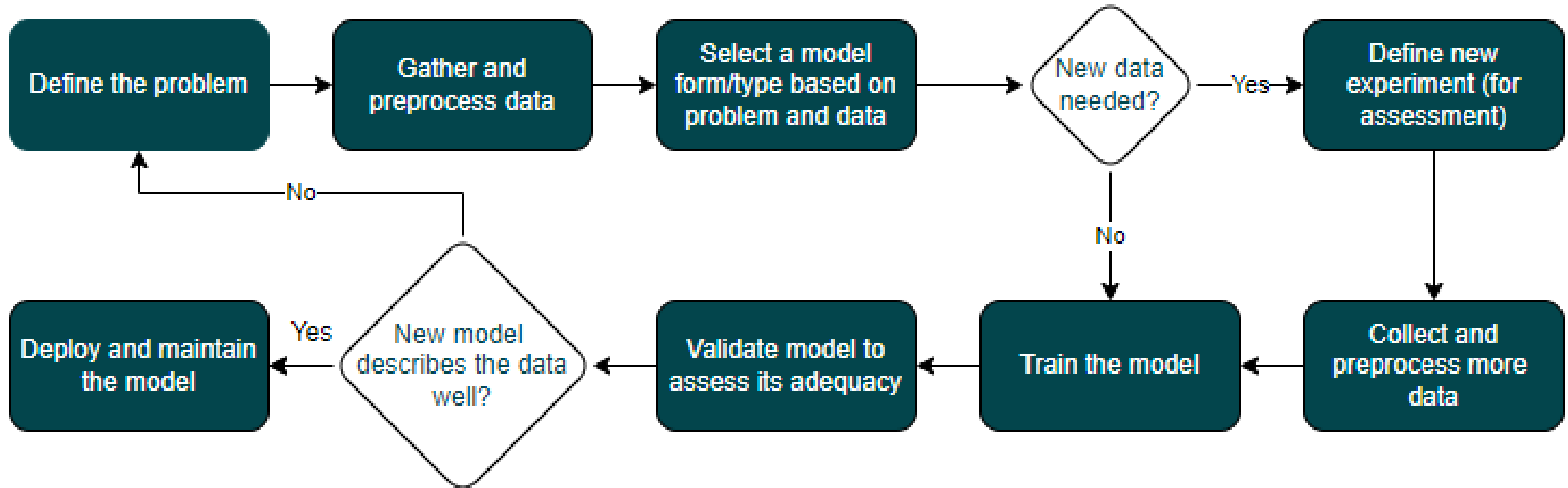
Building/Training a Model (example) (cont'd)



Building/Training a Model (example) (cont'd)



Building/Training a Model (example) (cont'd)



Building/Training a Model (example) (cont'd)

Example:

- 1) **Define the Problem:** Would like to be able to identify and highlight potentially sensitive information in a document (PDF), i.e., names, dates, phone numbers, etc., to facilitate an anonymization QC process.
- 2) **Gather Data:** Data is currently limited for this “Proof of Concept”. We only have a couple of documents to start with.
- 3) **Preprocess Data:** The data is in PDF form, so we are going to need to extract the data into text form, and use an annotation tool to create the entities to use for training the model.

Annotation Tool:

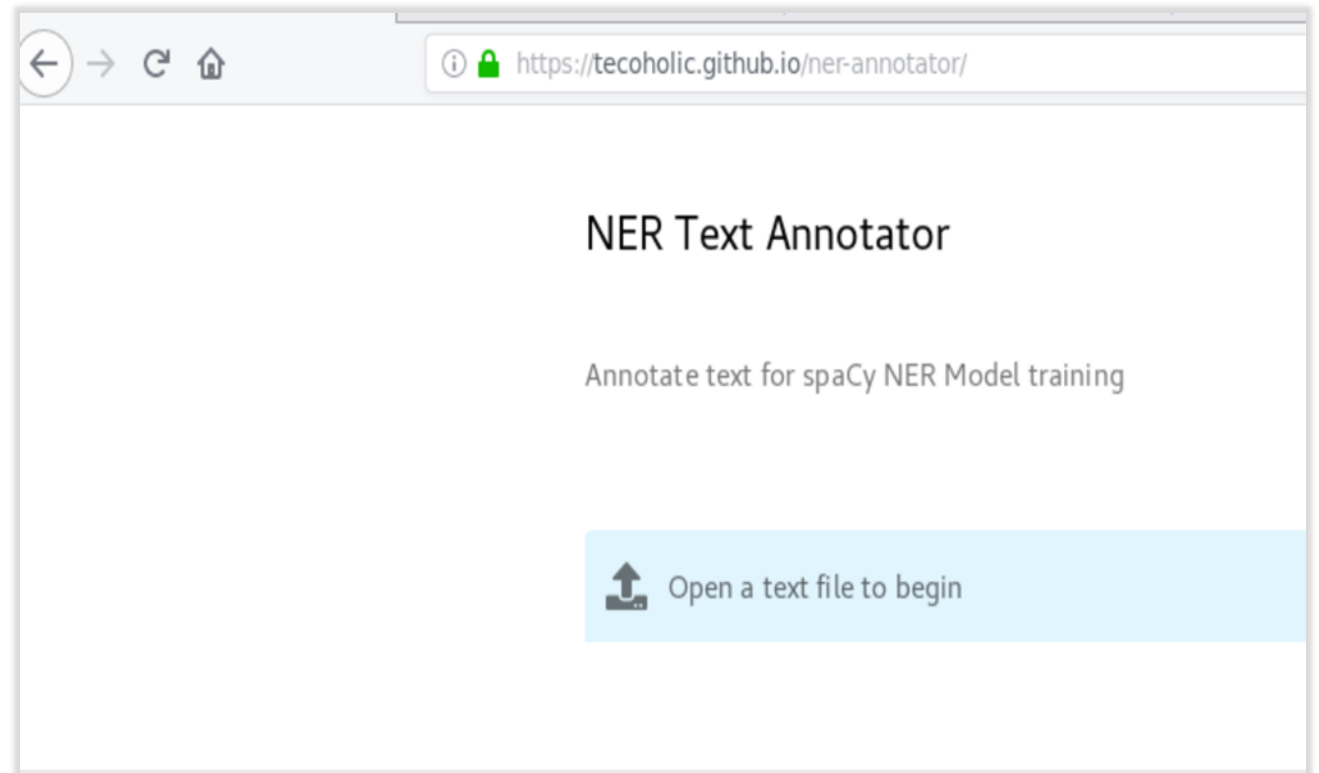


Fig. 8: Screen Grab of the NER Text Annotator, an open source tool available at <https://tecoholic.github.io/ner-annotator/>.

Building/Training a Model (example) (cont'd)

Annotation Tool: Annotation Process

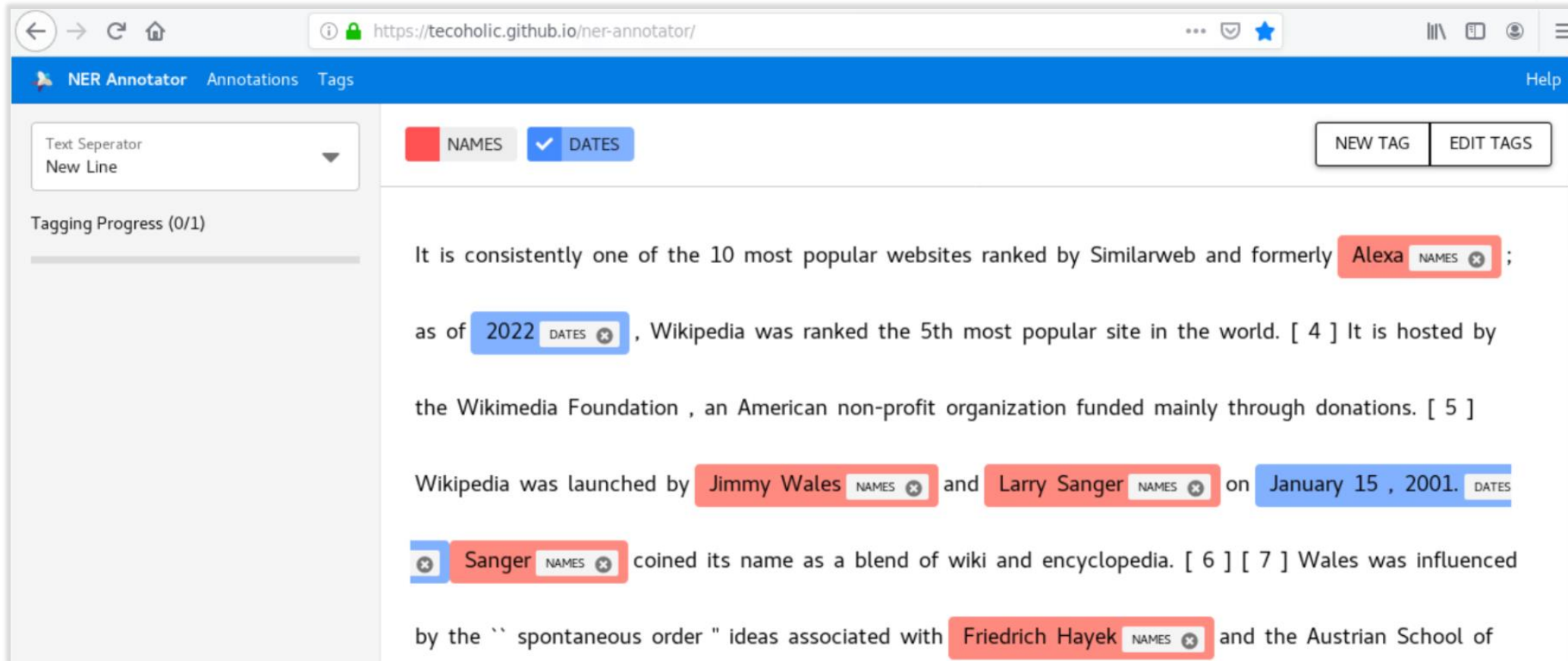


Fig. 9: Screen Grab of the NER Text Annotator, an open source tool available at <https://tecoholic.github.io/ner-annotator>.

Building/Training a Model (example) (cont'd)

Annotation Tool: Generates labeled data to use for model training

```
1 {"classes":["NAMES","DATES"],"annotations":[" It is consistently one of the 10 most popular websites ranked by Similarweb and  
  formerly Alexa; as of 2022, Wikipedia was ranked the 5th most popular site in the world.[4] It is hosted by the Wikimedia  
  Foundation, an American non-profit organization funded mainly through donations.[5] Wikipedia was launched by Jimmy Wales and  
  Larry Sanger on January 15, 2001. Sanger coined its name as a blend of wiki and encyclopedia.[6][7] Wales was influenced by the \  
  spontaneous order\" ideas associated with Friedrich Hayek and the Austrian School of economics after being exposed to these ideas  
  by the libertarian economist Mark Thornton.[8] Initially available only in English, versions in other languages were quickly  
  developed. Its combined editions comprise more than 60 million articles, attracting around 2 billion unique device visits per  
  month and more than 15 million edits per month (about 5.7 edits per second on average) as of January 2023.[9][10] In 2006, Time  
  magazine stated that the policy of allowing anyone to edit had made Wikipedia the \"biggest (and perhaps best) encyclopedia in  
  the world\".\"},{\"entities\":[[90,95,\"NAMES\"],[103,107,\"DATES\"],[313,324,\"NAMES\"],[329,341,\"NAMES\"],[345,362,\"DATES\"],[363,369,\"NAMES  
  \"],[499,514,\"NAMES\"],[616,630,\"NAMES\"],[931,944,\"DATES\"],[955,959,\"DATES\"]]]}]}
```

Fig. 10: Screen Grab of the JSON file generated by NER Text Annotator, an open source tool available at <https://tecoholic.github.io/ner-annotator>.

Building/Training a Model (example) (cont'd)

Example (cont'd):

4) Select a Model Type: As we are working with text, we need a model proficient with NLP. The documents are few and relatively small, and for this POC, we are not yet concerned with scalability but rather addressing functional requirements. So, we are going to go with a spaCy (blank/new) model type with only a few features; consequently, this will be an ML model.

5) Preprocess Data (Revisited): After considering and deciding on a model type, we recognized that there will need to be an additional step to ensure the training and test data are in a spaCy-compliant format. These requirements are available freely on spaCy's website for each of the 5 model types available. So, let's run code to reformat the training data for use with spaCy...

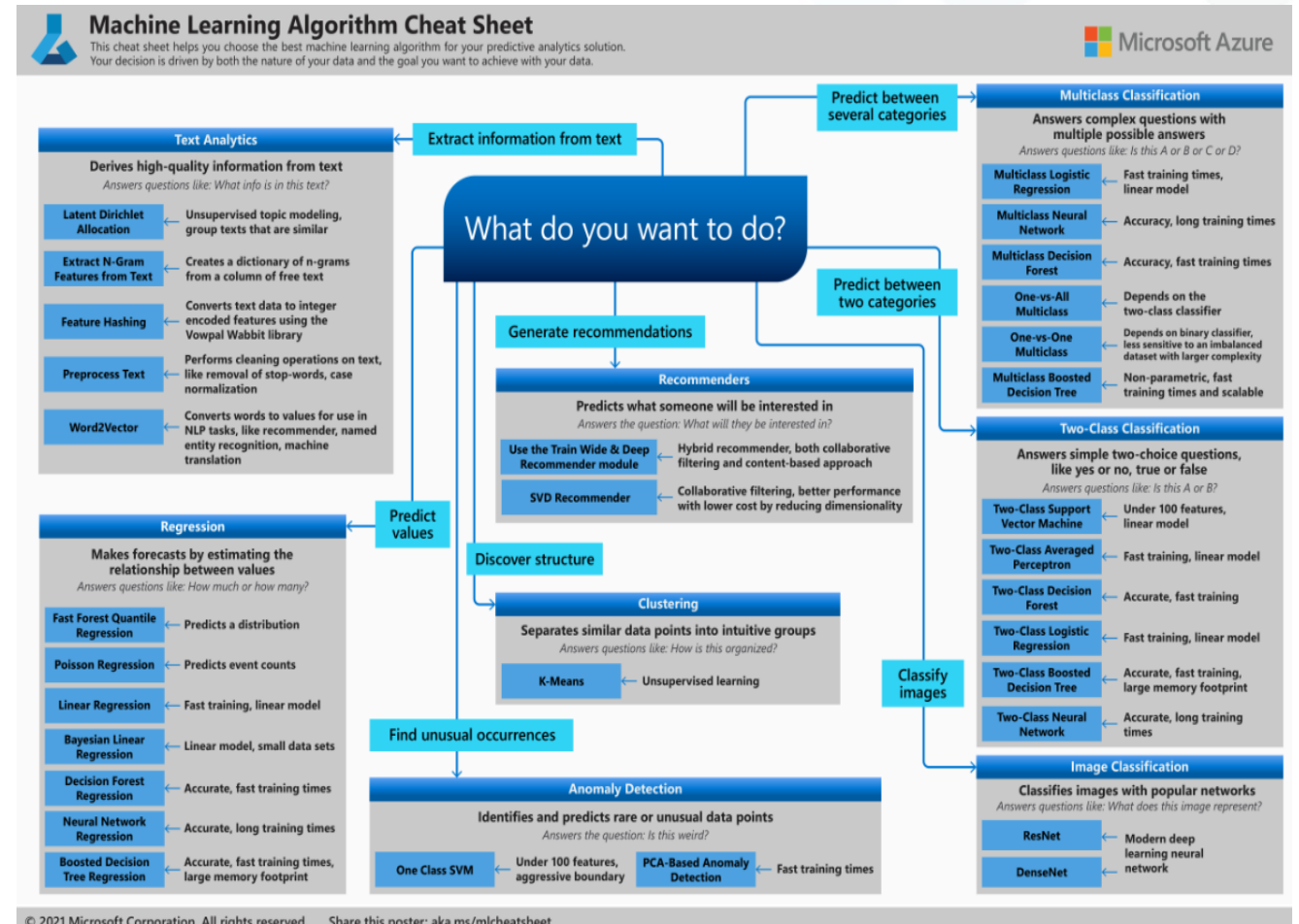


Fig. 12: Screen Grab of the NER Text Annotator, an open source tool available at <https://tecoholic.github.io/ner-annotator>.

Building/Training a Model (example) (cont'd)

Example (cont'd):

- 6) **Train the Model:** Again, as we are data-limited in the POC, we are going to use a standard 70/30% split of each of our 2 files for training/test data. Now that our data is in a spaCy-compliant format, let's run some code to fit the model using our data...
- 7) **Assess the Model:** We already know that we are working with very limited data and that our model is just getting started. We can see this in the performance metrics shared in the log. We will run some tests now to see how well this model fares on our test data/docs and others as they become available, and proceed from there.

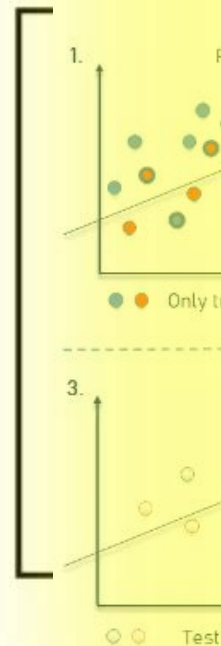
```
In [*]: 1
        2 doc = nlp_best_3Feb(text)
        3 spacy.displacy.serve(doc, style='ent')
```

Various collaborative online encyclopedias were attempted before the start of Wikipedia ORG , but with limited success.[24] Wikipedia ORG began as a complementary project for Nupedia ORG , a free online English-language MISC encyclopedia project whose articles were written by experts and reviewed under a formal process.[25] It was founded on March 9, 2000, under the ownership of Bomis ORG , a web portal company. Its main figures were Bomis ORG CEO Jimmy Wales PER and Larry Sanger PER , editor-in-chief for Nupedia ORG and later Wikipedia.[1][26] Nupedia ORG was initially licensed under its own Nupedia Open Content License MISC , but before Wikipedia ORG was founded, Nupedia ORG switched to the GNU Free Documentation License MISC at the urging of Richard Stallman.[27 PER] Wales PER is credited with defining the goal of making a publicly editable encyclopedia,[28][29] while Sanger PER is credited with the strategy of using a wiki to reach that goal.[30] On January 10, 2001, Sanger PER proposed on the Nupedia ORG mailing list to create a wiki as a "feeder" project for Nupedia ORG .

Assessing Model Effectiveness (notes)

Assessing a model's effectiveness involves using a set of data that was not used for training. Here are some ways to assess a model's effectiveness:

- 1) Train/test split
- 2) Cross-validation
- 3) Performance metrics
- 4) Visualizations
- 5) Refinement



a set of data
using a model's



Machine Learning Model
Trains on the
training dataset

4. Model
predicts on
test data

Diagnosing and Fixing Poorly Behaving Models (reference)

Issue	Example	Resolution	Additional Notes
Overfitting	Model performs well on training data but poorly on test data	Regularization, cross-validation, early stopping	Overfitting occurs when a model fits the training data too closely, to the point where it memorizes the data and performs poorly on new data. Regularization techniques such as L1 and L2 regularization add a penalty term to the loss function to discourage complex models. Cross validation helps to evaluate the model's performance on new data, and early stopping stops the training process before the model starts to overfit.
Underfitting	Model performs poorly on both training and test data	Increase model complexity, feature engineering	underfitting occurs when a model is too simple to capture the underlying patterns in the data. Increasing the model complexity, for example by adding more layers to a neural network or increasing the degree of a polynomial regression, can help to address underfitting. Feature engineering involves creating new features or transforming existing features to better capture the relationships between the features and target variable.
Imbalanced Classes	One class has significantly fewer examples than the others	Resampling, weighting classes	Imbalanced classes can cause a model to be biased towards the majority class, resulting in poor performance on the minority class. Resampling techniques such as oversampling or undersampling can help to balance the class distribution. Weighting the classes during training can also help to balance the class distribution.
Hyperparameter Tuning	Choosing the optimal values for the model's hyperparameters	Grid search, random search, Bayesian optimization	Hyperparameters are parameters that are set before training the model, such as learning rate or regularization strength. Choosing optimal hyperparameters is crucial for a model's performance. Grid search involves exhaustively searching over a predefined set of hyperparameters. Random search involves randomly sampling from a predefined hyperparameter space. Bayesian optimization involves modeling the relationship between hyperparameters and model performance and using this to guide the search for optimal hyperparameters.
Data Quality	Noisy or missing data, irrelevant features	Data cleaning, feature selection, feature engineering	Poor data quality can lead to poor model performance. Data cleaning involves removing or correcting noisy or missing data. Feature selection involves choosing a subset of the available features that are most relevant to the target variable. Feature engineering involves creating new features or transforming existing features to better capture the relationships between the features and target variable.

Future of ML in Life Sciences

- AI/ML has the potential to transform the life sciences industry and improve patient outcomes.
- As reported earlier, there are many examples for how this potential is being realized in drug discovery, personalized medicine, and medical imaging to name just a few.
- However, there are many challenges needing to be addressed such as data privacy and security, ethical concerns, and regulatory barriers.
- As these issues are addressed and the technology continues to improve, we can expect to see even greater benefits of AI/ML in life sciences.



Conclusion

- In summary, we covered the following:
 - 1) Intro to Machine Learning models
 - 2) Examples of AI/ML in Life Sciences
 - 3) Examples for how to build and test a simple model
- Hopefully this presentation has inspired you to try it for yourself, and that you will continue to test, learn and explore machine learning in the life science industry!
- Be sure to attend the afternoon sessions in the Emerging Trends stream to see how AI/ML-empowered innovations are enhancing capabilities within biometrics and medical writing!

References & Thank You's

- 1) OpenAI's ChatGPT – for significantly reducing the amount of “googling” time normally required for producing these presentations
- 2) GenInvo's Data Science and Product Development Teams – particularly Nadia and Neeraj for their reviews, sharing experiences and to Neeraj for also compiling some basic code samples



Ideas
Delivered
Intuitively

Thank You!

+1-833-GenInvo (436-4686) | info@geninvo.com | www.geninvo.com

