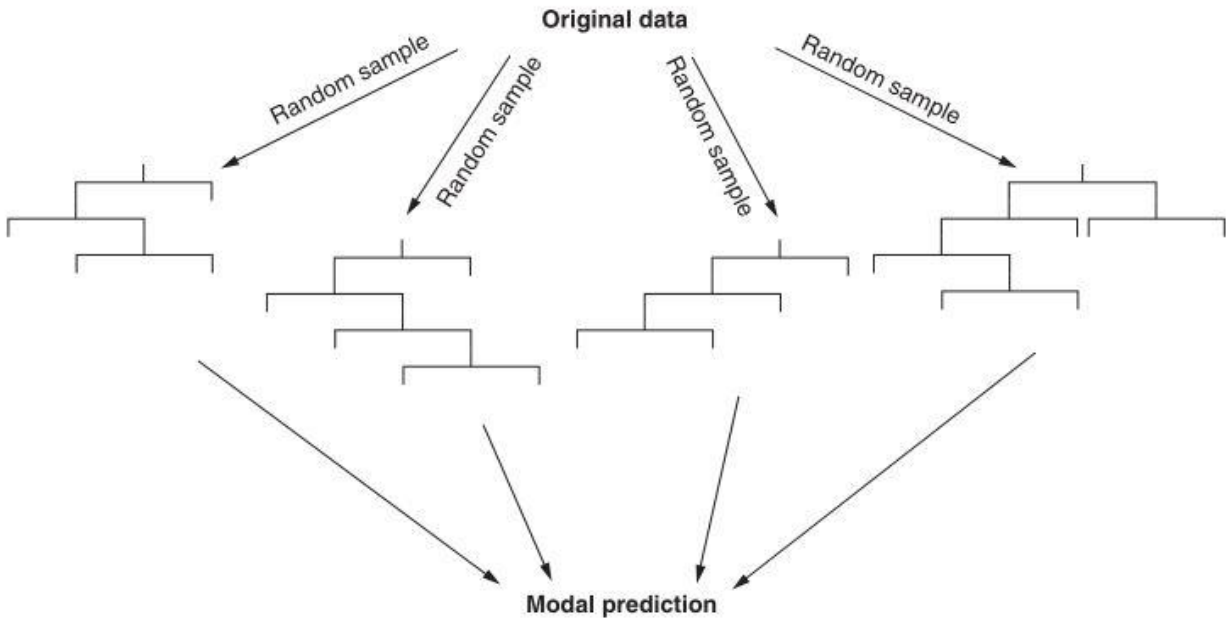
Classificação do uso e cobertura da terra utilizando o algoritmo Random Forest na Plataforma Google Earth Engine

Eduardo Ribeiro Lacerda

Fluminense Federal University (UFF)
International Institute for Sustainability (IIS)

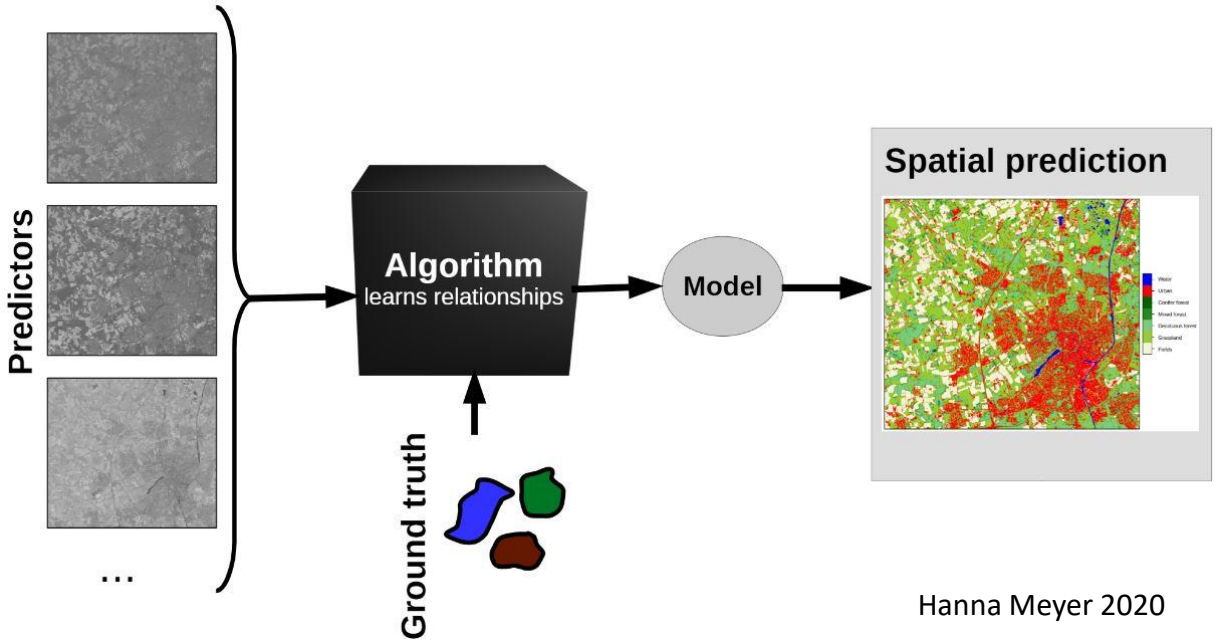
Sobre o que vamos falar?



O que vamos aprender?

- Como trabalhar com dados espaciais no Google Earth Engine
- Como coletar boas amostras para o seu modelo
- Como limpar e preparar os seus dados
- Como aplicar algoritmos como o Random Forest para classificar imagens de satélite
- Como melhorar seu modelo
- Como validar o modelo e seus resultados
- Como visualizer e exporter seus dados

A ideia principal é...



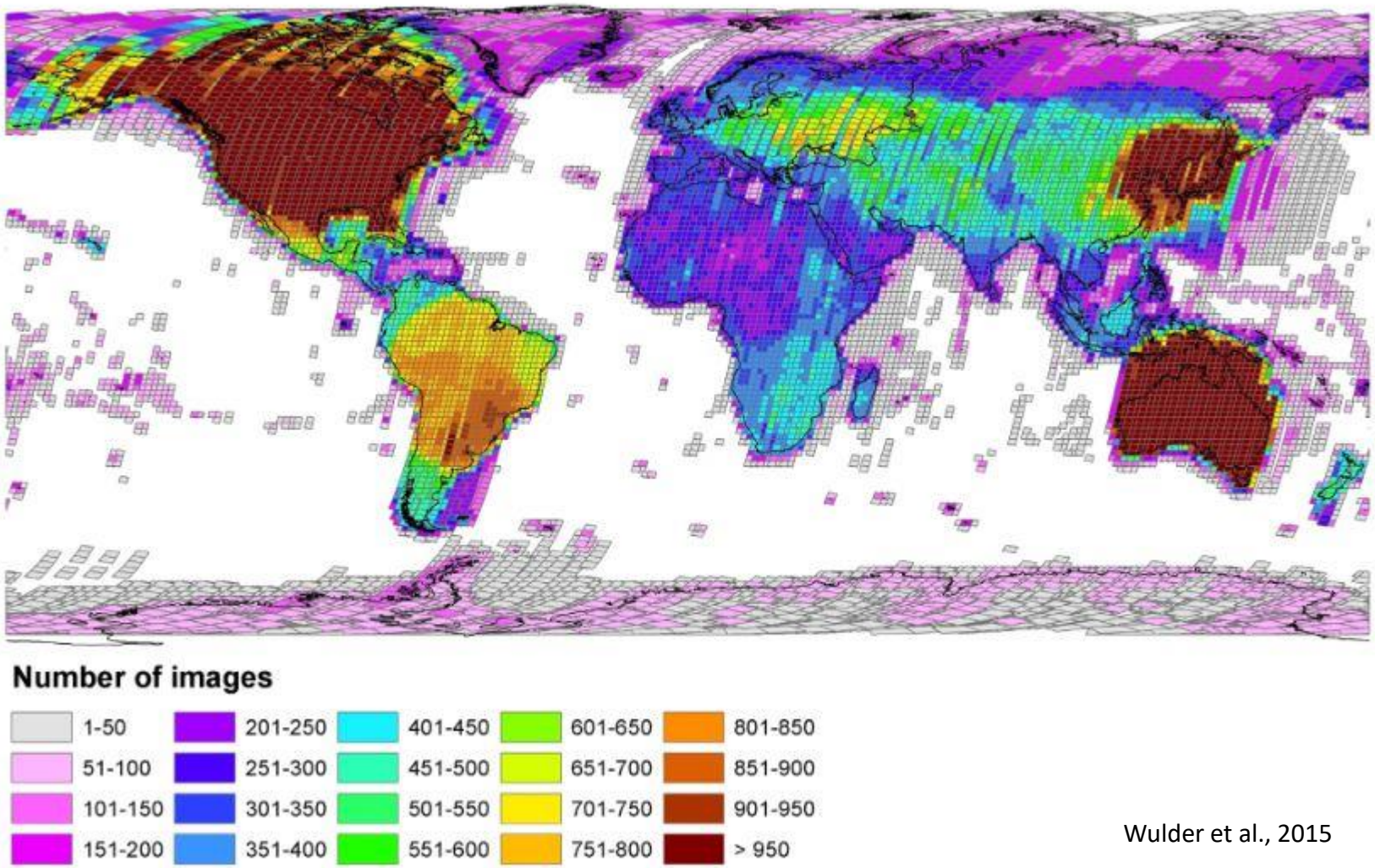
Hanna Meyer 2020

- Primeiro, selecionar imagens de satélite que sirvam como bons dados de entrada pro seu modelo
- Coletar boas amostras
- Treinar o modelo e fazer testes que melhorem seu desempenho
- Usar o modelo para classificar dados raster e criar bons resultados

Exemplos de dados de entrada

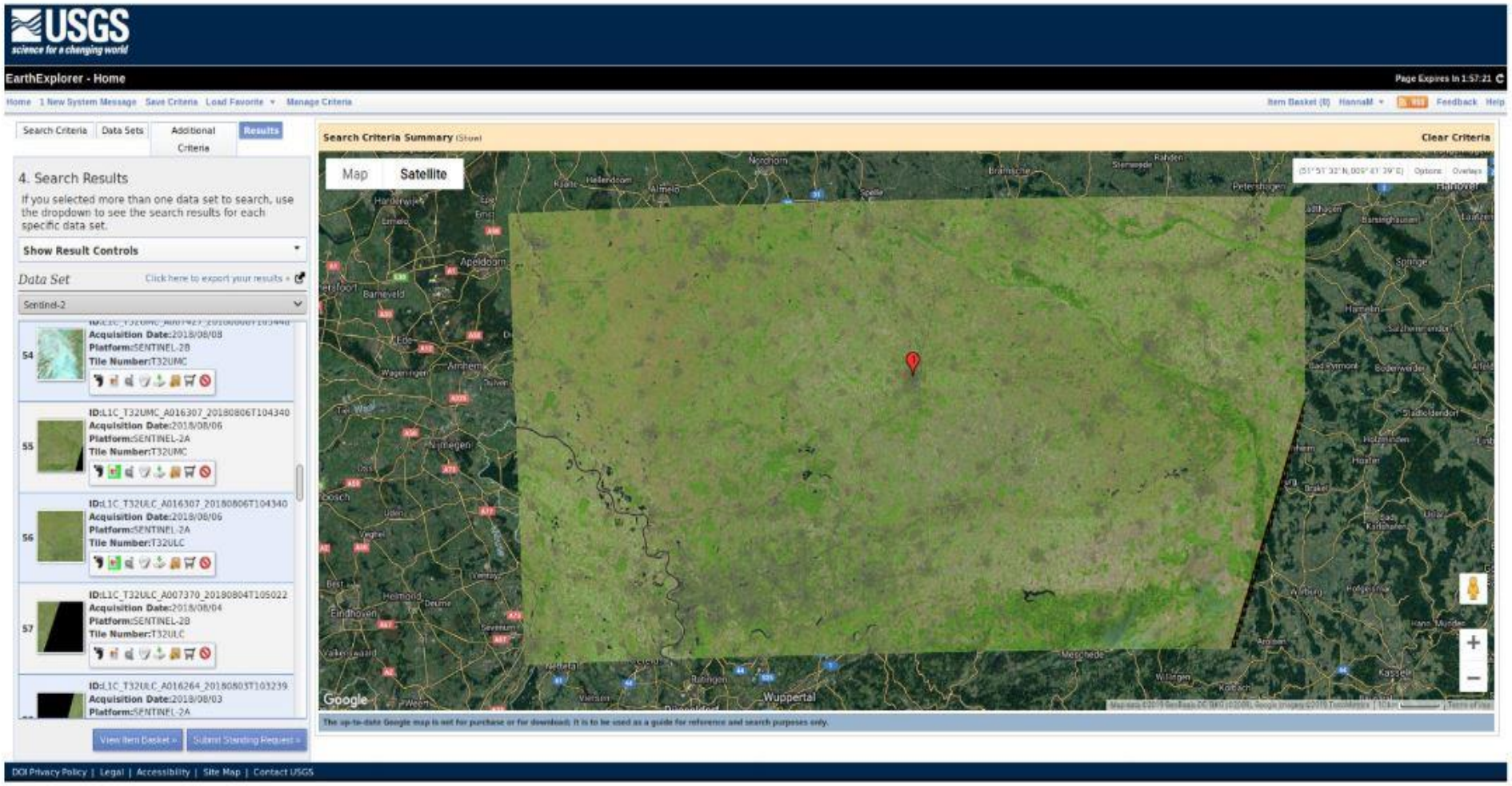
Platform/Sensor	Spatial Resolution	Temporal Resolution	Availability
Landsat MSS	79	16 days	started in 1972
Landsat TM	30	16 days	started in 1982
Landsat ETM+	30	16 days	started in 1999
Landsat 8 OLI	30	16 days	started in 2013
Landsat 9 OLI-2	30	16 days	mid 2021
Sentinel 2	10-20	5/10 days	started in 2014
MODIS	250-1000	4 per day	started in 2000

Acessibilidade do sensor Landsat OLI (Landsat 8)

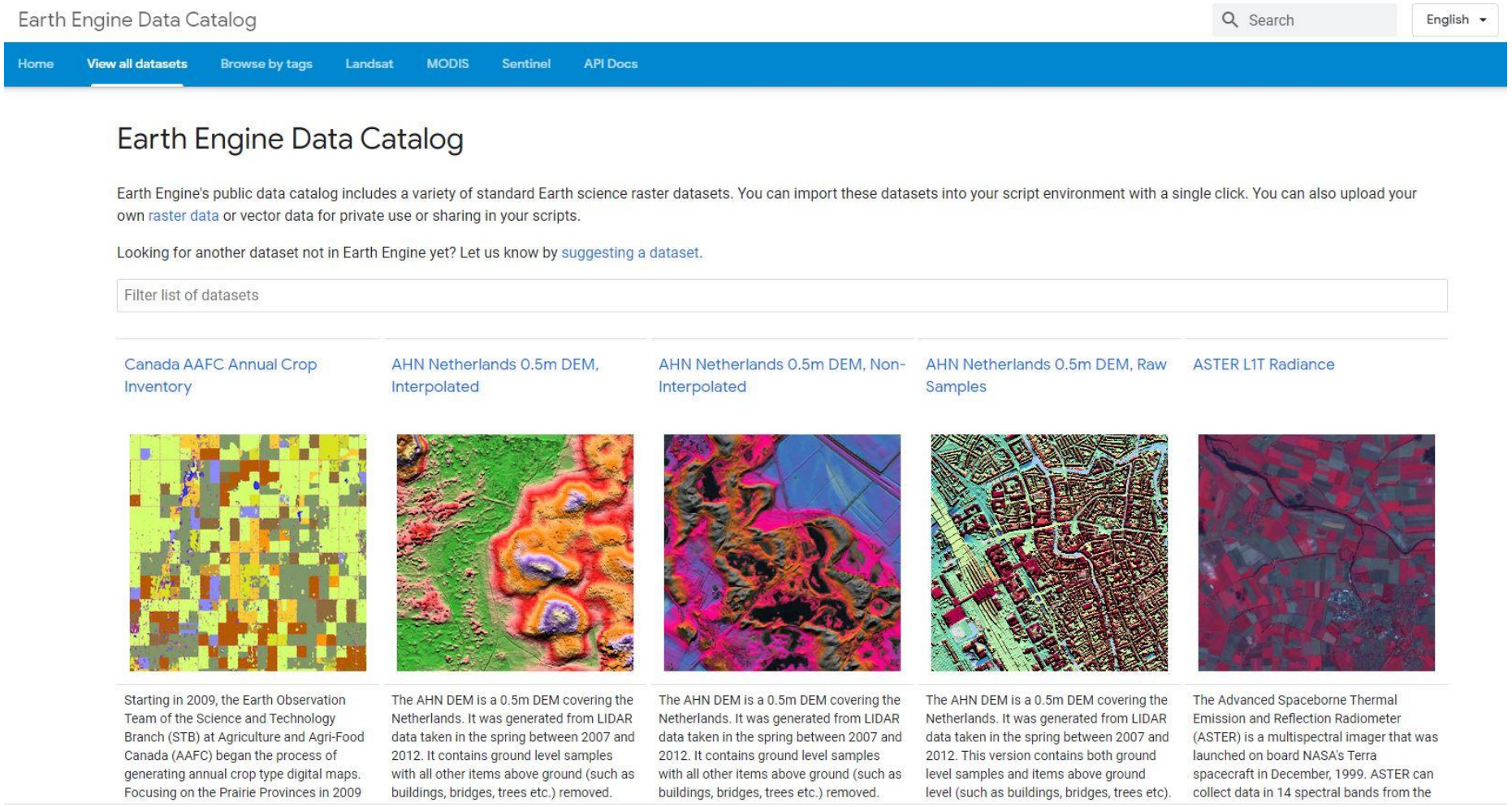


Wulder et al., 2015

Como ter acesso?



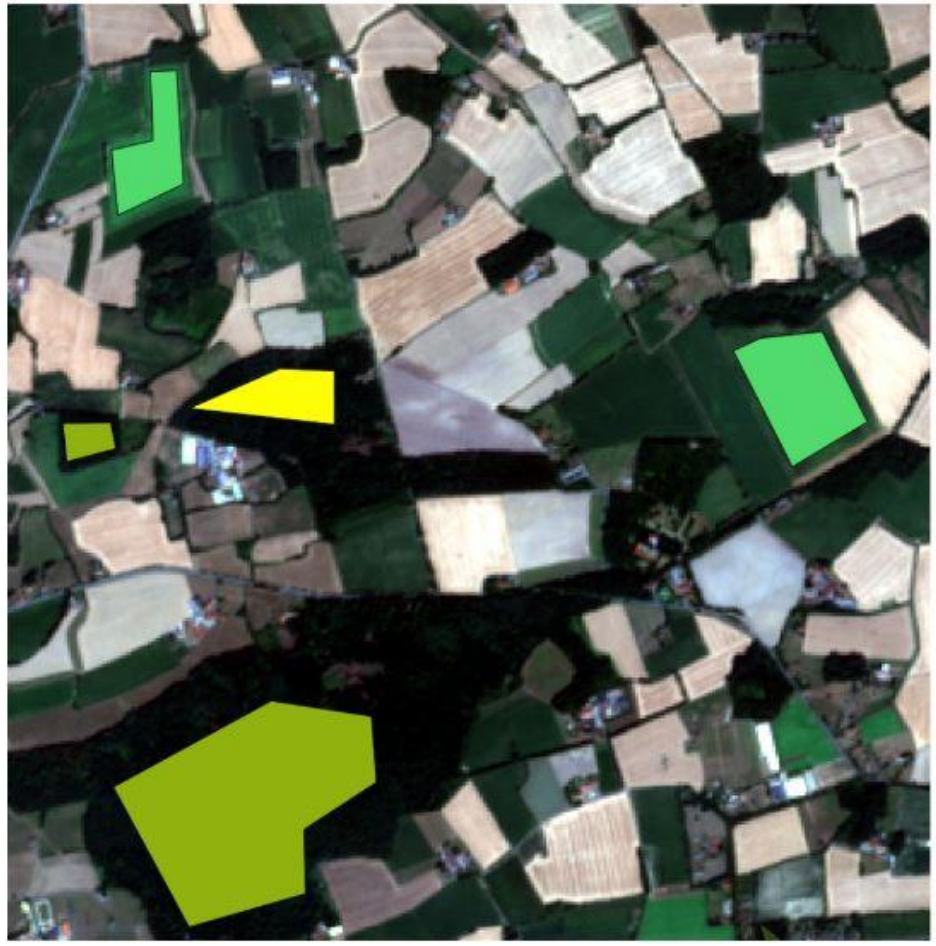
Usando a base de dados do Google Earth Engine



Fiz o download da imagem. E agora?

Pontos

Polígonos



Hanna Meyer 2020

Escolhendo o algoritmo a ser usado

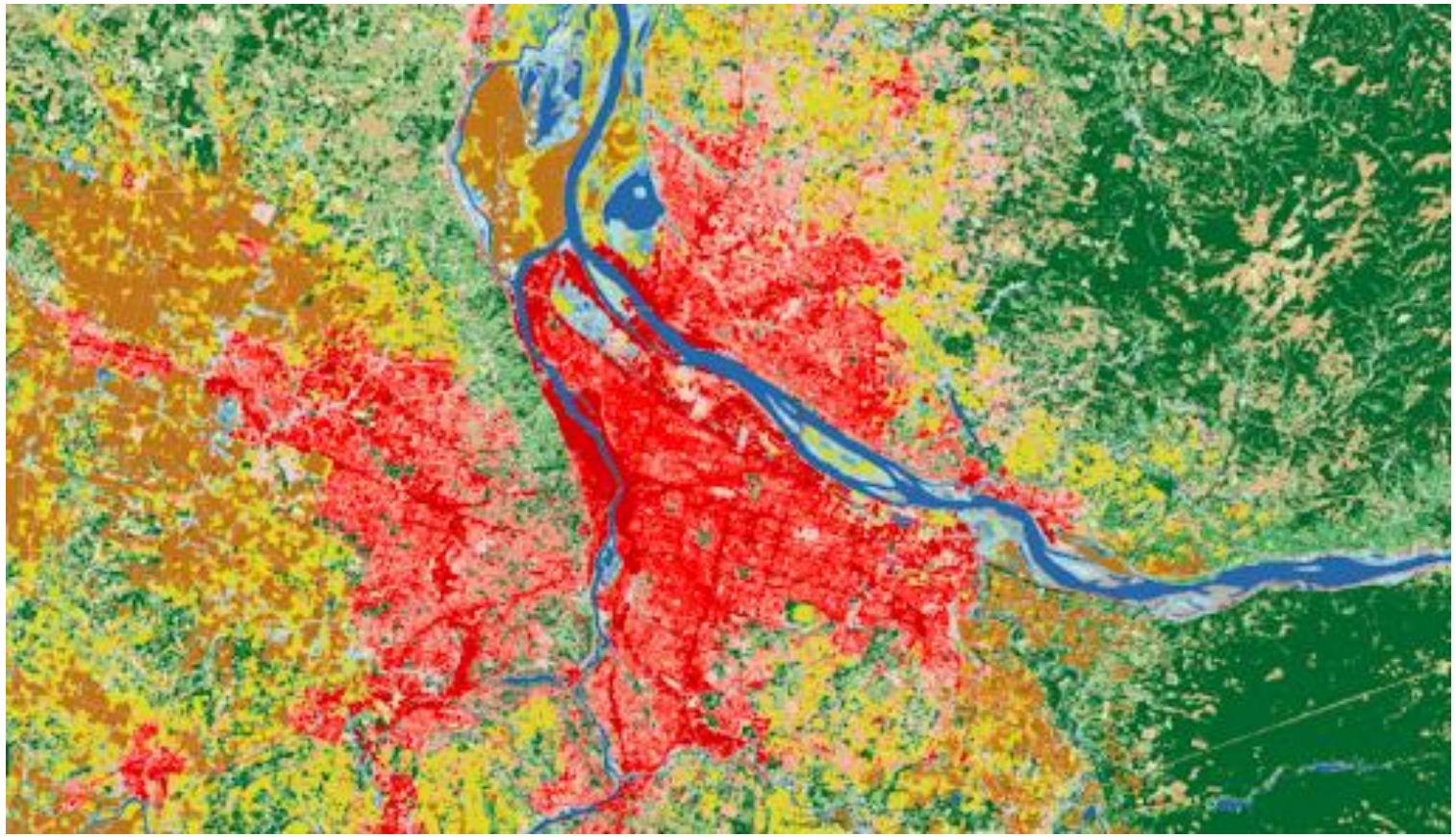
Existem muitas opções...

- **ee.Classifier**
  - ee.Classifier.cart(*crossvalidationFactor, maxDepth, minLeafPopula...*
  - ee.Classifier.decisionTree(treeString)
  - ee.Classifier.decisionTreeEnsemble(treeStrings)
  - ee.Classifier.gmoMaxEnt(*weight1, weight2, epsilon, minIterations, ...*
  - ee.Classifier.libsvm(*decisionProcedure, svmType, kernelType, shri...*
  - ee.Classifier.minimumDistance(*metric*)
  - ee.Classifier.naiveBayes(*lambda*)
  - ee.Classifier.randomForest(*numberOfTrees, variablesPerSplit, min...*
  - ee.Classifier.smileCart(*maxNodes, minLeafPopulation*)
  - ee.Classifier.smileNaiveBayes(*lambda*)
  - ee.Classifier.smileRandomForest(numberOfTrees, *variablesPerSpl...*
  - ee.Classifier.spectralRegion(coordinates, *schema*)
  - ee.Classifier.svm(*decisionProcedure, svmType, kernelType, shrinki...*

Filter

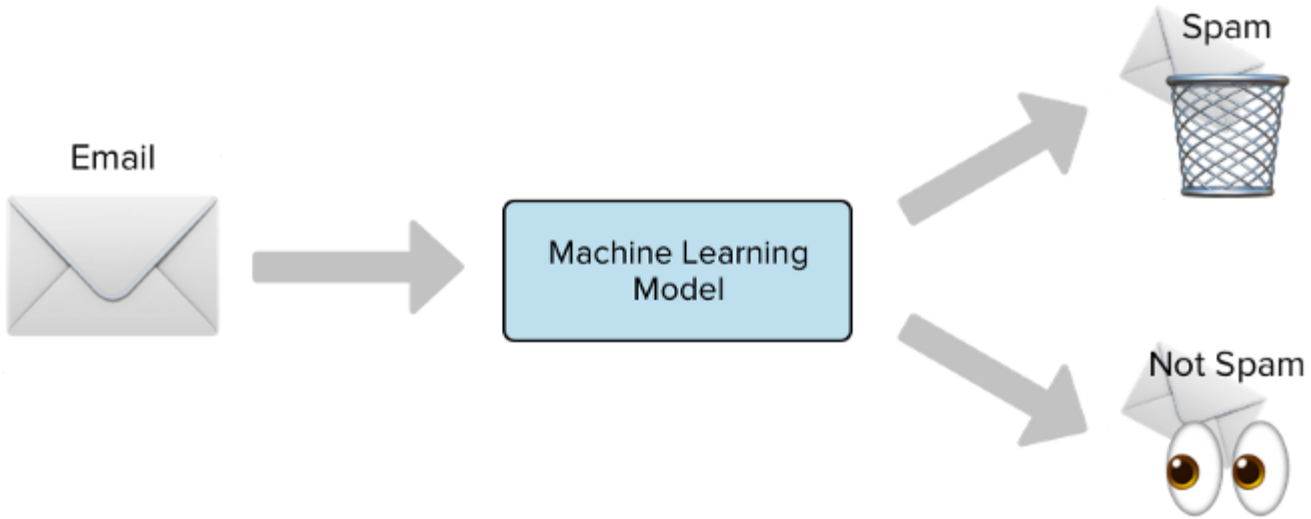
	class	name	short.name	package
1	classif.ada	ada Boosting	ada	ada,rpart
2	classif.adaboostm1	ada Boosting M1	adaboostm1	RWeka
3	classif.bartMachine	Bayesian Additive Regression Trees	bartmachine	bartMachine
4	classif.binomial	Binomial Regression	binomial	stats
5	classif.boosting	Adabag Boosting	adabag	adabag,rpart
6	classif.bst	Gradient Boosting	bst	bst,rpart
7	classif.C50	C50	C50	C50
8	classif.cforest	Random forest based on conditional inference trees	cforest	party
9	classif.clusterSVM	Clustered Support Vector Machines	clusterSVM	SwarmSVM,LiblineaR
10	classif.ctree	Conditional Inference Trees	ctree	party
11	classif.cvglmnet	GLM with Lasso or Elasticnet Regularization (Cross Validated...	cvglmnet	glmnet
12	classif.dbnDNN	Deep neural network with weights initialized by DBN	dbn.dnn	deepnet
13	classif.dcSVM	Divided-Conquer Support Vector Machines	dcSVM	SwarmSVM,e1071
14	classif.earth	Flexible Discriminant Analysis	fda	earth,stats
15	classif.evtree	Evolutionary learning of globally optimal trees	evtree	evtree
16	classif.extraTrees	Extremely Randomized Trees	extraTrees	extraTrees
17	classif.fdausc.glm	Generalized Linear Models classification on FDA	fdausc.glm	fda.usc
18	classif.fdausc.kernel	Kernel classification on FDA	fdausc.kernel	fda.usc
19	classif.fdausc.knn	fdausc.knn	fdausc.knn	fda.usc
20	classif.fdausc.np	Nonparametric classification on FDA	fdausc.np	fda.usc
21	classif.featureless	Featureless classifier	featureless	mlr
22	classif.fnn	Fast k-Nearest Neighbour	fnn	FNN
23	classif.gamboost	Gradient boosting with smooth components	gamboost	mboost
24	classif.gaterSVM	Mixture of SVMs with Neural Network Gater Function	gaterSVM	SwarmSVM

Dado de saída



- Urbano
- Floresta
- Solo Exposto
- Gramíneas
- Pasto
- Outros

Bias-variance trade-off



Bias-variance trade-off

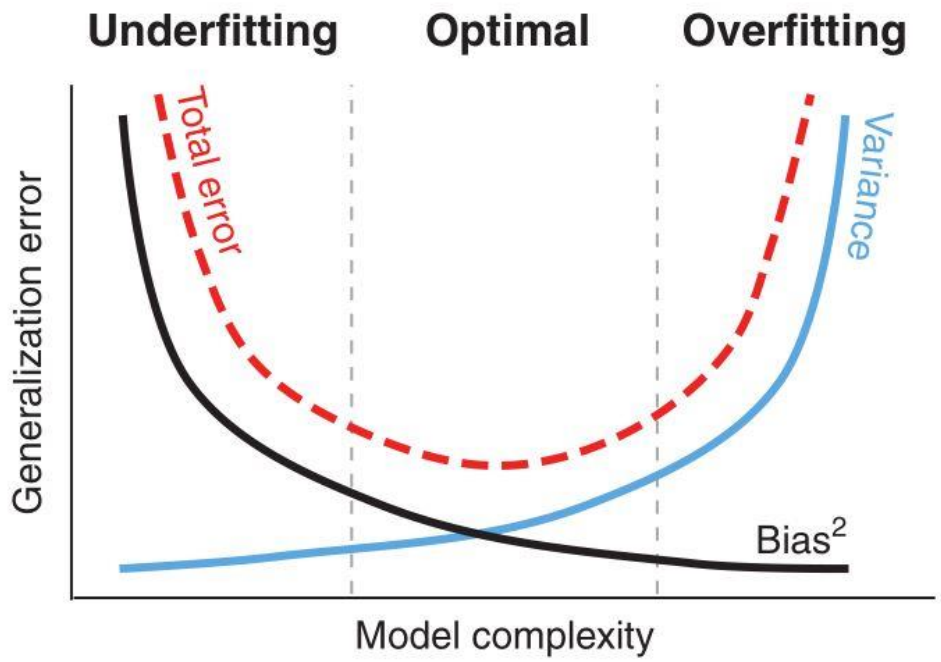
Underfitting e **Overfitting** são duas fontes de erro na construção de modelos.

Underfitted (subajuste) – Acontece quando seu modelo é simples demais e pode acabar gerando erros na classificação de algumas classes. Um modelo com essa caracteristica terá uma performance ruim quando for usado para classificar novos dados.

Overfitted (sobreajuste) – O modelo é complex demais e está considerando ruídos no dado que você utilizou para treinar o modelo. Um modelo com esta característica irá ter performance ruim quando for usado para classificar novos dados, mas se sairá muito bem classificando os dados que foram usados para treinar o modelo.

Ryns, 2020

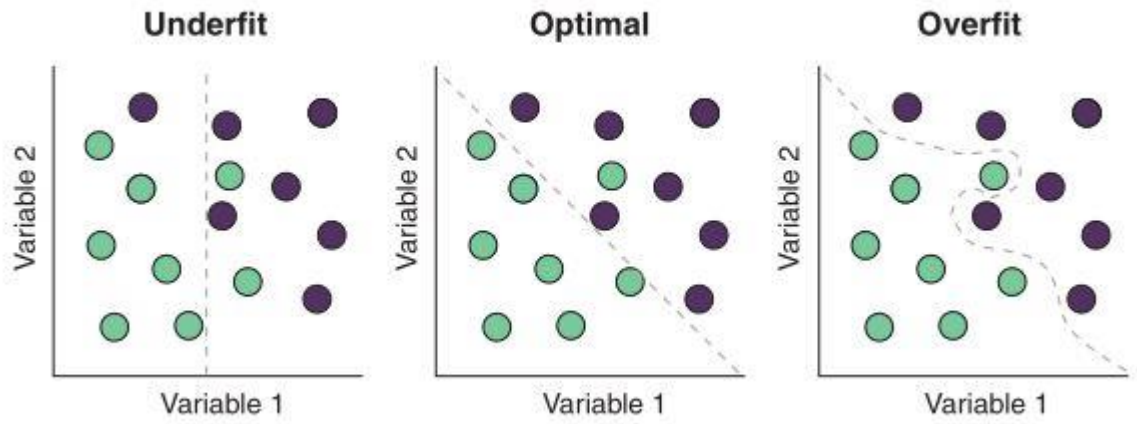
Bias-variance trade-off



- O “generalization error” é a proporção de previsões erradas que um modelo faz quando influenciado tanto em situações de subajuste (underfitting) como sobreajuste (overfitting).
- O erro associado ao sobreajuste acontece quando o modelo é complexo demais. Ou seja, quando utilizamos variáveis demais para treinar o modelo.
- O erro associado ao subajuste acontece quando o modelo é simples demais. Ou seja, quando utilizamos variáveis de menos para treinar o modelo.
- Um modelo ideal (optimal/balanced) equilibra esse trade-off.

Ryns, 2020

Bias-variance trade-off



- A linha pontilhada representa o limite de decisão do modelo
- Então, como podemos resolver este problema? A resposta está em utilizar uma técnica chamada cross-validation (validação cruzada).

Ryns, 2020

Cross-validation (Validação Cruzada)

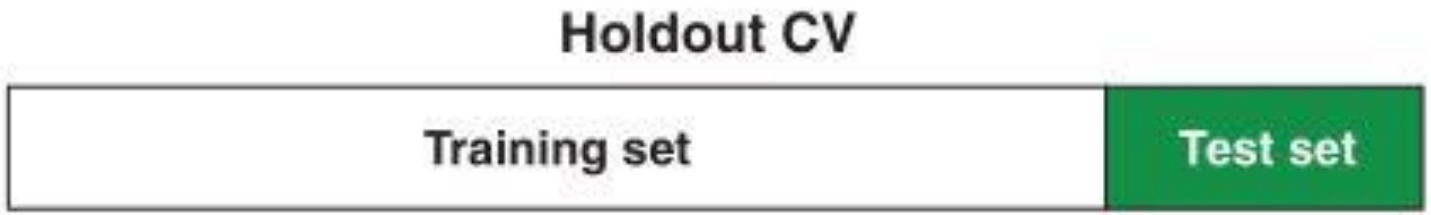
A solução é avaliar o desempenho do nosso modelo usando dados que o modelo ainda não viu! Podemos realizar uma nova coleta de dados, mas também podemos apenas divider o conjunto de dados entre amostras de treinamento e testes.

Fazendo isso, nós Podemos usar algumas métricas de desempenho para mostrar como o nosso modelo se irá se comportar com um novo conjunto de dados.

Tipos de cross-validation (validação cruzada)

- Holdout cross-validation
- K-fold cross-validation
- Leave-one-out cross-validation

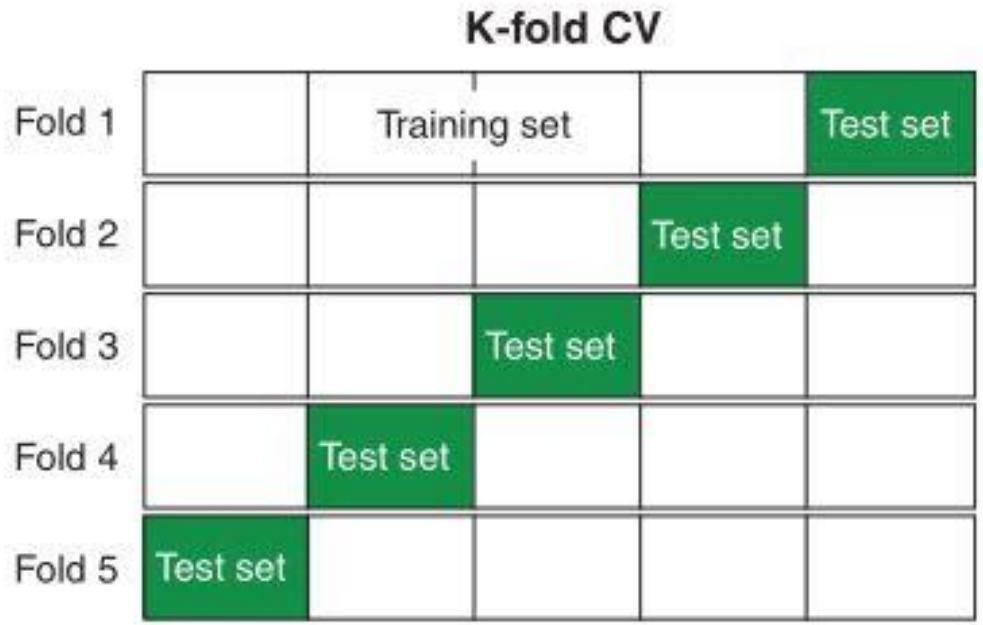
Cross-validation (Validação Cruzada)



1. The data is randomly split into a training and test set.
2. A model is trained using only the training set.
3. Predictions are made on the test set.
4. The predictions are compared to the true values.

Ryns, 2020

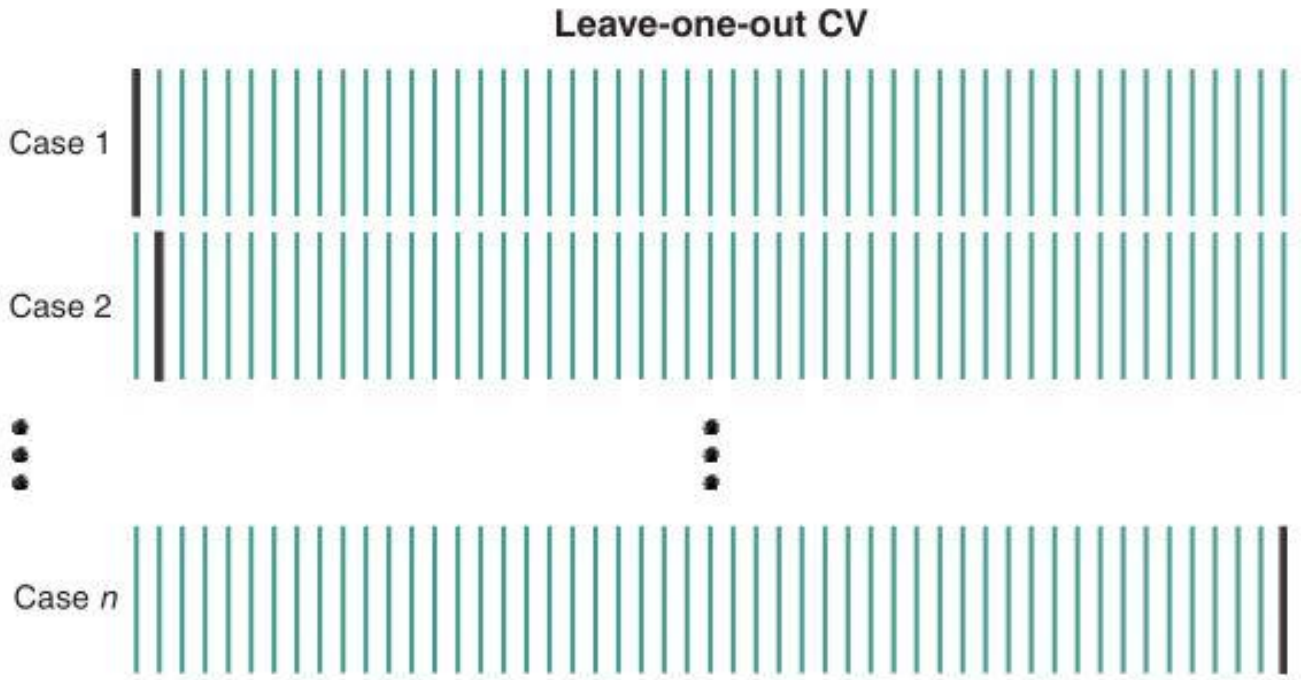
Cross-validation (Validação Cruzada)



1. The data is randomly split into *k* equal-sized folds.
2. Each fold is used as the test set once, where the rest of the data makes the training set.
3. For each fold, predictions are made on the test set.
4. The predictions are compared to the true values.

Ryns, 2020

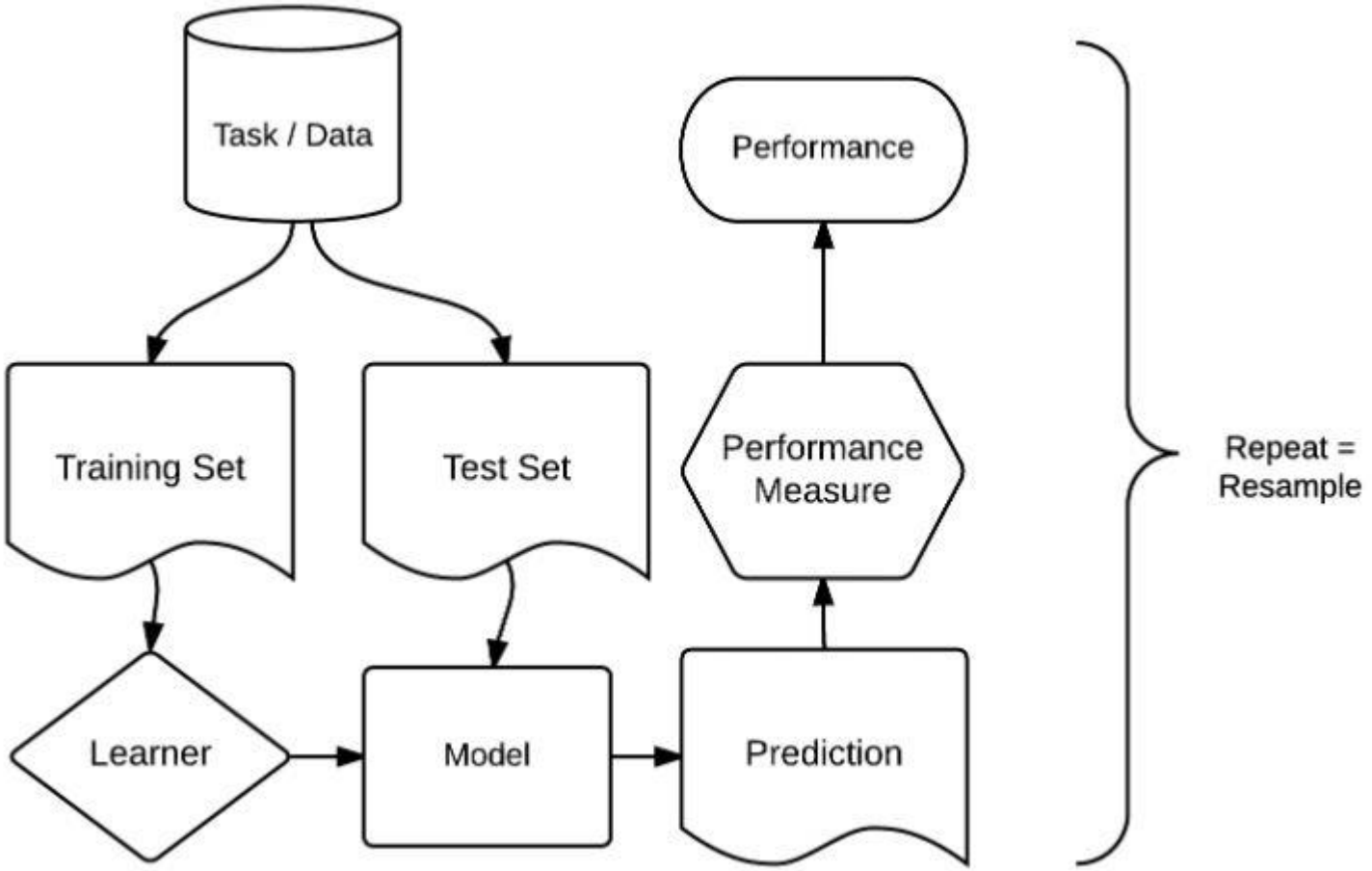
Cross-validation (Validação Cruzada)



1. Use all of the data except a single case as the training set.
2. Predict the value of the single test case.
3. Repeat until every case has been the test case.
4. The predictions for each case are compared to the true values.

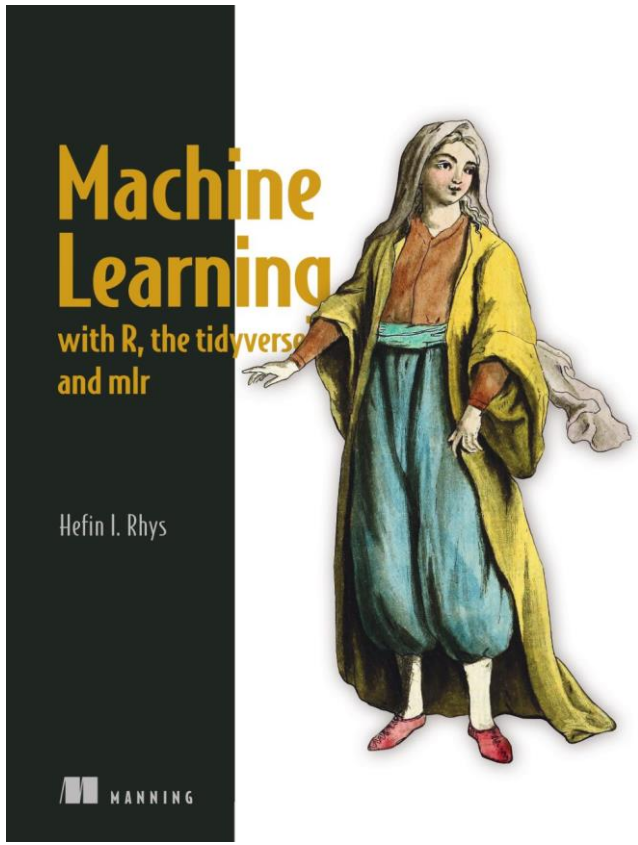
Ryns, 2020

Típico fluxo de trabalho



Yay! :)

Material Extra



Decision Trees:

https://www.youtube.com/watch?v=7VeUPuFGJHk&ab_channel=StatQuestwithJoshStarmer

Random Forest:

https://www.youtube.com/watch?v=J4Wdy0Wc_xQ&t=7s&ab_channel=StatQuestwithJoshStarmer

https://www.youtube.com/watch?v=nyxTdL_4Q-Q&ab_channel=StatQuestwithJoshStarmer

Material Extra

Curso Google Earth Engine (Sadeck)

https://www.youtube.com/watch?v=Dqjtoj9AJak&list=PLNFvG6bTA4NReWtgC93Mh9Tw1RNG4EBMP&ab_channel=LuisSadeck

Google Earth Engine: Explaining all Classifiers

https://www.youtube.com/watch?v=D_KaouS3q20&ab_channel=ProgramSam