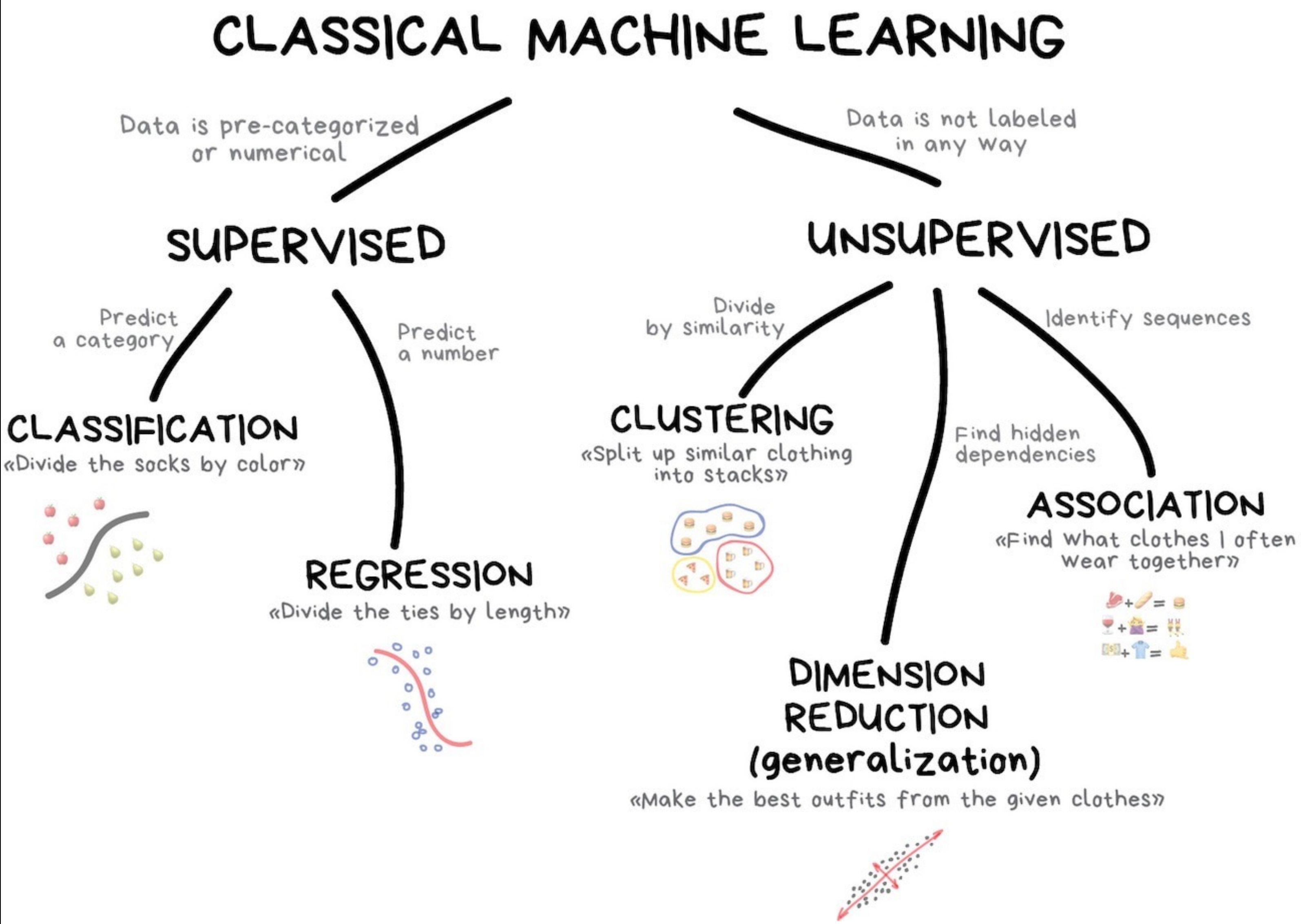
Machine Learning II

Carolina A. de Lima Salge
Assistant Professor
Terry College of Business
University of Georgia

Business Intelligence
Spring 2021



https://vas3k.com/blog/machine_learning/

Terry College of Business
UNIVERSITY OF GEORGIA

What Is Machine Learning (ML)?

No accepted definition but several are available:

- Field of study that gives computers the ability to learn without being explicitly programmed (Samuel 1959)

- **A computer program is said to learn** from experience E with respect to task T and some performance measure P, **if its performance** on T, as measured by P, **improves** with experience E (Mitchell 1998)

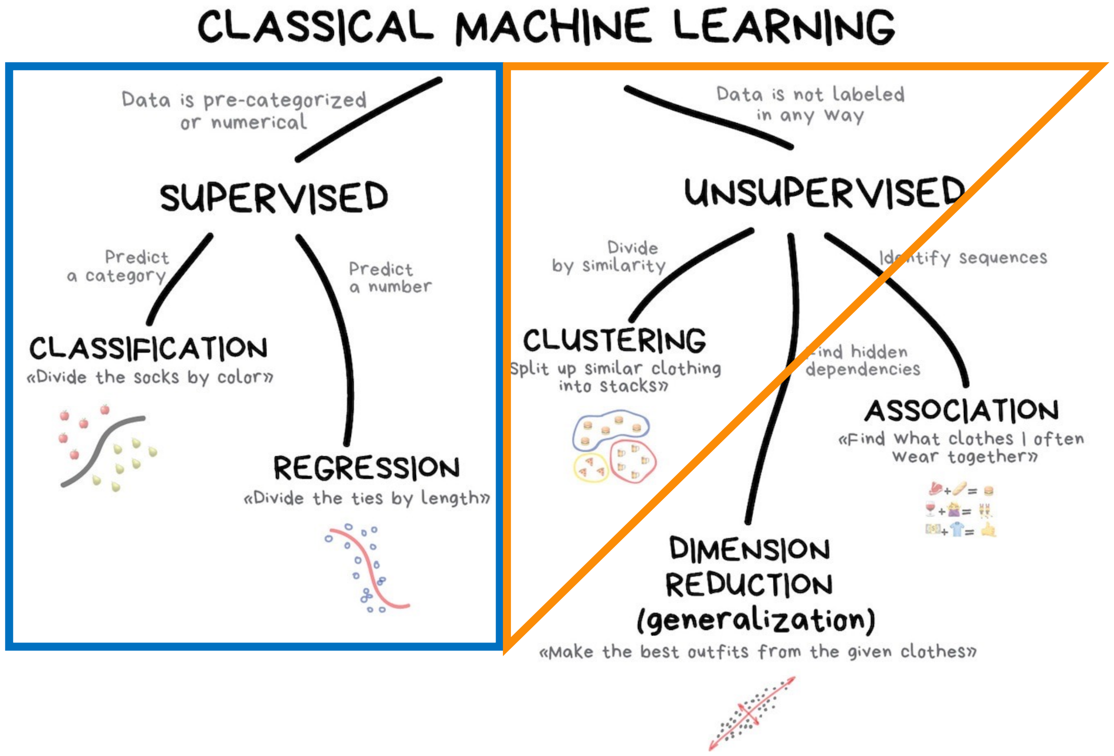
Two Types of ML

Supervised

- We teach the computer how to learn something
  - E.g., Which transactions are fraudulent?
  - A specific purpose
  - Requires specifying a target

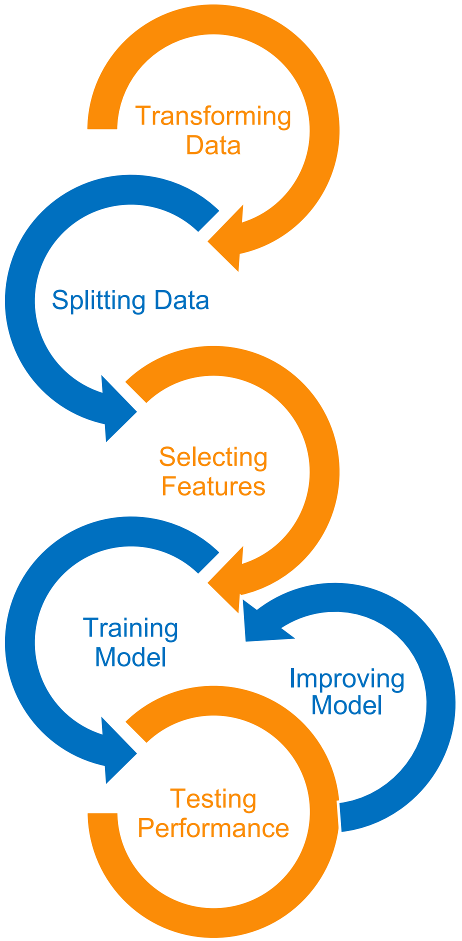
Unsupervised

- We let the computer learn by itself
  - E.g., Do our transactions naturally fall into different groups?
  - No specific purpose
  - No need to specify a target



Typical Supervised Learning Process

From variable character to numeric
Training and test sets
Manual or automated feature selection
Choosing an algorithm (LR, Logistic)
RMSE or Accuracy, Precision, Recall
Different algorithm, regularization



Objective of ML

1. Draw causal insights
   > "**What is causing** our customers to cancel their subscription to our services?"*
2. Predict future events
   > "**Which customers** are likely to cancel their subscription next month?"*
3. Understand patterns in data
   > "**Are there groups of customers** who are similar and use our services in a similar way?"*

Why ML?

Being able to **accurately and precisely** predict future events is **valuable**

MIT News
ON CAMPUS AND AROUND THE WORLD

SUBSCRIBE | BROWSE | SEARCH NEWS

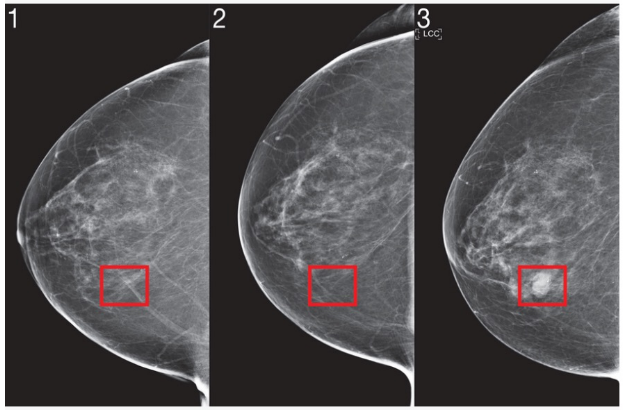
Robust artificial intelligence tools to predict future cancer

Researchers created a risk-assessment algorithm that shows consistent performance across datasets from US, Europe, and Asia.

Watch Video

Rachel Gordon | MIT CSAIL
January 28, 2021

PRESS INQUIRIES



MIT researchers have improved their machine learning system developed to predict cancer risk from mammogram images, and validated their effectiveness with studies across several hospitals.

Images courtesy of the researchers.

To catch cancer earlier, we need to predict who is going to get it in the future. The complex nature of forecasting risk has been bolstered by artificial intelligence (AI) tools, but the adoption of AI in medicine has been limited by poor performance on new patient populations and neglect to racial minorities.

Two years ago, a team of scientists from MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) and Jameel Clinic demonstrated a deep learning system to predict cancer risk using just a patient's mammogram. The model showed significant promise and even

SHARE

Paper: "Toward robust mammography-based models for breast cancer risk"

Business Requirements (SOA)

Business scope - fraud example

1. **Situation** - The fraud rate has started increasing
2. **Opportunity** - Reduce fraud rate by X %, resulting in Y USD savings
3. **Action** - Work on improving fraud detection system, reduce fraud drivers, and manually review transactions at risk

Business Requirements (SOA)

Business scope - churn example

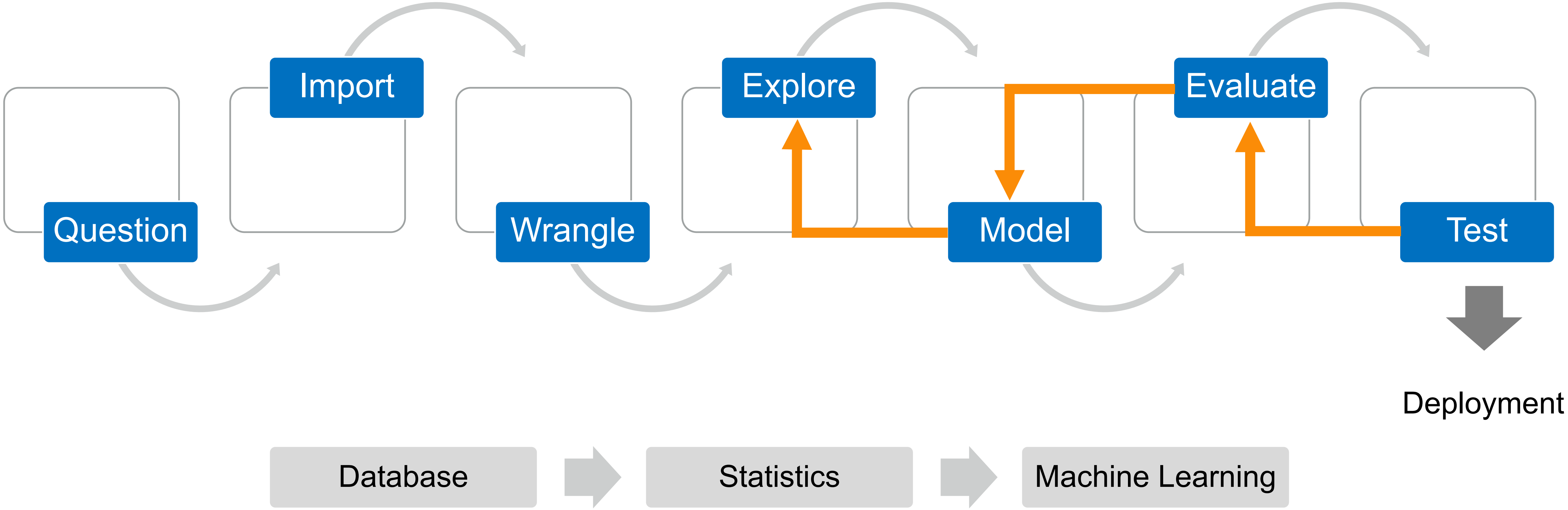
1. **Situation** - The customers started to churn more
2. **Opportunity** - Reduce churn rate by X %, resulting in Y USD revenue saved
3. **Action** - Work on identifying and improving churn drivers (website errors, too much/little advertising, customer service issues etc.); identify customers at risk and introduce retention campaigns

Things to Consider

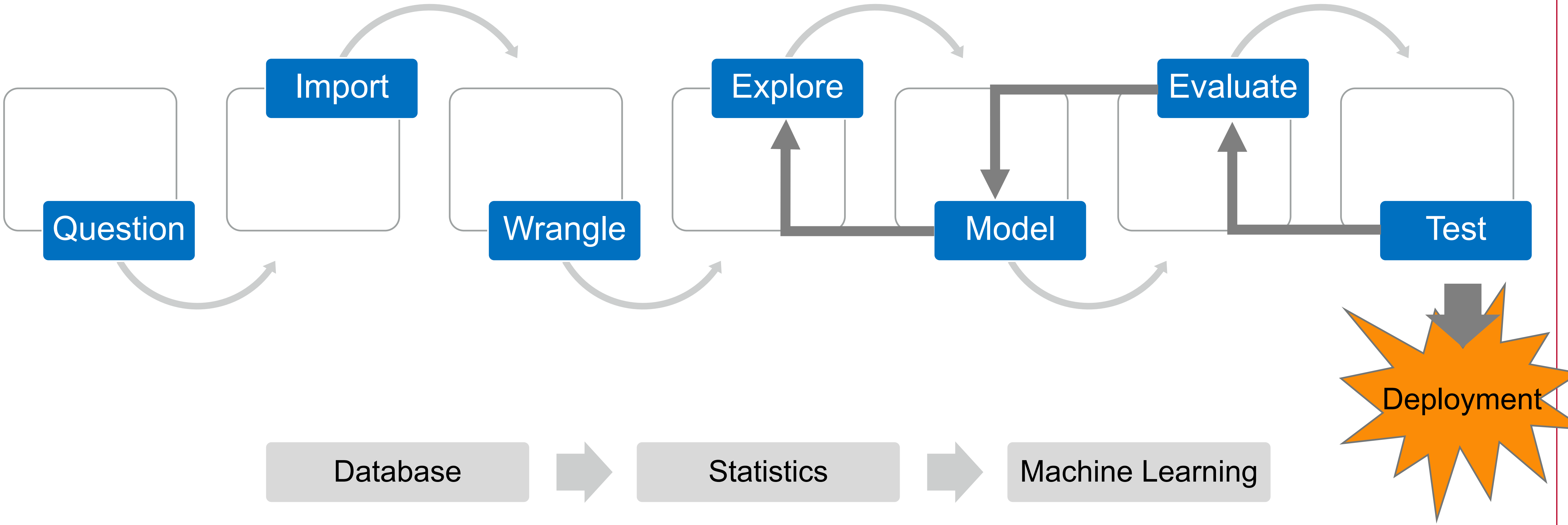
Defining the situation requires **asking the right question**, which can be difficult!

- Familiarize yourself with the context to identify a problem
- Engage in research to learn how others have handled the problem
- Start with causal question to address problem
- Define prediction question from causal question

Data Value Creation Model



Data Value Creation Model



Things to Consider

Even with great accuracy, the results might not be actionable – i.e., **you still may not be able to affect the predicted outcome**

- Look at historical data identify people susceptible to the problem
- Run experiments targeting such individuals with a particular treatment
- Repeat experiments to ensure results replicate

Example

Assume we have a great model to predict customer churn, how do we know whether such a model is **actionable**?

- Run randomized (field) experiments with 2 groups of customers predicted to churn – target one group (A) with an incentive (e.g., discounts) and do nothing to the other group (B), which is your control

Business scope - churn example

1. **Situation** - The customers started to churn more
2. **Opportunity** - Reduce churn rate by X %, resulting in Y USD revenue saved
3. **Action** - Work on identifying and improving churn drivers (website errors, too much/little advertising, customer service issues etc.); identify customers at risk and introduce retention campaigns

Example

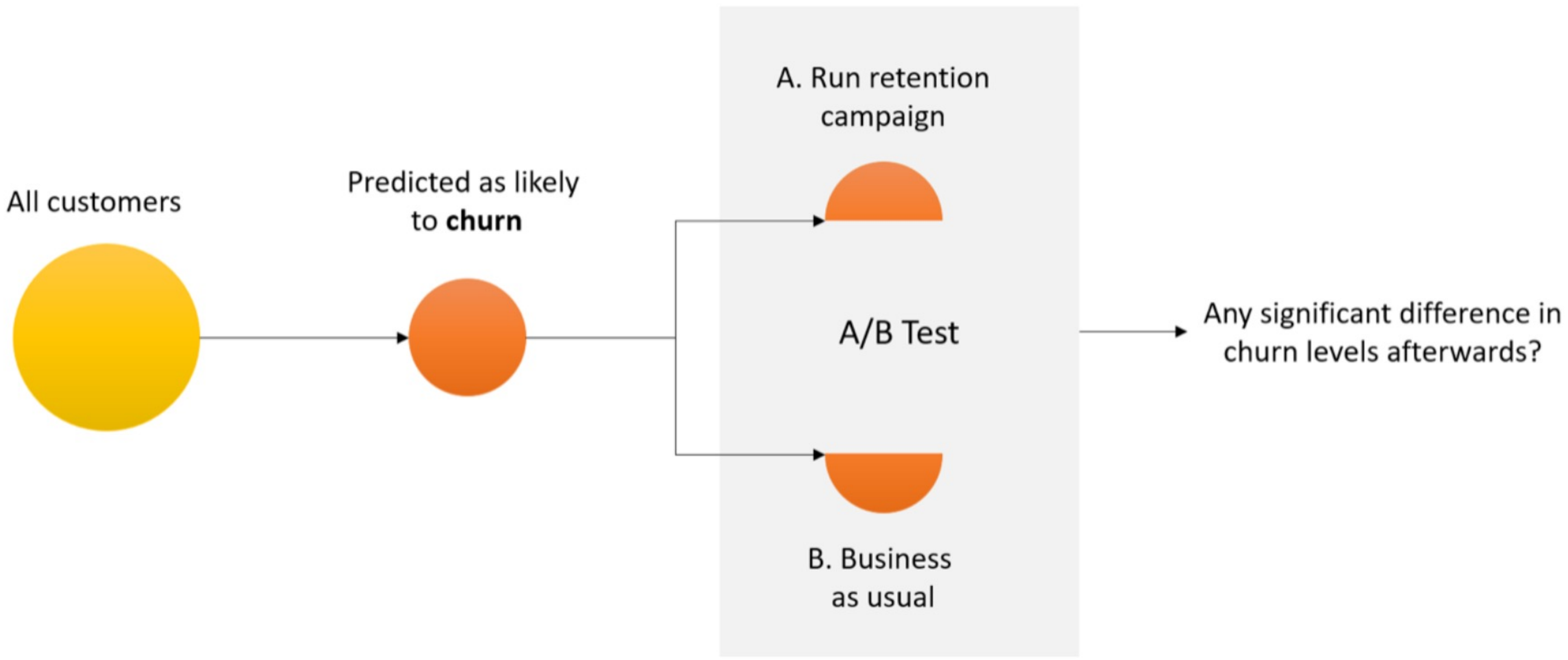
Assume we have a great model to predict customer churn, how do we know whether such a model is **actionable**?

- Do you find statistical differences in churn level between A and B across different experiments? If yes, is it in the expected direction? i.e., did offering discounts decrease churn? If yes, then your model is actionable

Business scope - churn example

1. **Situation** - The customers started to churn more
2. **Opportunity** - Reduce churn rate by X %, resulting in Y USD revenue saved
3. **Action** - Work on identifying and improving churn drivers (website errors, too much/little advertising, customer service issues etc.); identify customers at risk and introduce retention campaigns

Example


All customers
Predicted as likely to **churn**
A. Run retention campaign
A/B Test
B. Business as usual
Any significant difference in churn levels afterwards?

Things to Consider

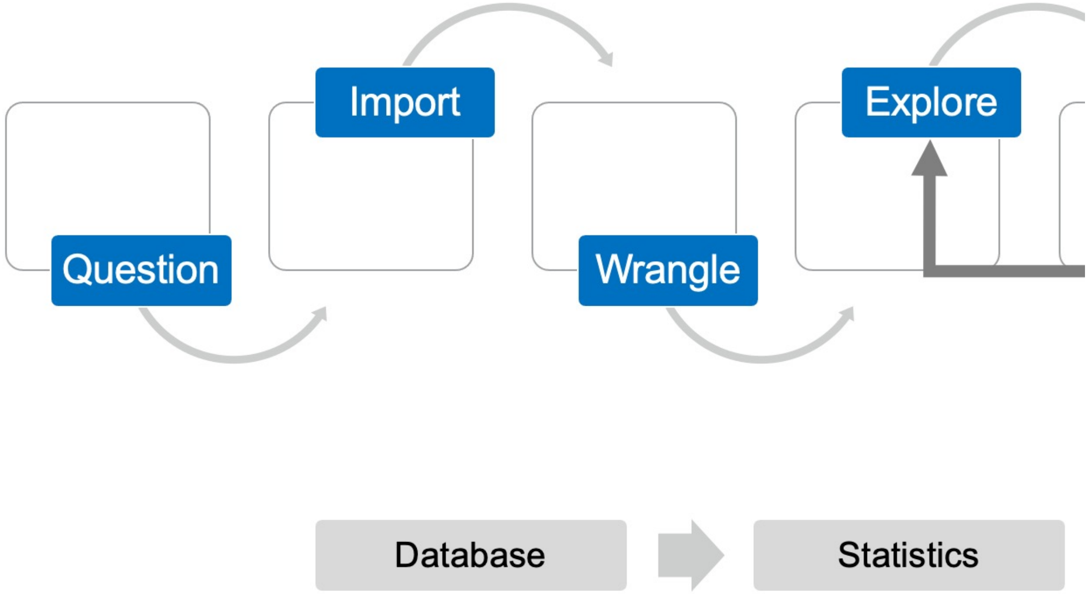
Do not lose sight of the opportunity – i.e., **do not spend more than what you can get**

- If your model is actionable, **estimate cost-benefit** of building automated process

- If your model is not actionable, consider collecting more data or engaging in qualitative research or narrowing down the scope of question or improving the model and testing again

https://campus.datacamp.com/courses/machine-learning-for-business/business-requirements-and-model-design?ex=1

Some Practical Advice for Moving Forward



Mistakes

- Machine learning first
- Not enough data
- Target variable definition
- Late testing, no impact
- Feature selection

Some Practical Advice for Moving Forward

7 How large do the dev/test sets need to be?

The dev set should be large enough to detect differences between algorithms that you are trying out. For example, if classifier A has an accuracy of 90.0% and classifier B has an accuracy of 90.1%, then a dev set of 100 examples would not be able to detect this 0.1% difference. Compared to other machine learning problems I've seen, a 100 example dev set is small. Dev sets with sizes from 1,000 to 10,000 examples are common. With 10,000 examples, you will have a good chance of detecting an improvement of 0.1%.[2]

Mistakes

- Machine learning first
- Not enough data
- Target variable definition
- Late testing, no impact
- Feature selection

Some Practical Advice for Moving Forward

What are you trying to predict? Need to be able to clearly define it

Can you observe that? If yes, how are you measuring it? If no, consider a different target

It is important that the definition of your target variable matches the measurement of the target variable – i.e., it is vital for you to predict what you are setting out to predict!

- e.g., adding items to cart is not equivalent to purchasing them but may reflect instead intention to purchase

Mistakes

- Machine learning first
- Not enough data
- $\boxed{\text{Target variable definition}}$
- Late testing, no impact
- Feature selection

https://campus.datacamp.com/courses/machine-learning-for-business/managing-machine-learning-projects?ex=1

Some Practical Advice for Moving Forward

You may spend too much time on trying to build the perfect model or the best model possible

Don't overthink it! Rather, start off with some idea on how to build your model

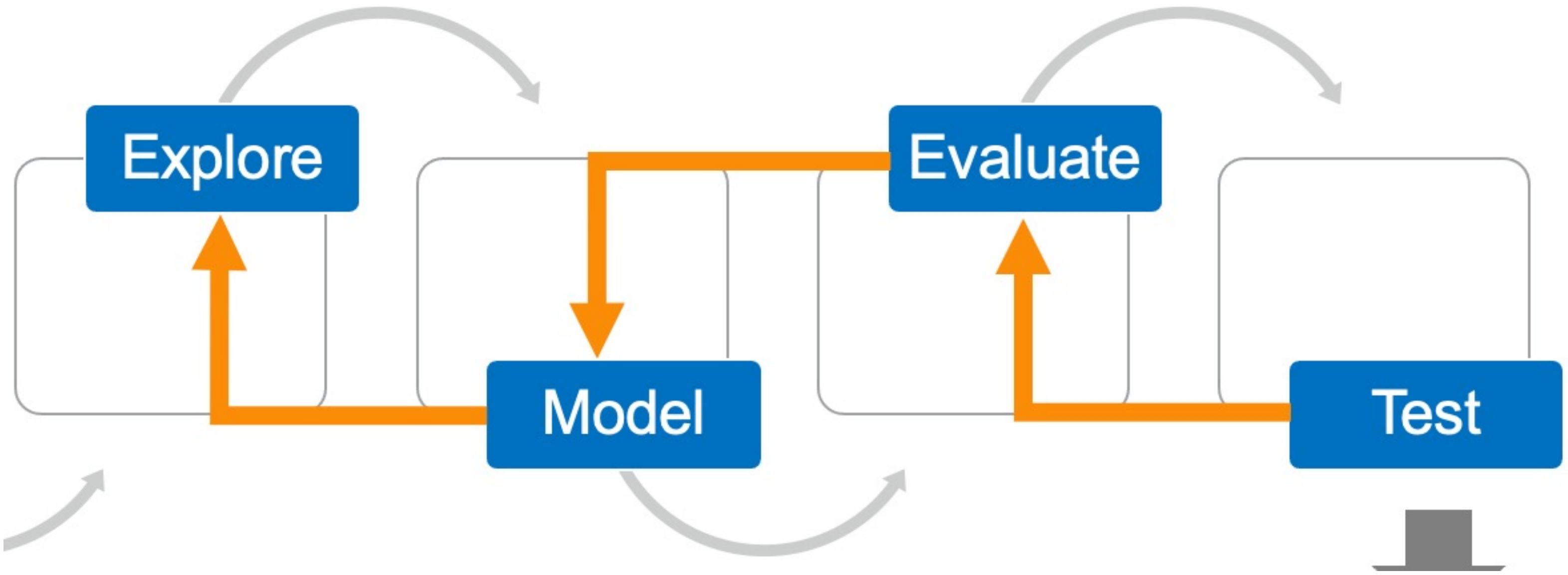
Implement the idea in code, then test the idea with experiments (if possible), use the learning to generate mode ideas and keep on iterating

Build your first model quickly, then iterate

Mistakes

- Machine learning first
- Not enough data
- Target variable definition
- Late testing, no impact
- Feature selection

Some Practical Advice for Moving Forward



Some Practical Advice for Moving Forward

Consider **starting with unsupervised learning** to explore the data (e.g., cluster analysis), or get it to a manageable size (e.g., PCA), then use the results from unsupervised learning to help predict something

Engage in **error analysis** – i.e., look at the classifications or estimated values that your model got wrong manually; do you see a pattern?

If your task is of classification, **compare results to human-level performance**

- Easy to get humans to label
- Draw on human intuition to improve model
- Use human-level performance to set a desired error rate

Ng, A. (2016). Machine Learning Yearning.

Some Practical Advice for Moving Forward

8 Establish a single-number evaluation metric for your team to optimize

Classification accuracy is an example of a **single-number evaluation metric**: You run your classifier on the dev set (or test set), and get back a single number about what fraction of examples it classified correctly. According to this metric, if classifier A obtains 97% accuracy, and classifier B obtains 90% accuracy, then we judge classifier A to be superior.

"If you really care about both **Precision and Recall**, I recommend using one of the standard ways to combine them into a single number. For example, one could take the average of precision and recall, to end up with a single number. Alternatively, you can compute the "**F1 score**," which is a modified way of computing their average, and works better than simply taking the mean"

Having a single-number evaluation metric speeds up your ability to make a decision when you are selecting among a large number of classifiers. It gives a clear preference ranking among all of them, and therefore a clear direction for progress.

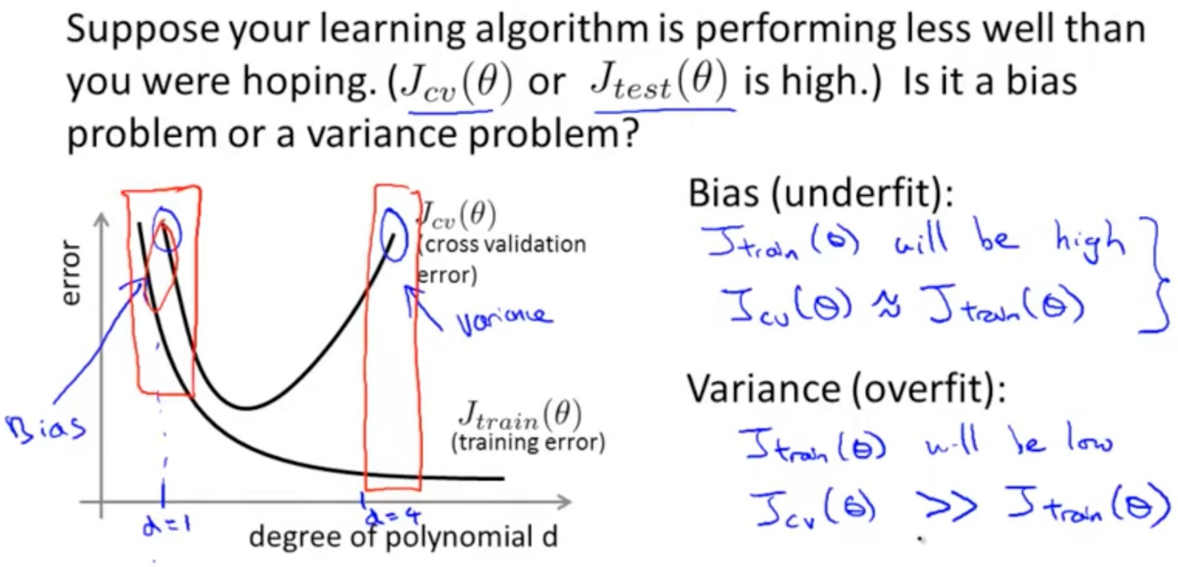
Ng, A. (2016). Machine Learning Yearning.

Some Practical Advice for Moving Forward

Diagnose the nature of the error

If the amount of error is high in both training and test sets, then you have a problem of **underfitting**

If the amount of error is high for the test set, but low for the training set, then you have a problem of **overfitting**



Some Practical Advice for Moving Forward

Diagnose the nature of the error

Test Set

```
> # how did we do with test set? confusion matrix
> confusionMatrix(data = churn_test$pred_churn,
+                 reference = churn_test$Churn,
+                 mode = "prec_recall",
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  925 176
       Yes 107 197

               Accuracy : 0.7986
                 95% CI : (0.7766, 0.8193)
    No Information Rate : 0.7345
    P-Value [Acc > NIR] : 1.342e-08

                  Kappa : 0.4511

 Mcnemar's Test P-Value : 5.296e-05

              Precision : 0.6480
                 Recall : 0.5282
                     F1 : 0.5820
             Prevalence : 0.2655
         Detection Rate : 0.1402
   Detection Prevalence : 0.2164
      Balanced Accuracy : 0.7122

       'Positive' Class : Yes
```

Training Set

```
> # how did we do with training set? confusion matrix
> confusionMatrix(data = churn_train$pred_churn,
+                 reference = churn_train$Churn,
+                 mode = "prec_recall",
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  3697  693
       Yes  434  803

               Accuracy : 0.7997
                 95% CI : (0.789, 0.8101)
    No Information Rate : 0.7341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4569

 Mcnemar's Test P-Value : 1.527e-14

              Precision : 0.6492
                 Recall : 0.5368
                     F1 : 0.5876
             Prevalence : 0.2659
         Detection Rate : 0.1427
   Detection Prevalence : 0.2198
      Balanced Accuracy : 0.7159

       'Positive' Class : Yes
```

```r
model <- train(Churn ~ InternetServiceN + PaperlessN + SeniorCitizen +
                 PartnerN + TechSupportN + DependentsN + OnlineSecurityN +
                 PaymentN + tenure + ContractN,
               data = churn_train, # use training set
               method = "glm") # simple additive logistic regression

# now predict outcomes in test set
p <- predict(model, churn_test, type = 'raw')

# also do it in the training set to check for underfitting / overfitting
pTr <- predict(model, churn_train, type = 'raw')

# add predictions to initial dataset
# test set
churn_test$pred_churn <- p
# training set
churn_train$pred_churn <- pTr

# how did we do with test set? confusion matrix
confusionMatrix(data = churn_test$pred_churn,
                reference = churn_test$Churn,
                mode = "prec_recall",
                positive = "Yes")

# how did we do with training set? confusion matrix
confusionMatrix(data = churn_train$pred_churn,
                reference = churn_train$Churn,
                mode = "prec_recall",
                positive = "Yes")
```

Some Practical Advice for Moving Forward

Diagnose the nature of the error

Test Set

```
> # how did we do with test set? confusion matrix
> confusionMatrix(data = churn_test$pred_churn,
+                 reference = churn_test$Churn,
+                 mode = "prec_recall",
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  906 186
       Yes 126 187

               Accuracy : 0.7779
                 95% CI : (0.7553, 0.7994)
    No Information Rate : 0.7345
    P-Value [Acc > NIR] : 9.989e-05

                  Kappa : 0.3998

 Mcnemar's Test P-Value : 0.0008371

              Precision : 0.5974
                 Recall : 0.5013
                     F1 : 0.5452
             Prevalence : 0.2655
         Detection Rate : 0.1331
   Detection Prevalence : 0.2228
      Balanced Accuracy : 0.6896

       'Positive' Class : Yes
```

Training Set

```
> # how did we do with training set? confusion matrix
> confusionMatrix(data = churn_train$pred_churn,
+                 reference = churn_train$Churn,
+                 mode = "prec_recall",
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  3727  650
       Yes  404  846

               Accuracy : 0.8127
                 95% CI : (0.8022, 0.8228)
    No Information Rate : 0.7341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4936

 Mcnemar's Test P-Value : 4.471e-14

              Precision : 0.6768
                 Recall : 0.5655
                     F1 : 0.6162
             Prevalence : 0.2659
         Detection Rate : 0.1503
   Detection Prevalence : 0.2221
      Balanced Accuracy : 0.7339

       'Positive' Class : Yes
```

```r
model <- train(Churn ~ InternetServiceN + PaperlessN + SeniorCitizen +
                 PartnerN + TechSupportN + DependentsN + OnlineSecurityN +
                 PaymentN + tenure + ContractN,
               data = churn_train, # use training set
               method = "knn") # k-Nearest neighbors

# now predict outcomes in test set
p <- predict(model, churn_test, type = 'raw')

# also do it in the training set to check for underfitting / overfitting
pTr <- predict(model, churn_train, type = 'raw')

# add predictions to initial dataset
# test set
churn_test$pred_churn <- p
# training set
churn_train$pred_churn <- pTr

# how did we do with test set? confusion matrix
confusionMatrix(data = churn_test$pred_churn,
                reference = churn_test$Churn,
                mode = "prec_recall",
                positive = "Yes")

# how did we do with training set? confusion matrix
confusionMatrix(data = churn_train$pred_churn,
                reference = churn_train$Churn,
                mode = "prec_recall",
                positive = "Yes")
```

Some Practical Advice for Moving Forward

To address **underfitting**, you might try the following:

- Increase model complexity (e.g., from linear to polynomial regression)
- Remove regularization

To address **overfitting**:

- Add more training data
- Decrease model complexity
- Add regularization (e.g., Lasso or Ridge)
- Significantly reduce the number of features

Modify input features based on insights from error analysis - **can help with both**

- In theory, adding more features could lead to overfitting; but if that is the case, then use regularization

Ng, A. (2016). Machine Learning Yearning.

Summary

Although predicting future events can be super valuable, this does not mean that ALL machine learning (ML) models should be deployed! Great models are not always actionable, and even when they are actionable, the opportunity might just not be there

Randomized experiments are very useful for testing how actionable ML models are, and for helping to estimate the costs-benefits of deploying an automated system

Summary

Underfitting and overfitting are problems that can harm the performance of your model and should be addressed differently

Avoiding key mistakes, such as mismatch between definition and operationalization of the target variable and late testing, is crucial for deriving value from ML

Error analysis can inform your understanding of the phenomenon and help to improve your models

Thank You!