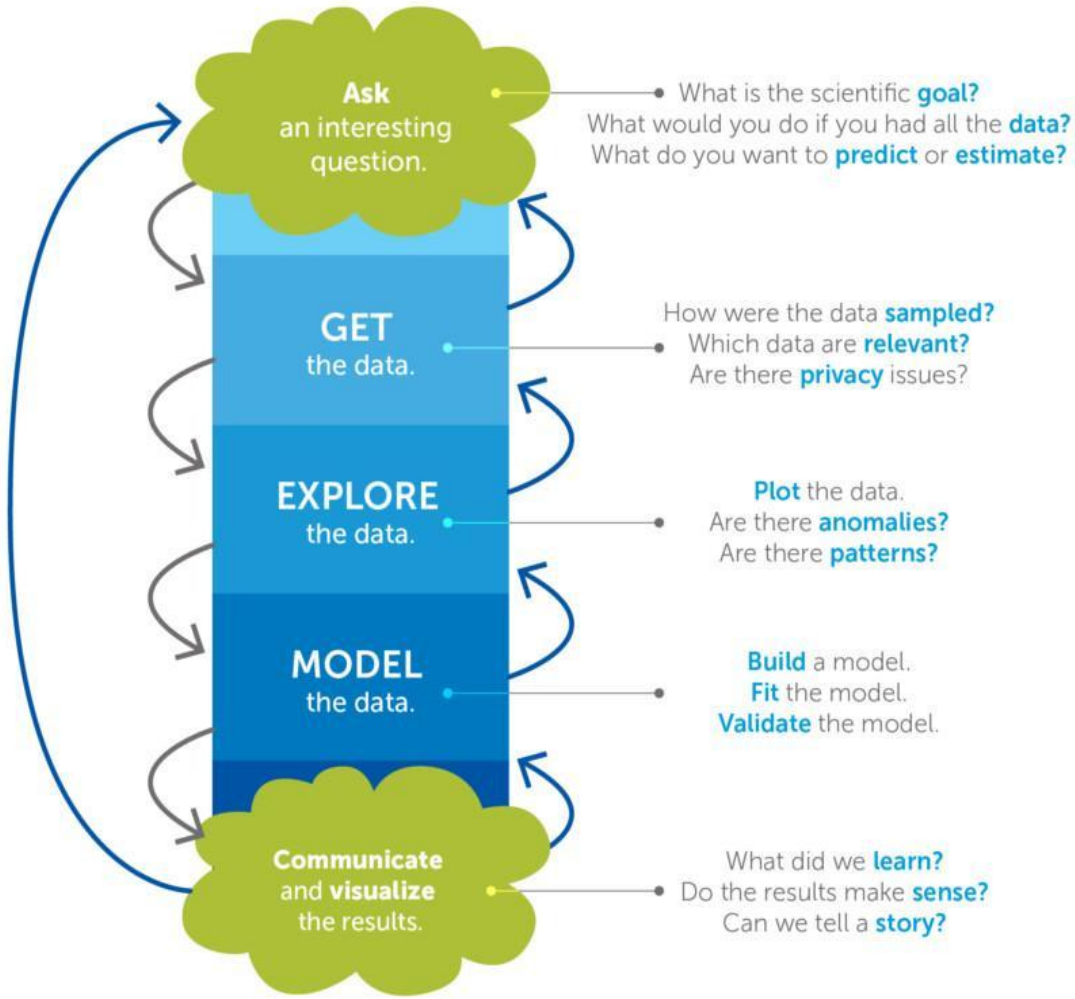
Deploy and Monitor

CPDSAI
Chantri Polprasert

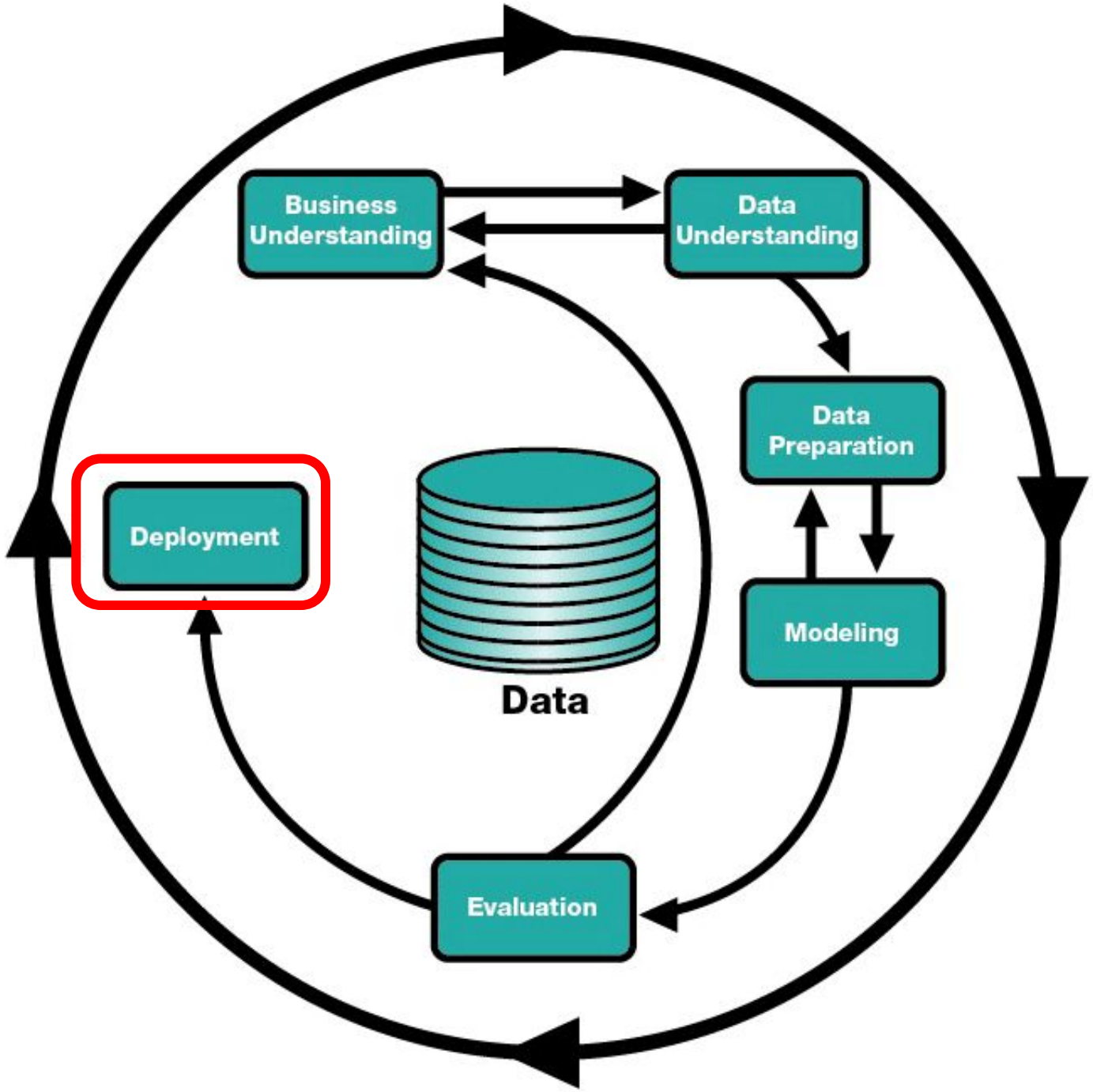
Outline

- Interpretability/Explainability in Machine Learning
- Deployment
- Workshop: Deploy ML using Flask
- Monitoring and Feedback Loops
- Concluding Remarks

The Data Science Process

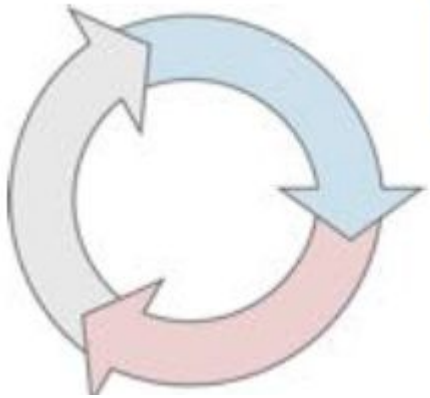


Derived from the work of Joe Blitzstein and Hanspeter Pfister,
originally created for the Harvard data science course http://cs109.org/.

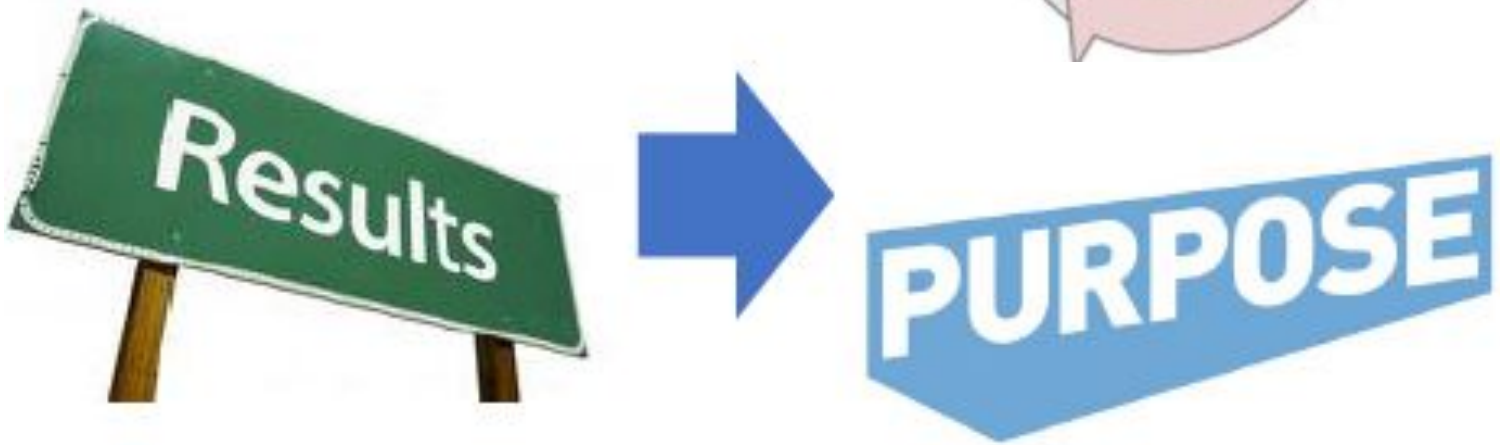


Cross Industry Standard Process for Data Mining (CRISP-DM)

Step 5: Apply Results

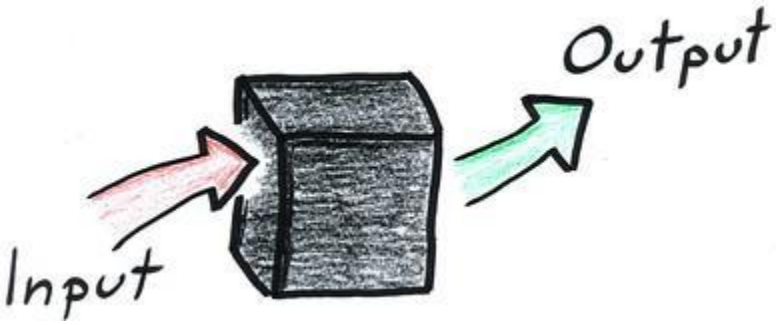


Iterative process



Interpretability/Explainability in Machine Learning

Interpretability



“Interpretability is the degree to which a human can understand the cause of a decision.”

-- Tim Miller (2017)

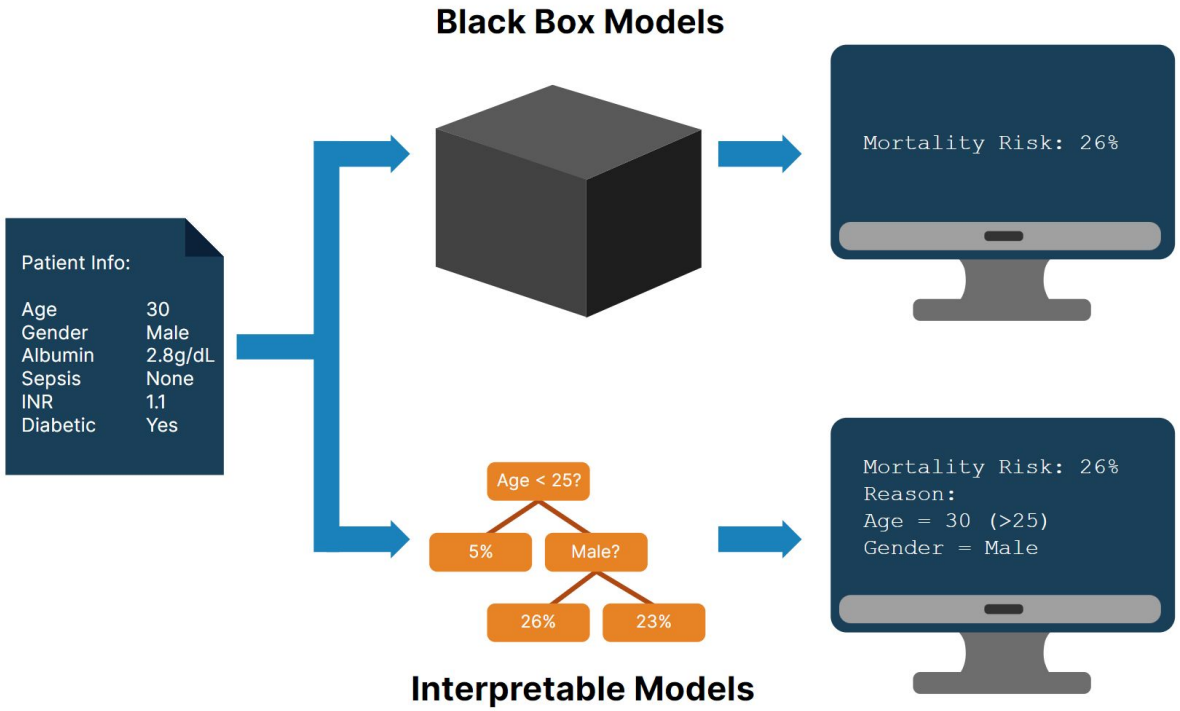
“Interpretability is the degree to which a human can consistently predict the model's result”

-- Been Kim et. al. (2016)

- The *higher* the interpretability of a machine learning model, the *easier* it is for someone to comprehend why certain decisions or predictions have been made.
- A model is better interpretable than another model if its decisions are easier for a human to comprehend than decisions from the other model.

Possible scenarios: healthcare

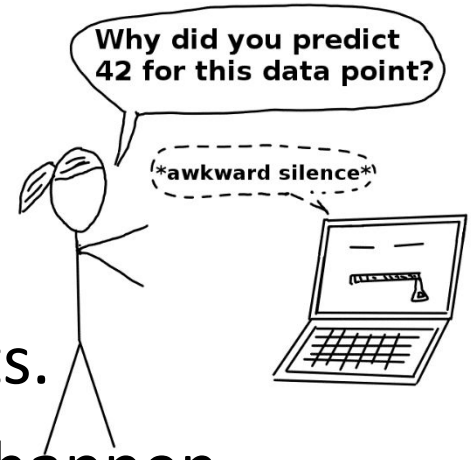
Interpretable AI



https://www.freepik.com/free-vector/autonomous-smart-car-automatic-wireless-sensor-driving-road-around-car-autonomous-smart-car-goes-scans-roads-observe-distance-automatic-braking-system_26413340.htm#query=self%20driving%20car&position=21&from_view=search&track=ais

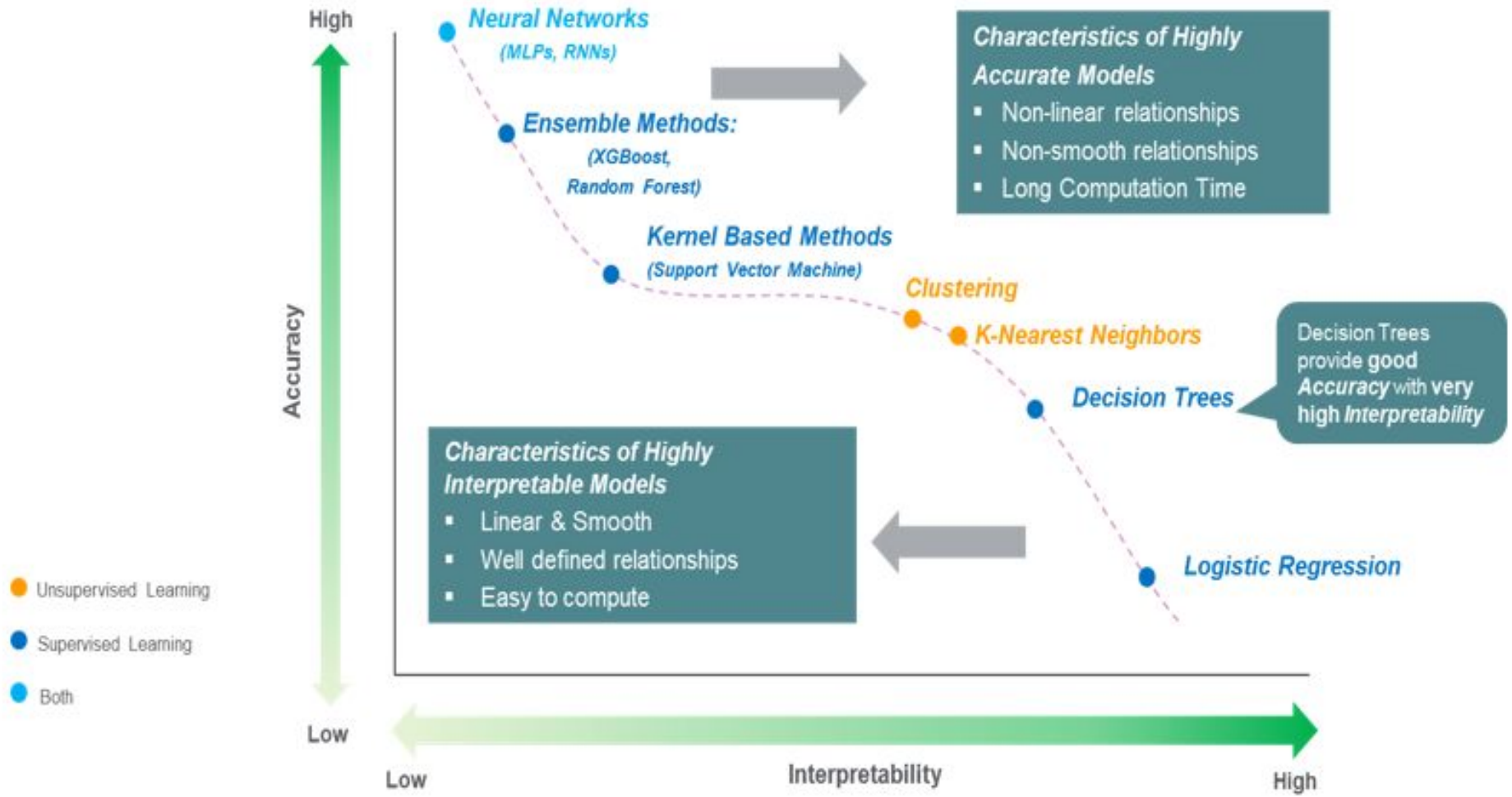
https://www.xcally.com/news/interpretability-vs-explainability-understanding-the-importance-in-artificial-intelligence/

Interpretability Vs. Explainability



- Both words were used <u>interchangeably</u> by some scientists.
- **Interpretability**: You are able to ***predict*** what is going to happen, given a change in input or <u>algorithmic parameters</u>. It's being able to look at an algorithm and go *yep, I can see what's happening here. Maybe without knowing why.*
- **Explainability**: You are able to ***explain*** the internal mechanics of a machine or deep learning system in human terms.
  - *Another term is explanations*
- **Interpretable Machine Learning** refers to methods and models that make the behavior and predictions of machine learning systems understandable to humans.

Tradeoff between Accuracy and Interpretability might not be True

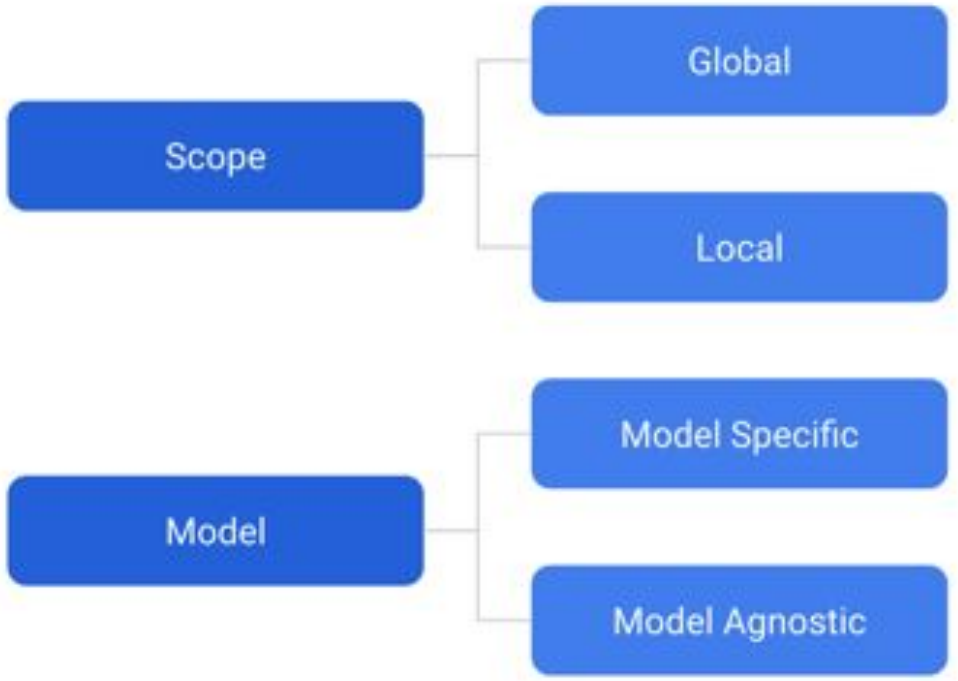


https://towardsdatascience.com/the-balance-accuracy-vs-interpretability-1b3861408062

Interpretability/Explainability
Why should we care?

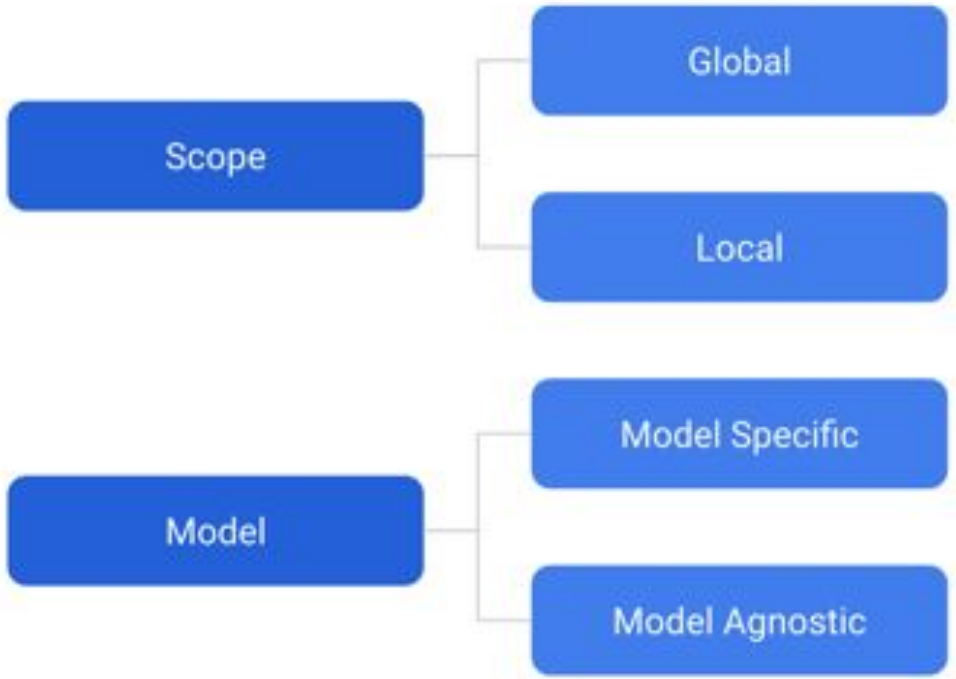
- Causality of features
- Understand model **bias**, **ethics**, **fairness**
  - Helps align with key business questions
- Increase trust in the model
- Satisfy regulatory requirements (e.g. GDPR, PDPA) ('A data subject has the right to "an explanation of the decision reached after [algorithmic] assessment'), Privacy
- Ability to debug or troubleshoot
- Help in communication to stakeholders
- Reliability or Robustness

Taxonomy of Interpretability Methods

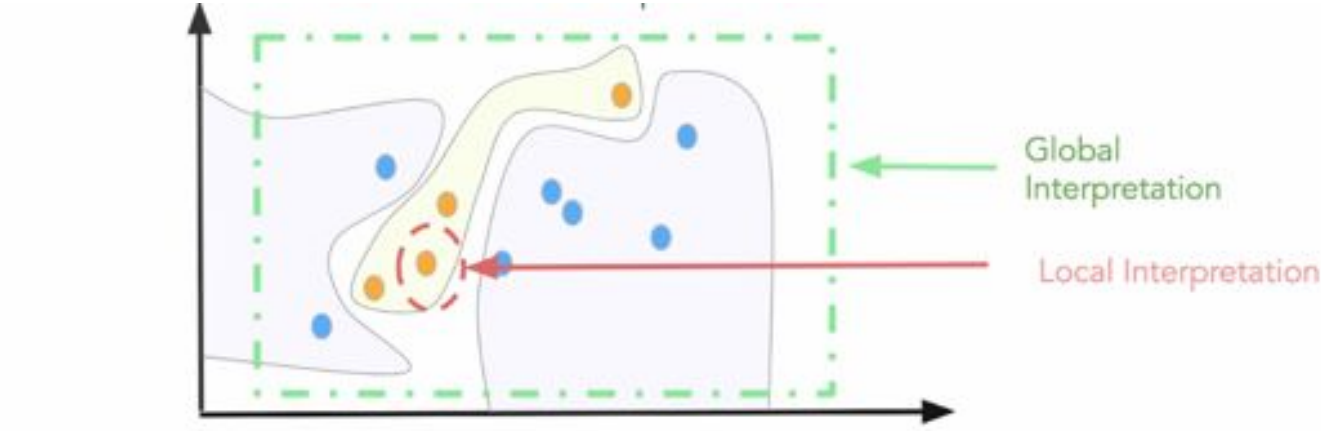


- **Model-specific or model-agnostic?**
  - Model-specific interpretation tools are limited to specific model classes: regression weights in a linear model
  - Model-agnostic tools can be used on any machine learning model and are applied after the model has been trained (post hoc).

Taxonomy of Interpretability Methods



- **Local or global?** Does the interpretation method explain an individual prediction or the entire model behavior?



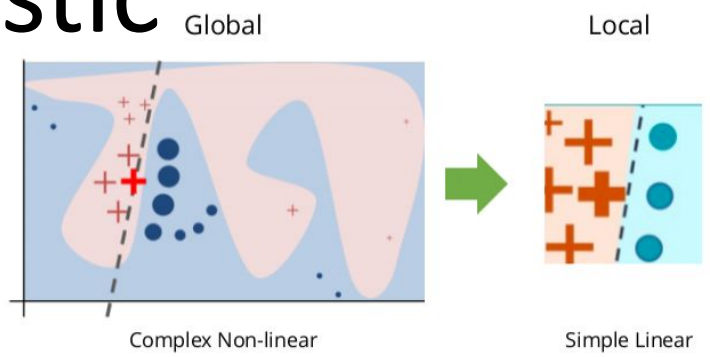
Summarizing Global and Local Interpretation (Source: DataScience.com)

Various Techniques / Libraries

- LIME (Local Interpretable Model-agnostic Explanations)
- SHAP (SHapley Additive exPlanation)
- Eli5 (Explain me like I am 5)
- PDP (Partial Dependence Plot)
- DeepLIFT (Deep Learning Important Features)

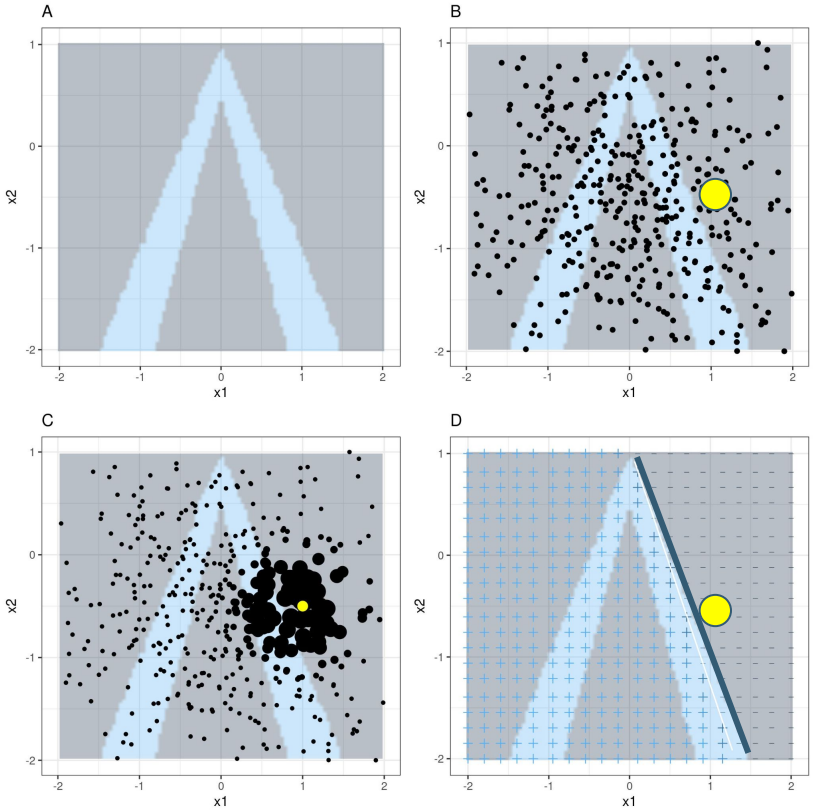
Local interpretable model-agnostic explanations (LIME)

- Use local surrogate model to approximate the predictions of the underlying black box model



How LIME works? (Given a prediction model and a test sample)

- LIME provides locally faithful explanations around the vicinity of the instance
- Many samples are created and points near the instance of interest are given more weight
- Simple model (Linear/Ridge) are obtained



Local interpretable model-agnostic explanations (LIME)

LIME in its current state is only able to give explanations for the following type of datasets:

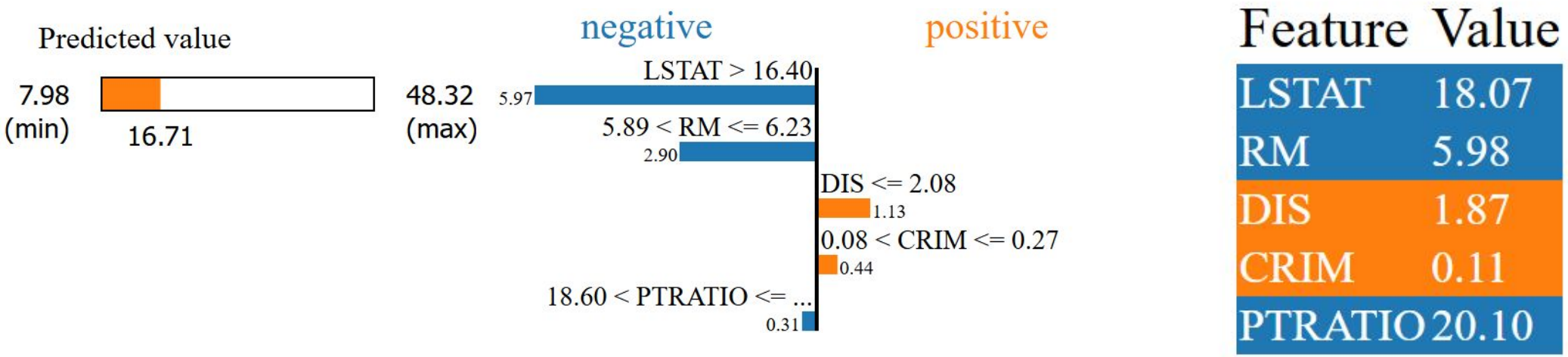
- Tabular datasets (lime.lime_tabular.LimeTabularExplainer): eg: Regression, classification datasets
- Image-related datasets (lime.lime_image.LimeImageExplainer)
- Text-related datasets (lime.lime_text.LimeTextExplainer)

Local interpretable model-agnostic explanations (LIME)

```
class lime.lime_tabular.LimeTabularExplainer(training_data, mode='classification',
training_labels=None, feature_names=None, categorical_features=None, categorical_names=None,
kernel_width=None, kernel=None, verbose=False, class_names=None, feature_selection='auto',
discretize_continuous=True, discretizer='quartile', sample_around_instance=False, random_state=None,
training_data_stats=None)
```

- Explains predictions on tabular (i.e. matrix) data. For numerical features, perturb them by sampling from a Normal(0,1)
- For categorical features, perturb by sampling according to the training distribution
- Sample Parameters:
  - training_data – numpy 2d array
  - mode – “classification” or “regression”
  - training_labels – labels for training data. Not required, but may be used by discretizer.
  - feature_names – list of names (strings) corresponding to the columns in the training data.

LIME on Boston Housing Dataset



- LSTAT (% lower status of the population) has 5.97 feature importance
- RM (Average number of rooms per dwelling) has 2.90 feature importance
- DIS (weighted distance to to employment center) has 1.13 feature importance

Deployment

Deployment

- Real-world machine learning involves more than just machine learning model training. – Josh Patterson
- To have an impact on the business, we want our model to be available for the end-users/stakeholders.
- How do we get our model to clients/stakeholders?
- What do we need to put the model into production?
- How do we begin?

3 Main Ways for Model Deployment

1. Model Server
   - Create web apps, API endpoints, HTTP request
   - Flask, Django, Streamlit
2. User's browser
   - Download and run (TensorFlow.jS)
3. Edge device (e.g. mobile device, IoT, sensors)
   - Install and run on the device (CoreML, TensorFlow Lite, tinyML)

Revolutionizing Traffic Safety: The Global tinyML Traffic Hackathon

https://www.wevolver.com/article/revolutionizing-traffic-safety-the-global-tinyml-traffic-hackathon

D. Sculley et.al. (2015)

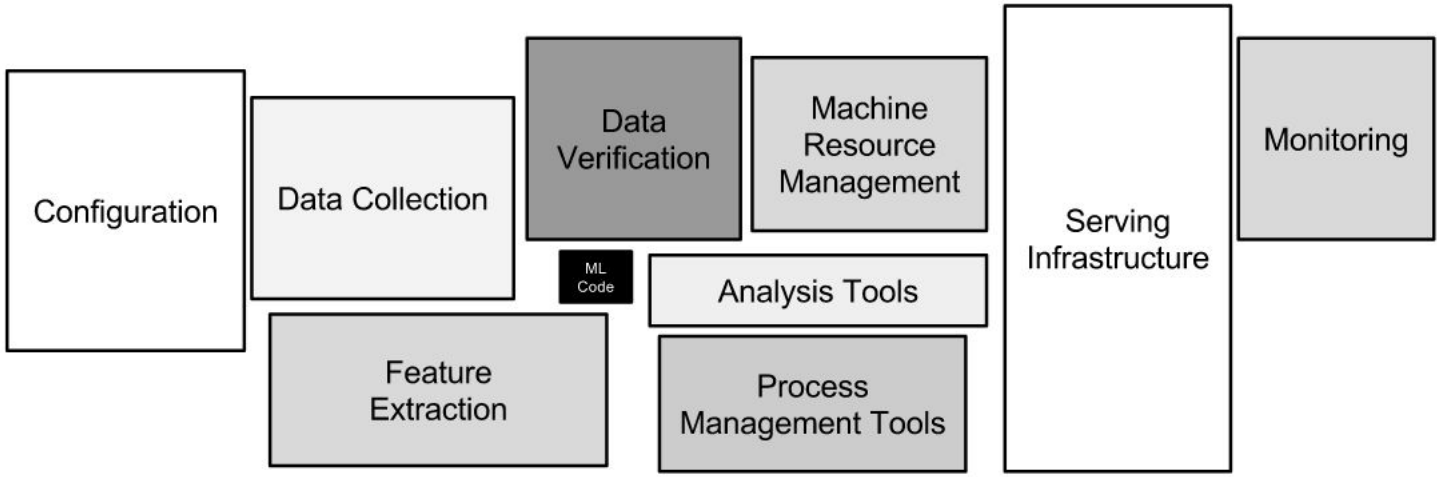


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Deploying is hard -- Vicki Boykis (2020)

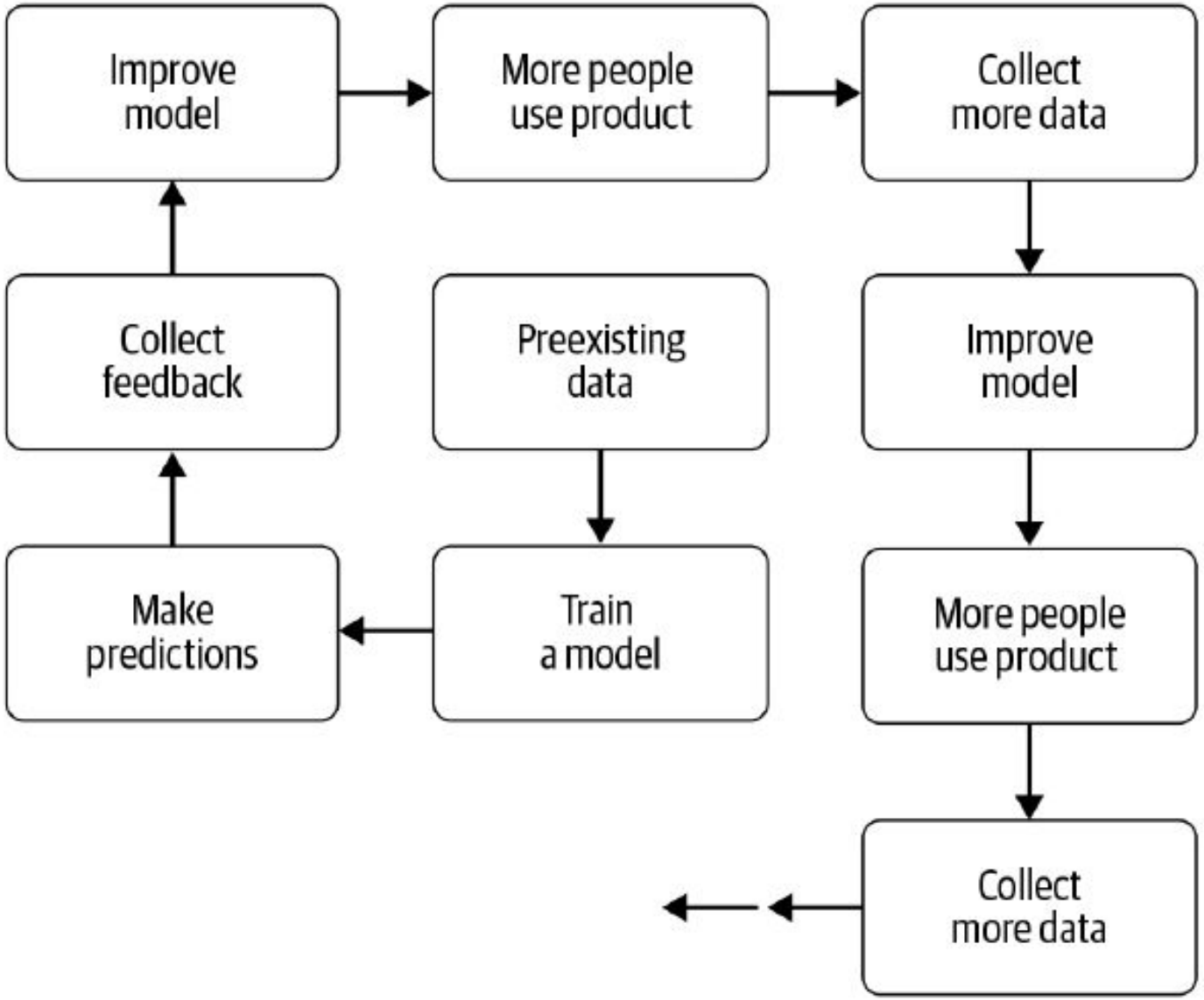
You're not only deploying software, you're also, in essence deploying data, that moves between various departments, in various formats, that changes as the model changes, and there are a ton of moving variables in the system that impact the other variables.

Monitoring and Feedback Loops

Not a One-off ML Model

- Software doesn’t just get deployed and forgotten
- It must be monitored, maintained, and updated regularly
- Training data itself must be treated as code (e.g, versioned)
  - New data pops up everyday
  - The behavior of a model is often quite opaque; it may pass all the tests on some data but fail entirely on others
- As soon as a model has users, it will require continuous updates and fine-tuning.
- Project grows.

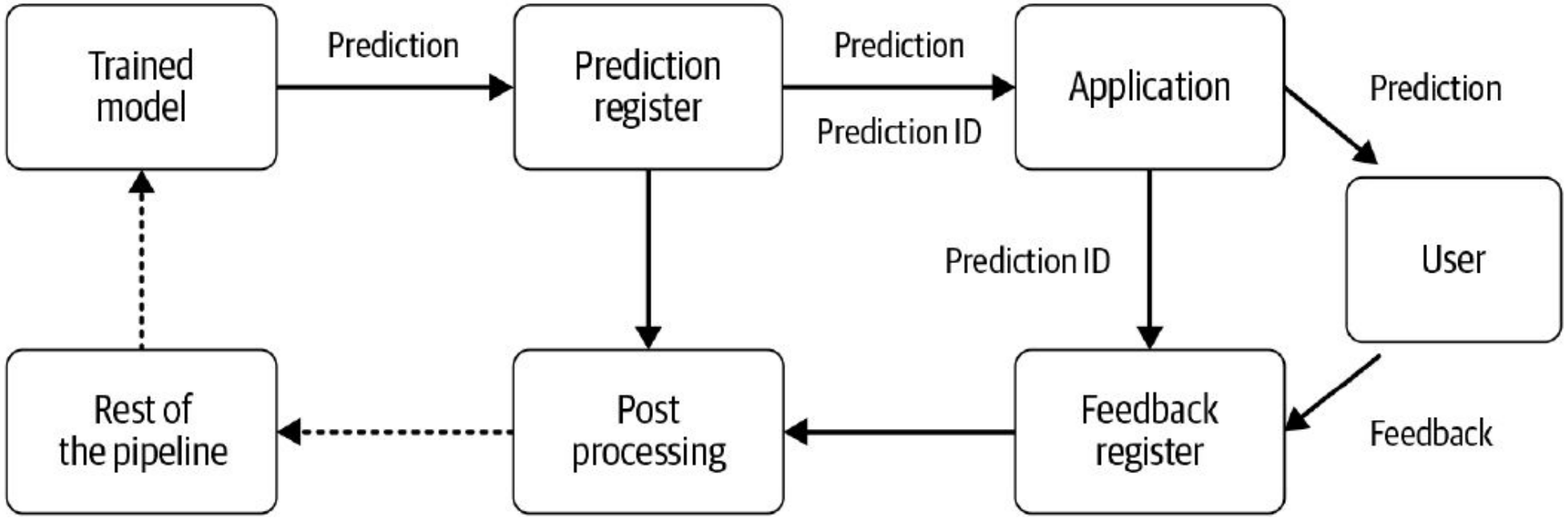
Data Flywheel



Model Monitoring

- Check to see if data inputs to the model are similar as seen during the training
- Check to see if model prediction distribution are similar to as seen during model training or model validation phase
- Ensure models are performing as expected per set service level agreement overtime
- Alerting on data or concept drift to take action as soon as model violates some of the key underlying assumptions
  - Data drift: new product category, data collection quality
  - Concept drift: for example, portability in Telecom industry impact churn
- Need feedbacks to improve mode

Tracking Feedback



Feedback Loops

- Intuition: we would like our ML system to continuously improve
- Feedback Loops help us
  - Collect new data to refresh models
    - Especially for models that are personalized, e.g. recommender system, predictive text
  - Provide information on the real-world use

Implicit and Explicit Feedback

- Implicit Feedback
  - People's actions in their normal usage
    - Click on the ad/search results
    - Buy products
    - Watch movies
- Explicit Feedback
  - Users give direct input on a prediction
    - Thumbs-up thumbs-down
    - Correcting prediction

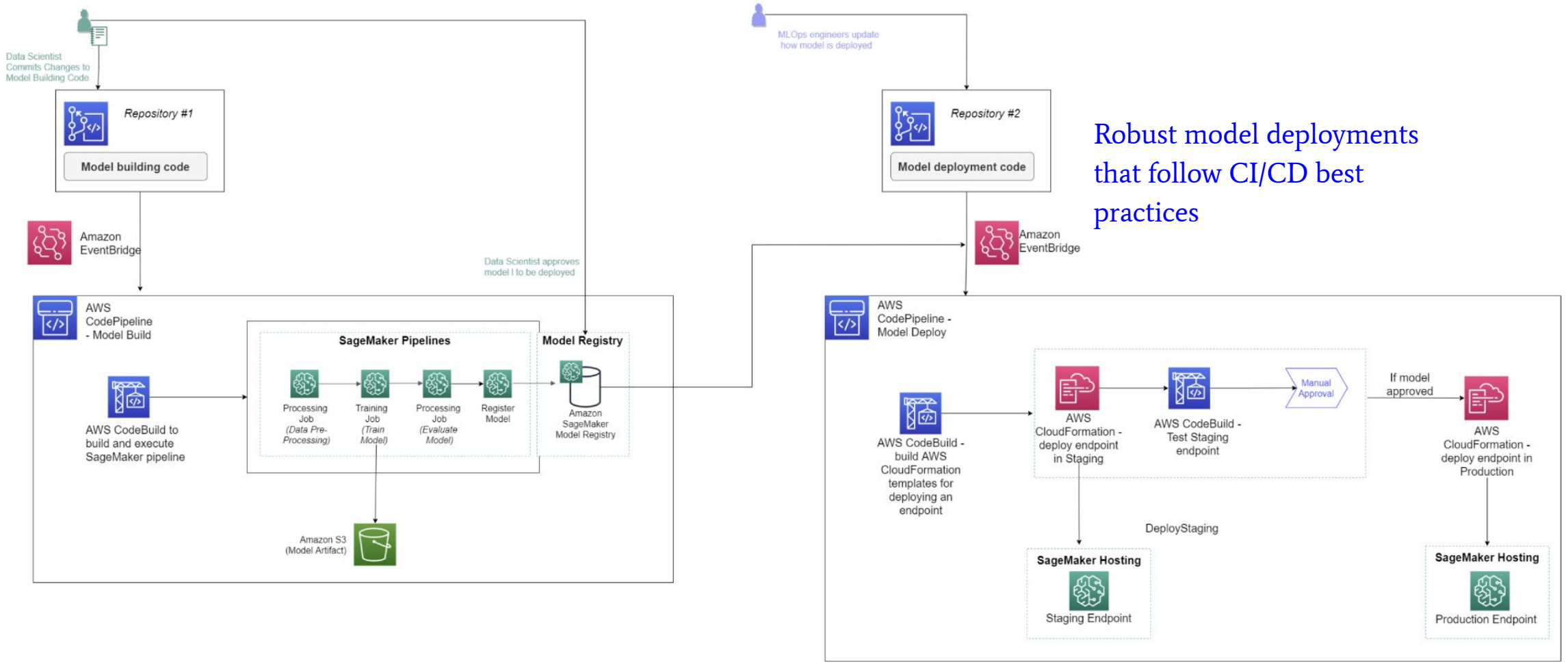
Options for Collecting Feedbacks

- Users take some action as a result of the prediction (Implicit)
- Users rate the quality of the prediction (explicit)
- Users correct the prediction (explicit)
- Crowdsourcing the annotations
- Expert annotations
- Producing feedback automatically

Machine Learning (Automated) Pipelines

- Version your data effectively and kick off a new model training run
- Validate the received data and check against data drift
- Efficiently preprocess data for your model training and validation
- Effectively train your machine learning models
- Track your model training
- Analyze and validate your trained and tuned models
- Deploy the validated model
- Scale the deployed model
- Capture new training data and model performance metrics with feedback loops

ML pipeline using AWS SageMaker



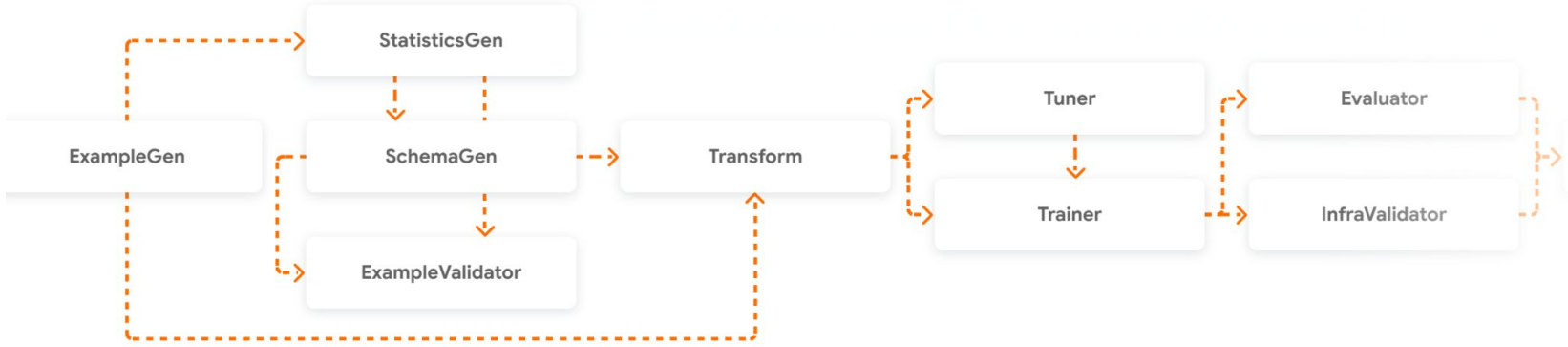
Robust model deployments that follow CI/CD best practices

End-to-End ML Platforms

- AWS SageMaker
- Google: TensorFlow eXtended (TFX)
- Uber: Michelangelo
- Airbnb: Bighead
- Netflix: Metaflow
- Lyft: Flyte

Amazon SageMaker

TensorFlow Extended



Concluding Remarks

Analytics Model Requirements

- **Business Relevance**: must solve the defined problem
- **Statistical Performance**: must have predictive power
  - What is the baseline?
- **Interpretable** and **Justifiable**: comprehensible, explainable and background justified
- **Operationally Efficient**
  - Machine learning systems are very complex and often fragile, don’t do ML first but data-driven first.
- **Economic Cost**: cost to acquire data
  - Free vs Expensive Data
- **Comply** to **Regulation** and **Legislation**
  - Ethical considerations

The combination of “right” data and data science techniques excels.

References

- https://christophm.github.io/interpretable-ml-book/
- https://lime-ml.readthedocs.io/en/latest/#
- Tim Miller. 2017. **Explanation in artificial intelligence: Insights from the social sciences**. arXiv Preprint arXiv:1706.07269.
- Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. 2016. **Examples are not enough, learn to criticize! Criticism for interpretability**. NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems, Pages 2288–2296.
- Kamal Mishra. 2020. **Model Explainability and JRT AI**. https://medium.com/@mishra.kamal/model-explainability-and-jrt-ai-b420531a0d49
- Christoph Molnar. 2020. **Interpretable Machine Learning: A Guide for Making Black Box Models Explainable**. *[online]* *https://christophm.github.io/interpretable-ml-book/*
- D. Sculley et al., “Hidden Technical Debt in Machine Learning Systems”