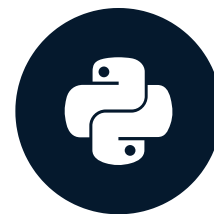


# Model generalization: bootstrapping and cross-validation

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

**Lisa Stuart**  
Data Scientist



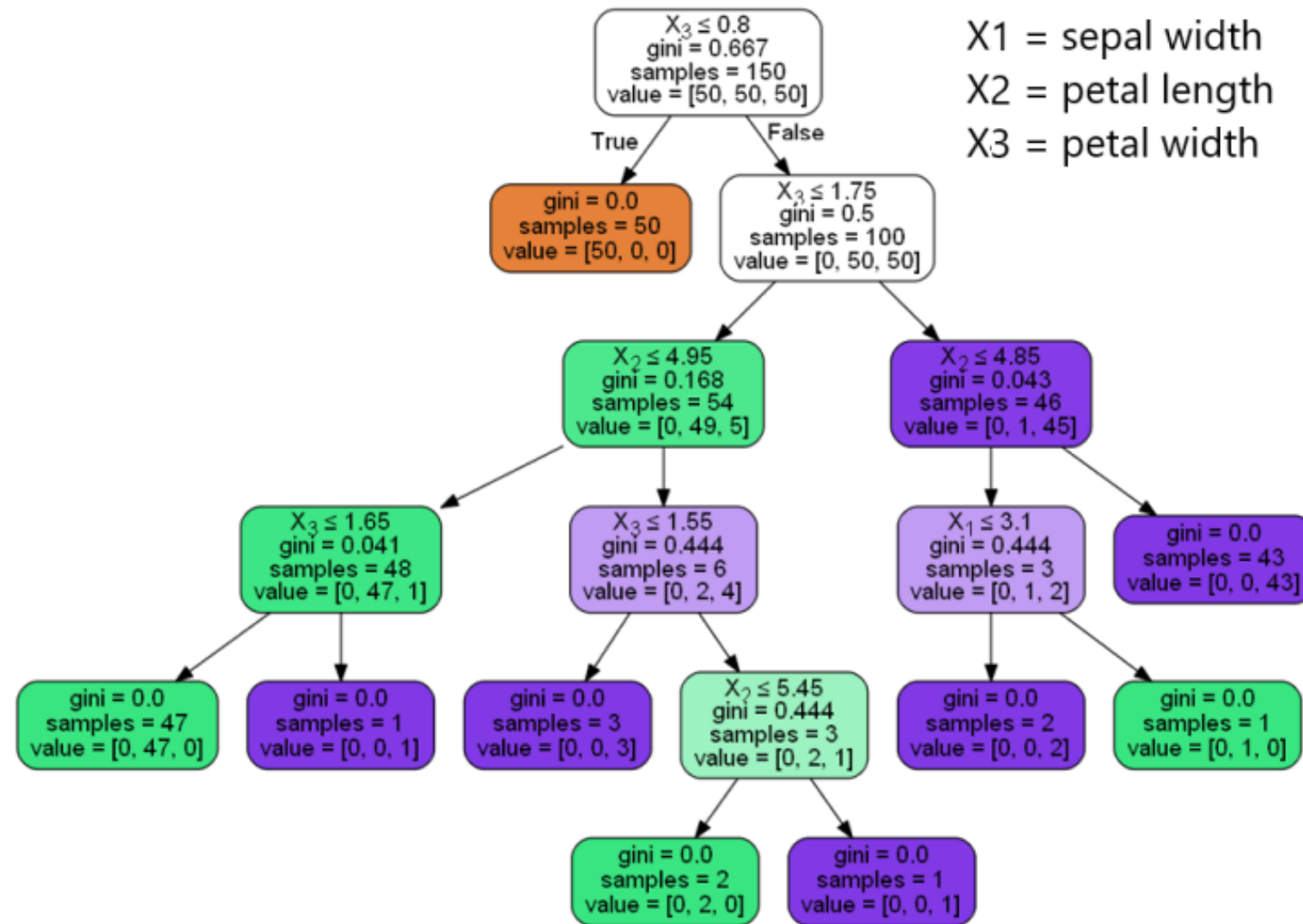
# Chapter 4 overview

- Bootstrapping/cross-validation --> model generalization
- Imbalanced classes
- Correlated features
- Ensemble model selection

# Model generalization

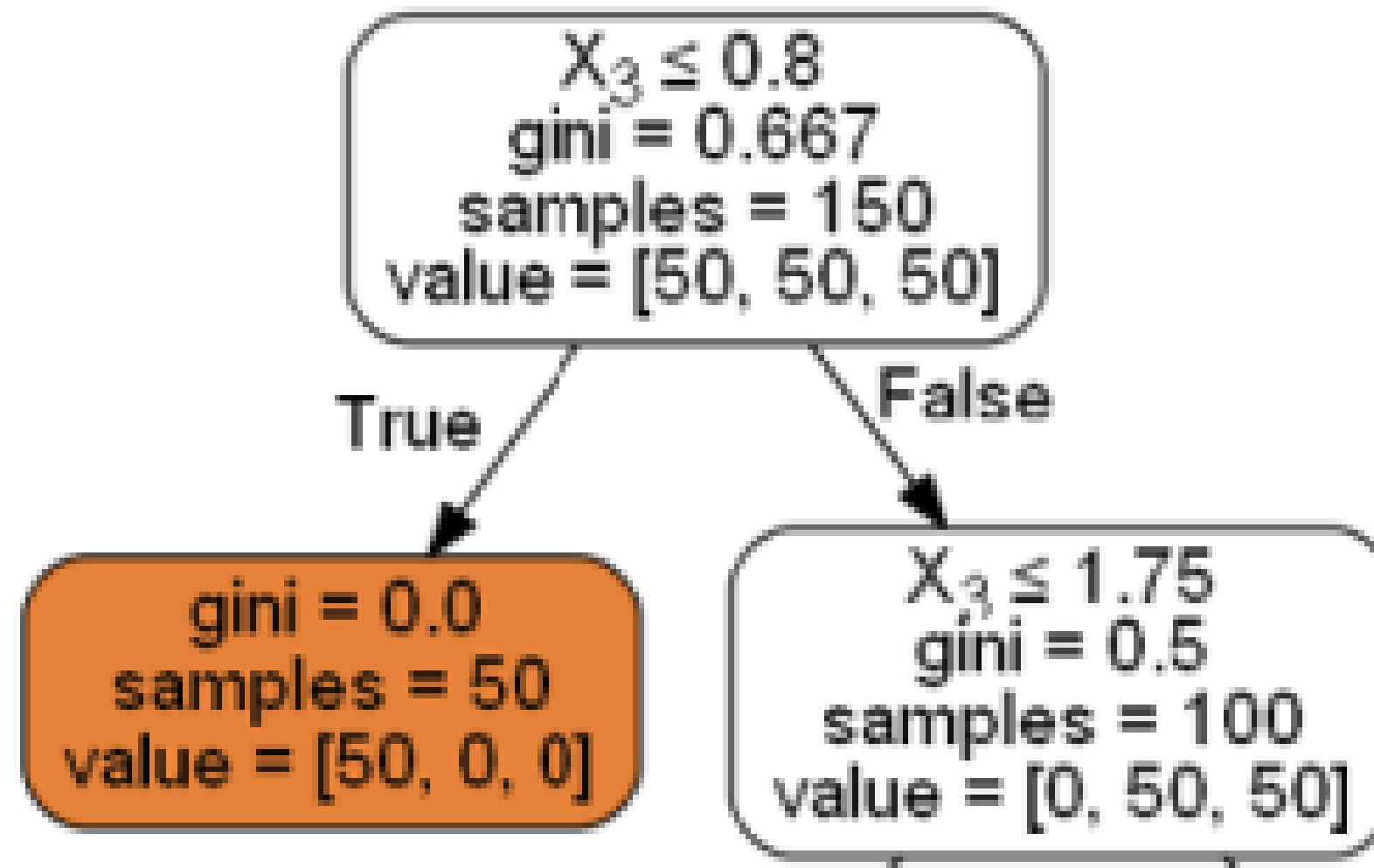
- A ML model's ability to perform well on unseen data
  - test dataset
  - future data
- Train metrics  $\approx$  test metrics
- Overfit models do not generalize

# Decision tree

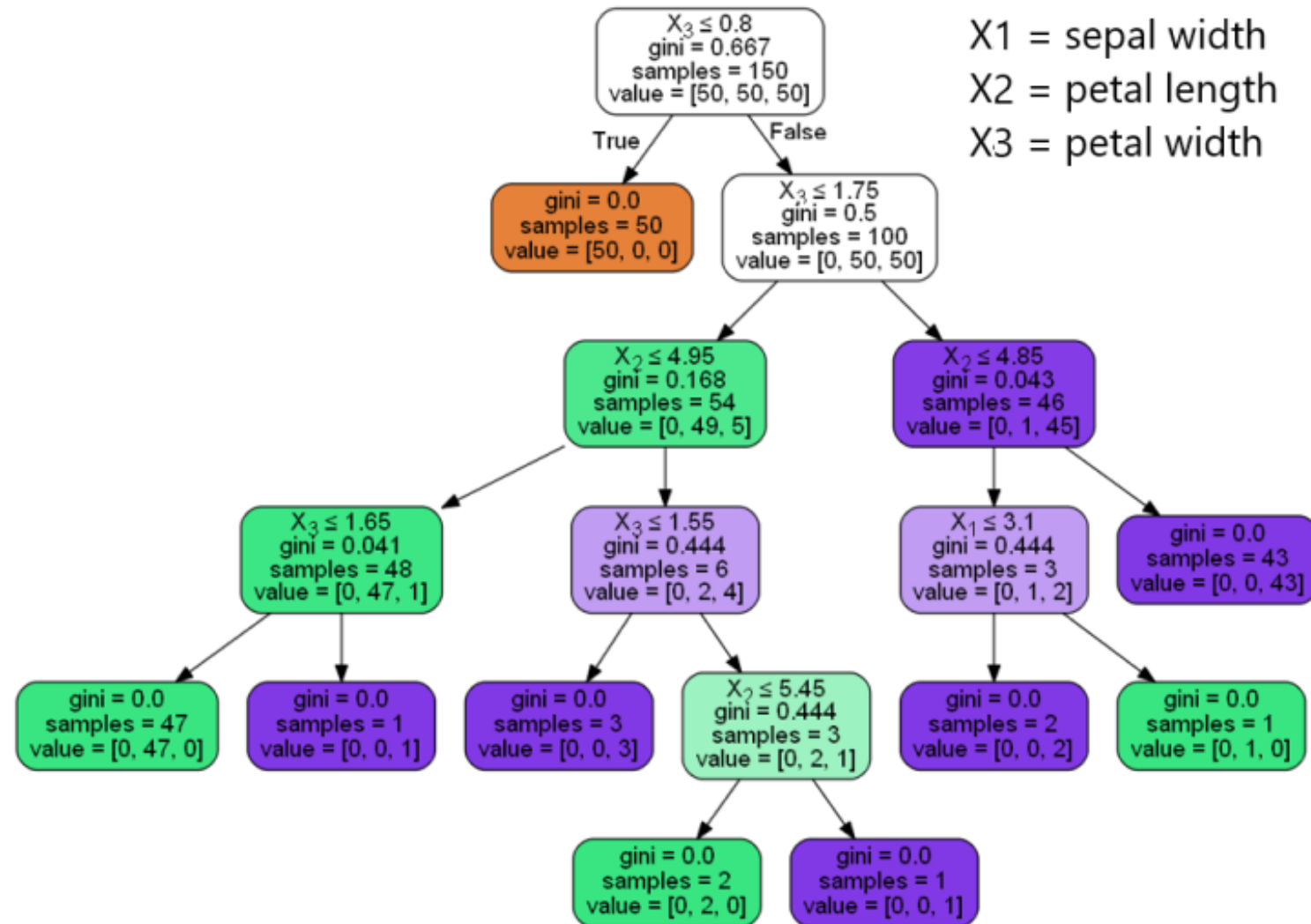


<sup>1</sup> <https://medium.com/@rnbrown/creating-and-visualizing-decision-trees-with-python-f8e8fa394176>

# Decision tree nodes

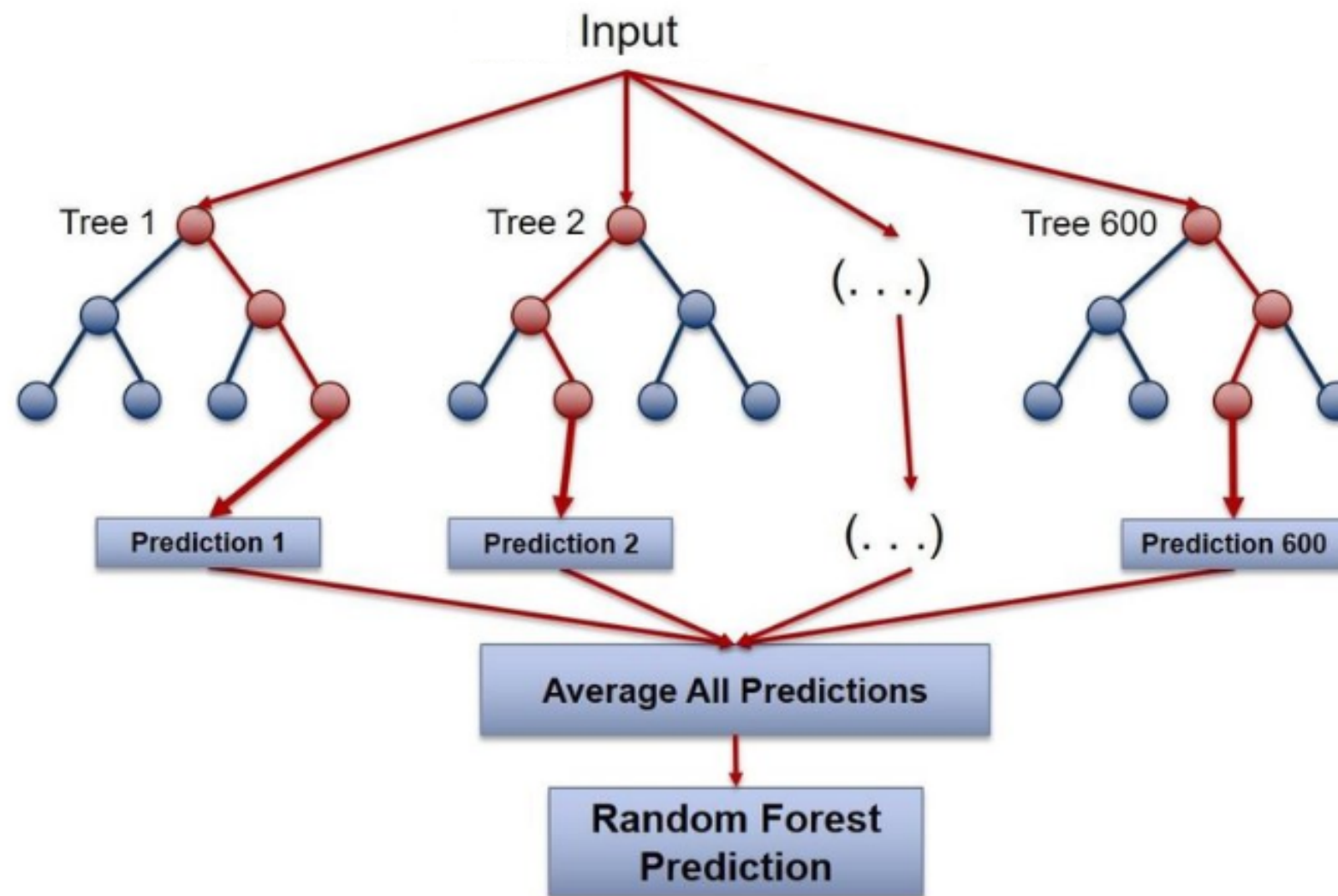


# Advantages vs disadvantages



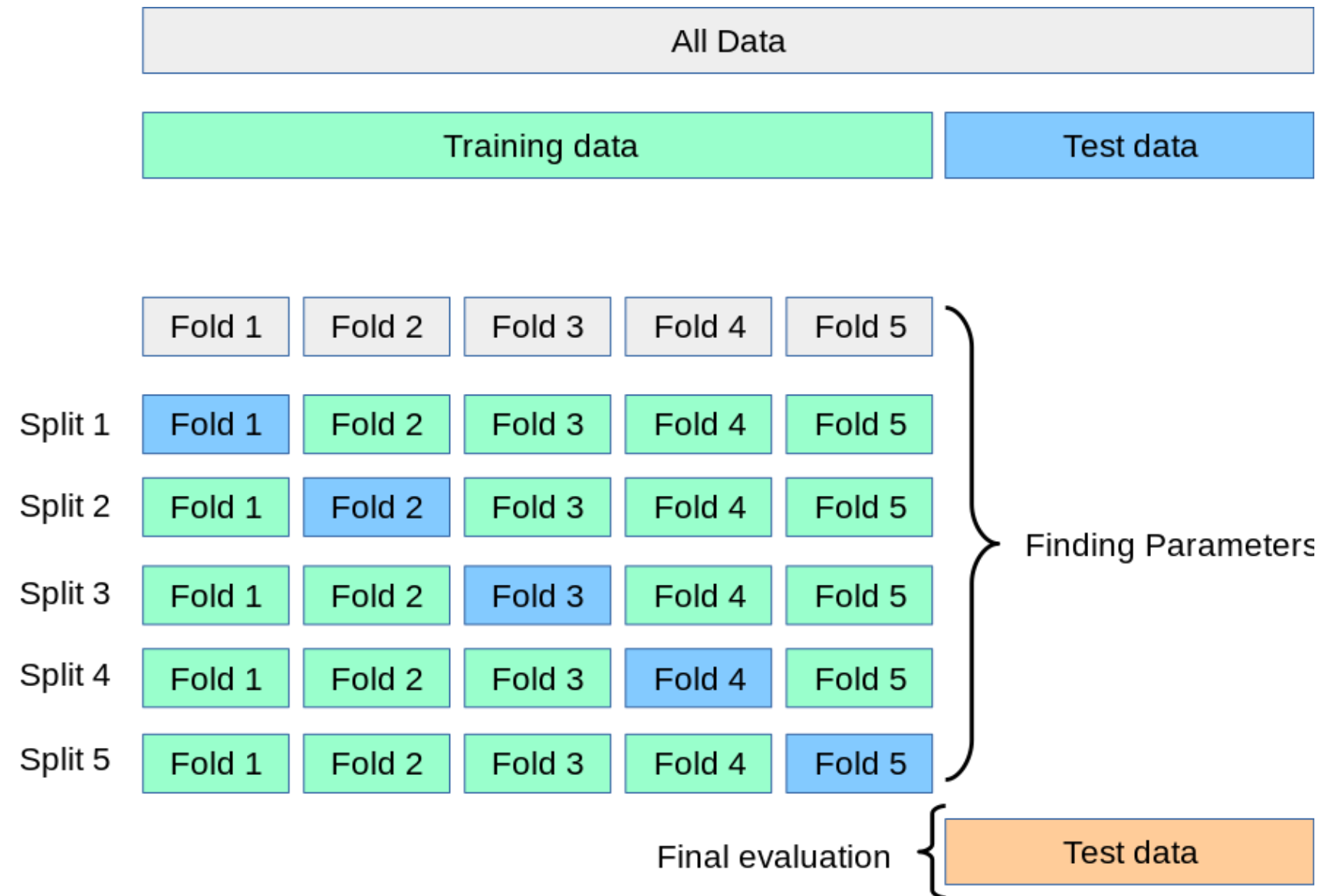
- Advantages:
  - Easy to understand
  - Easy to visualize
- Disadvantages:
  - Easily overfit
  - Considered greedy
  - Biased in cases of class imbalance

# Random Forest



<sup>1</sup> [https://www.researchgate.net/figure/Random-Forest-visualization\\_fig11\\_326560291](https://www.researchgate.net/figure/Random-Forest-visualization_fig11_326560291)

# K-fold cross-validation



<sup>1</sup> [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html)



# Functions

```
# decision tree
`sklearn.tree.DecisionTreeClassifier`

# random forest
`sklearn.ensemble.RandomForestClassifier`

# cross-validated grid search
`sklearn.model_selection.GridSearchCV`

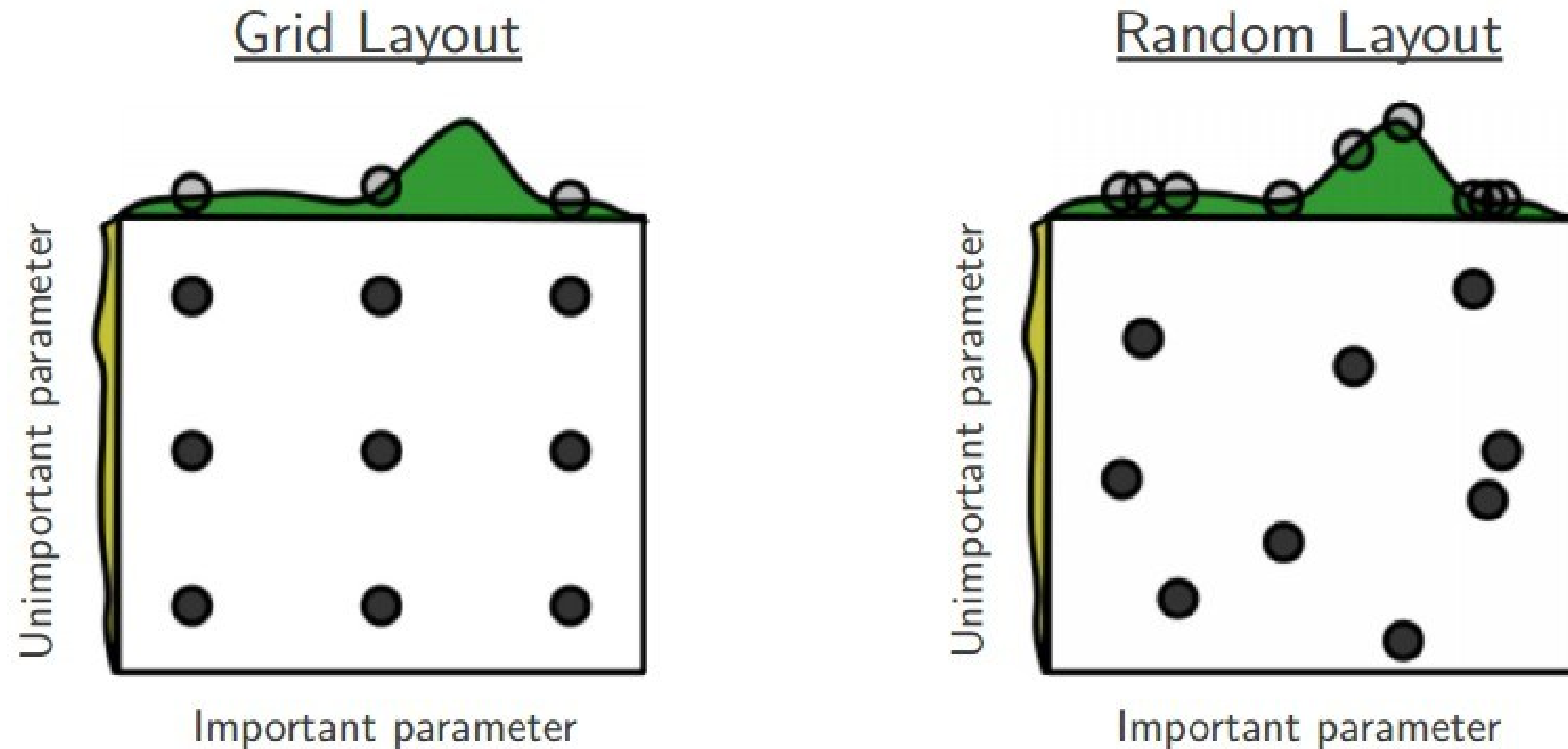
# model accuracy
`sklearn.metrics.accuracy_score`
```

```
# train/test split function
`sklearn.model_selection.train_test_split`

# Parameters that gave best results
`cross_val_model.best_params_`

# Mean cross-validated score of
# estimator with best params
`cross_val_model.best_score_`
```

# GridSearchCV vs RandomSearchCV



# Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

# Model evaluation: imbalanced classification models

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

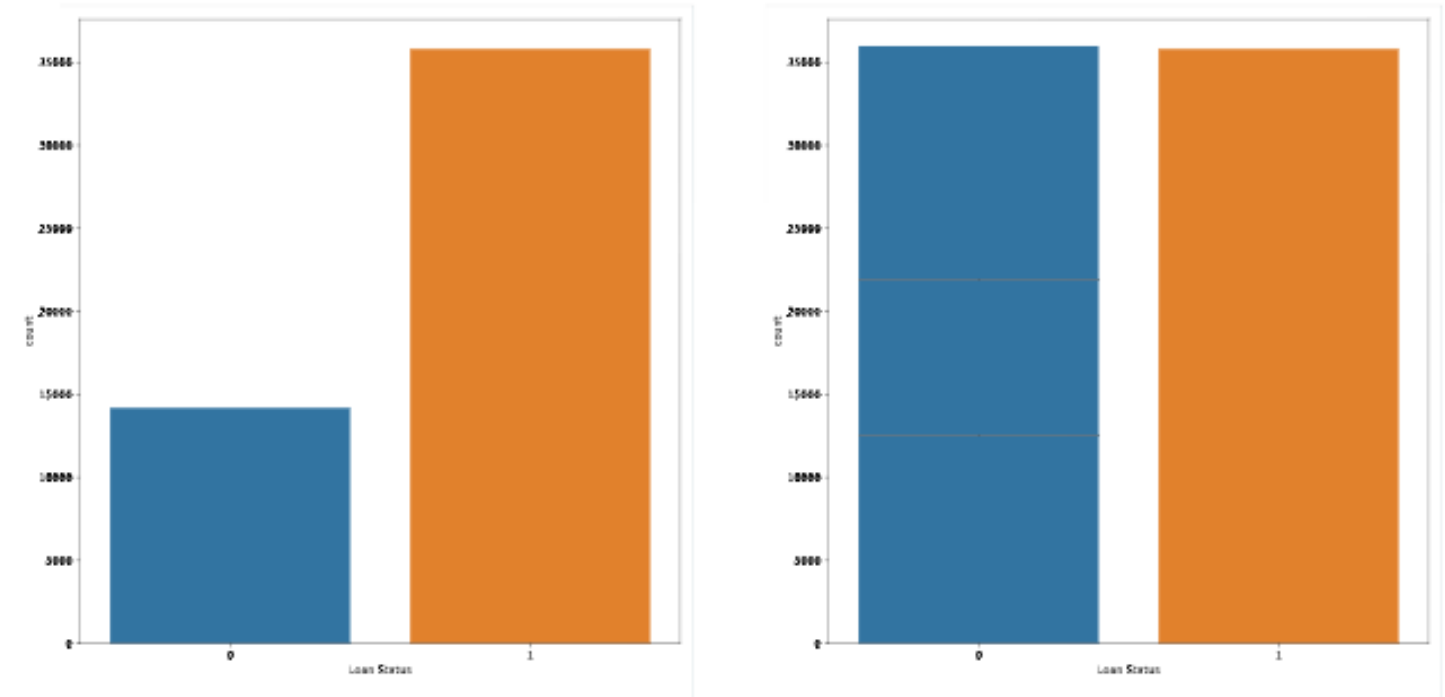


**Lisa Stuart**  
Data Scientist

# Class imbalance

- Categorical target variable
  - Approx equal number observations/class
  - Large difference --> misleading results

Imbalanced Classes vs Balanced Classes



# Confusion matrix

Confusion Matrix

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

<sup>1</sup> <https://scaryscientist.blogspot.com/2016/03/confusion-matrix.html>

# Performance metrics

## Model Performance

Accuracy  $= (TN+TP)/(TN+FP+FN+TP)$

Precision  $= TP/(FP+TP)$

Recall/ Sensitivity  $= TP/(TP+FN)$

Specificity  $= TN/(TN+FP)$

F1  $= 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}$

<sup>1</sup> <https://scaryscientist.blogspot.com/2016/03/confusion-matrix.html>

# Metrics from the matrix

## Confusion Matrix and ROC Curve

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

## Model Performance

Accuracy =  $(TN+TP)/(TN+FP+FN+TP)$

Precision =  $TP/(FP+TP)$

Recall/ Sensitivity =  $TP/(TP+FN)$

Specificity =  $TN/(TN+FP)$

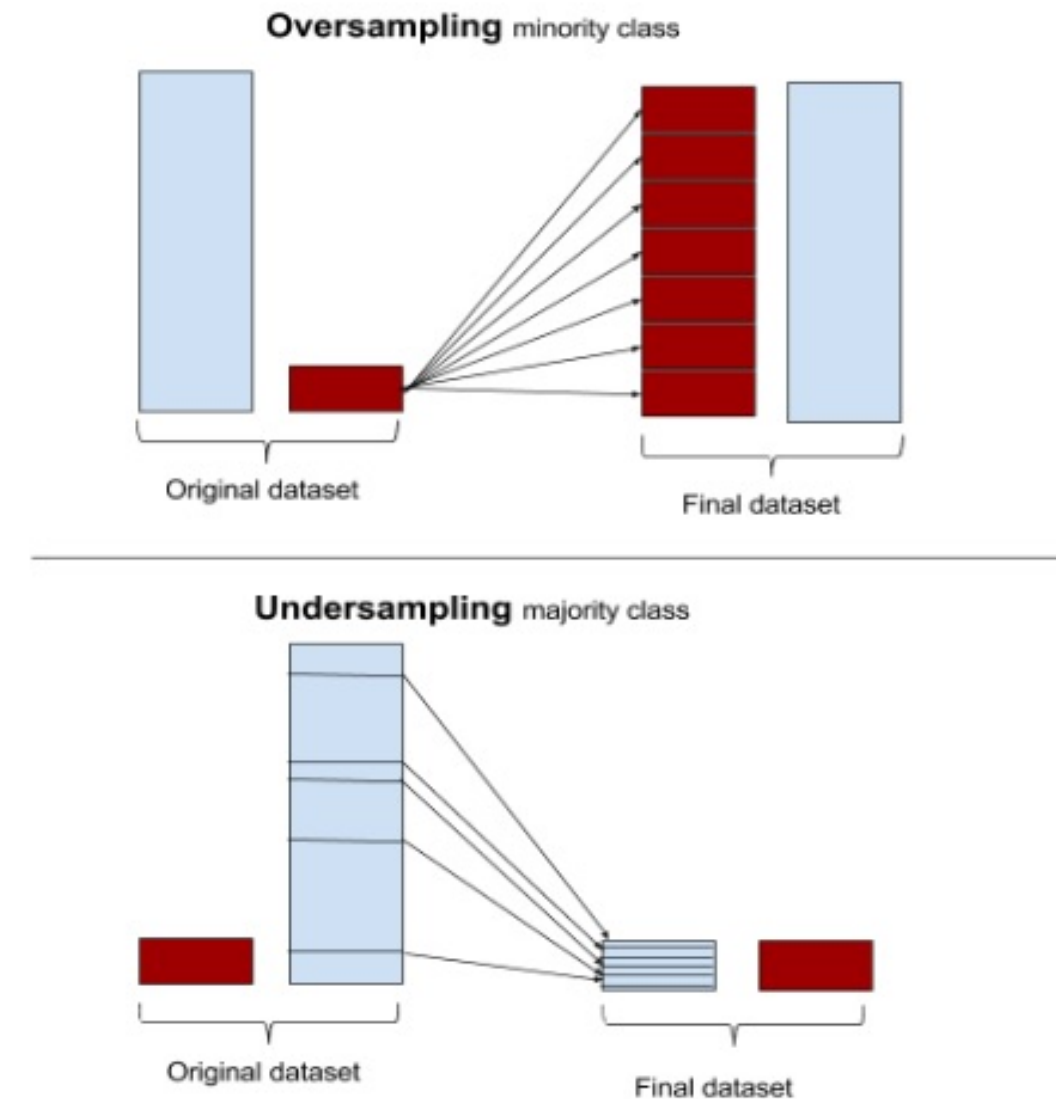
F1 =  $2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}$

<sup>1</sup> <https://scaryscientist.blogspot.com/2016/03/confusion-matrix.html>



# Resampling techniques

- Oversample minority class
- Undersample majority class
- NOTE: Split into test and train sets BEFORE re-sampling!



<sup>1</sup> <https://www.svds.com/learning-imbalanced-classes/>

# Functions

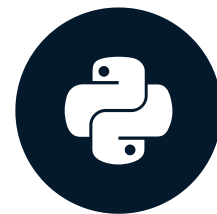
Function	returns
<code>sklearn.linear_model.LogisticRegression</code>	logistic regression
<code>sklearn.metrics.confusion_matrix(y_test, y_pred)</code>	confusion matrix
<code>sklearn.metrics.precision_score(y_test, y_pred)</code>	precision
<code>sklearn.metrics.recall_score(y_test, y_pred)</code>	recall
<code>sklearn.metrics.f1_score(y_test, y_pred)</code>	f1 score
<code>sklearn.utils.resample(deny, n_samples=len(approve))</code>	resamples

# Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

# Model selection: regression models

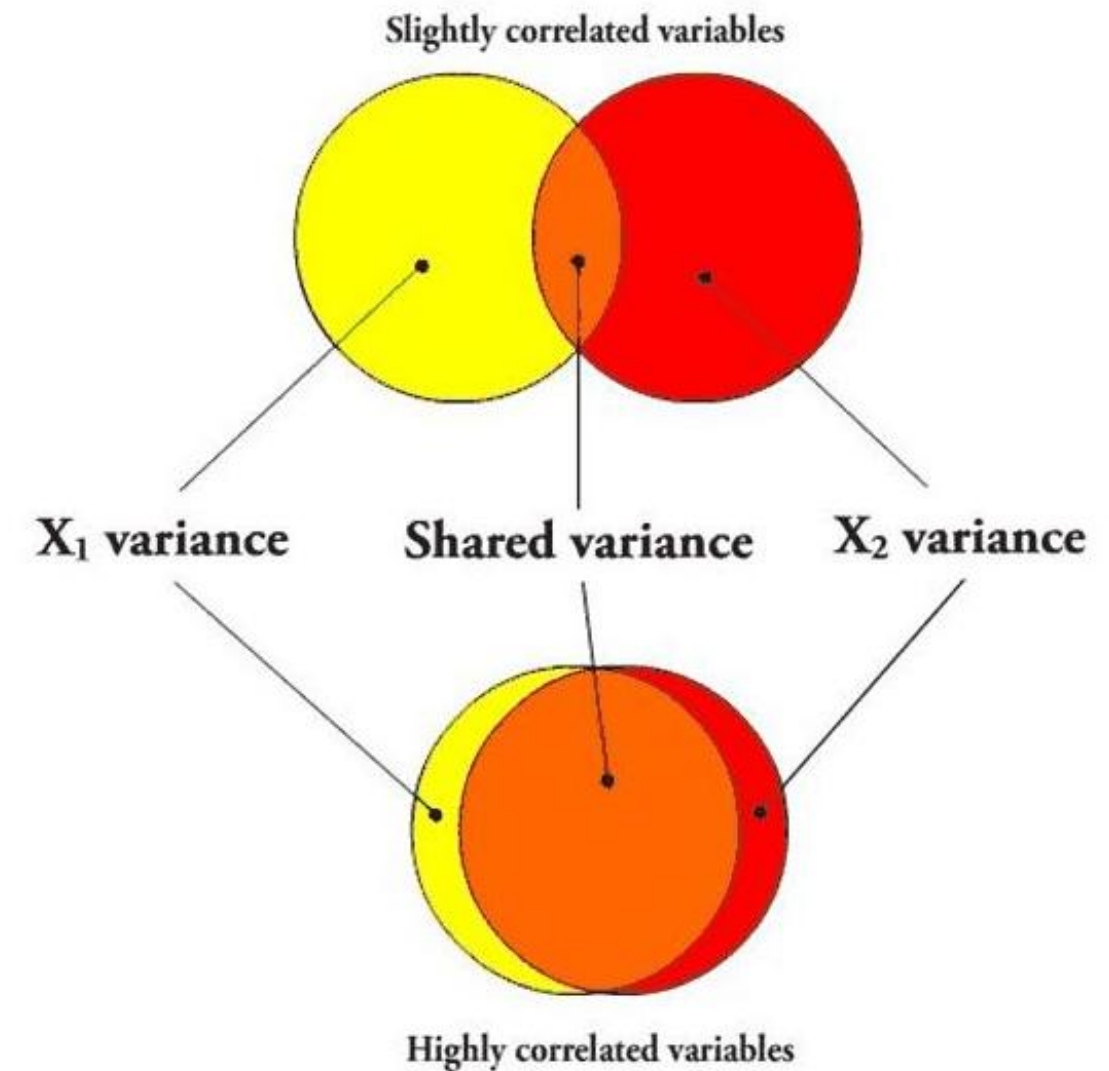
PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON



**Lisa Stuart**  
Data Scientist

# Multicollinearity

- High correlation of independent variables
- Estimated regression coefficients
  - Change in DV explained by IV
  - While holding other vars constant



<sup>1</sup> <https://eigenblogger.com/2010/03/26/post1426/>

# Effects of multicollinearity

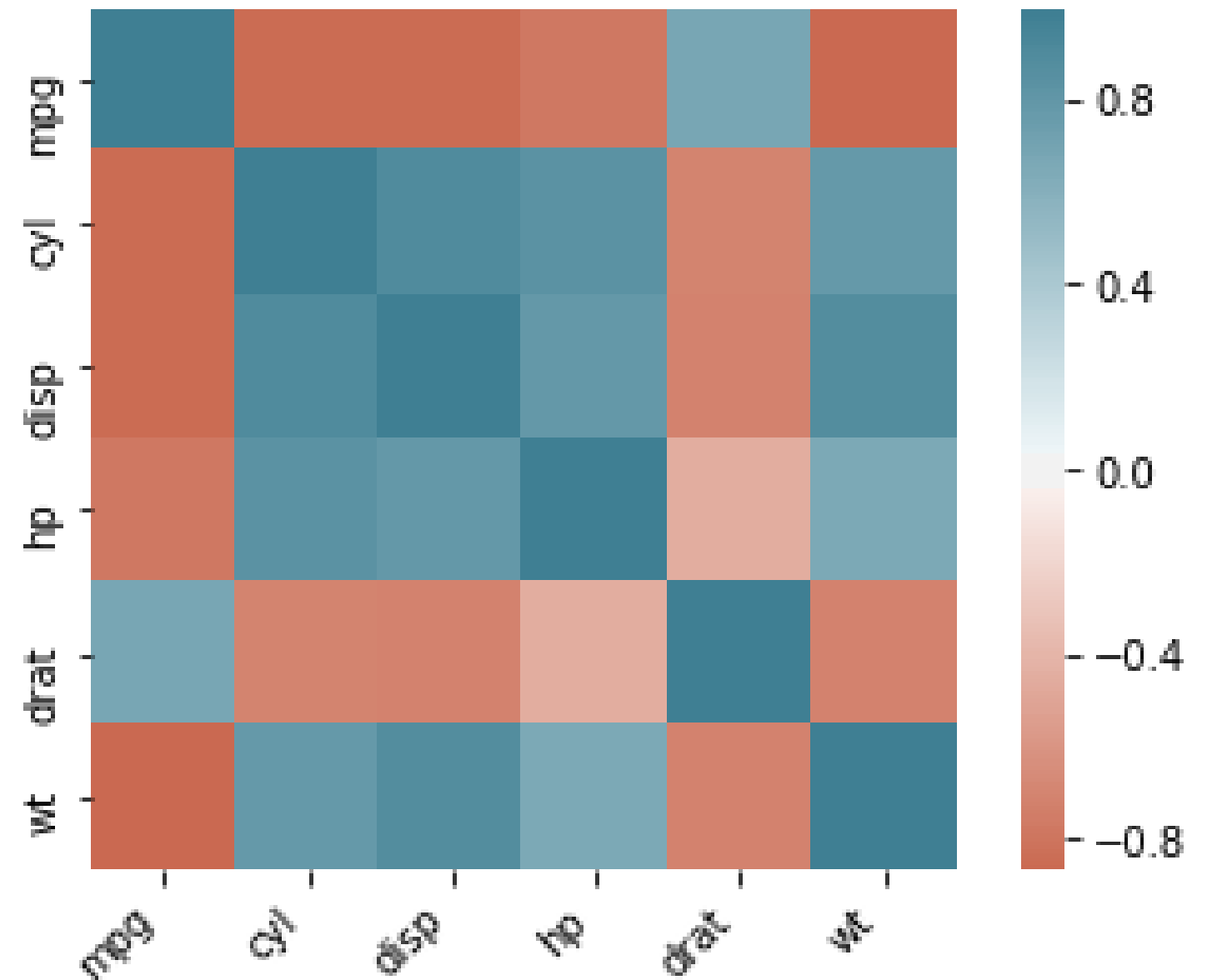
- Reducing coefficients
- Reducing p-values
- Unstable variance
- Overfitting
- Decreased statistical significance due to increased standard error
- True relationship with target variable unclear

# Techniques to address multicollinearity

- Correlation matrix
- Heatmap of correlations
- Calculate the variance inflation factor (VIF)
- Introduce penalizations (Ridge, Lasso)
- PCA

# Correlation matrix vs heatmap

	mpg	cyl	disp	hp	drat	wt
mpg	1.000000	-0.852162	-0.847551	-0.776168	0.681172	-0.867659
cyl	-0.852162	1.000000	0.902033	0.832447	-0.699938	0.782496
disp	-0.847551	0.902033	1.000000	0.790949	-0.710214	0.887980
hp	-0.776168	0.832447	0.790949	1.000000	-0.448759	0.658748
drat	0.681172	-0.699938	-0.710214	-0.448759	1.000000	-0.712441
wt	-0.867659	0.782496	0.887980	0.658748	-0.712441	1.000000





# Variance inflation factor

VIF value	Multicollinearity
$\leq 1$	no
$> 1$	yes, but can ignore
$> 5$	yes, need to address

# Functions

Function/method	returns
<code>sklearn.linear_model.LinearRegression</code>	Linear Regression
<code>data.corr()</code>	correlation matrix
<code>sns.heatmap(corr)</code>	heatmap of correlations
<code>mod.coef_</code>	estimated model coefficients
<code>mean_squared_error(y_test, y_pred)</code>	MSE
<code>r2_score(y_test, y_pred)</code>	R-squared score
<code>df.columns</code>	column names

# Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

# Model selection: ensemble models

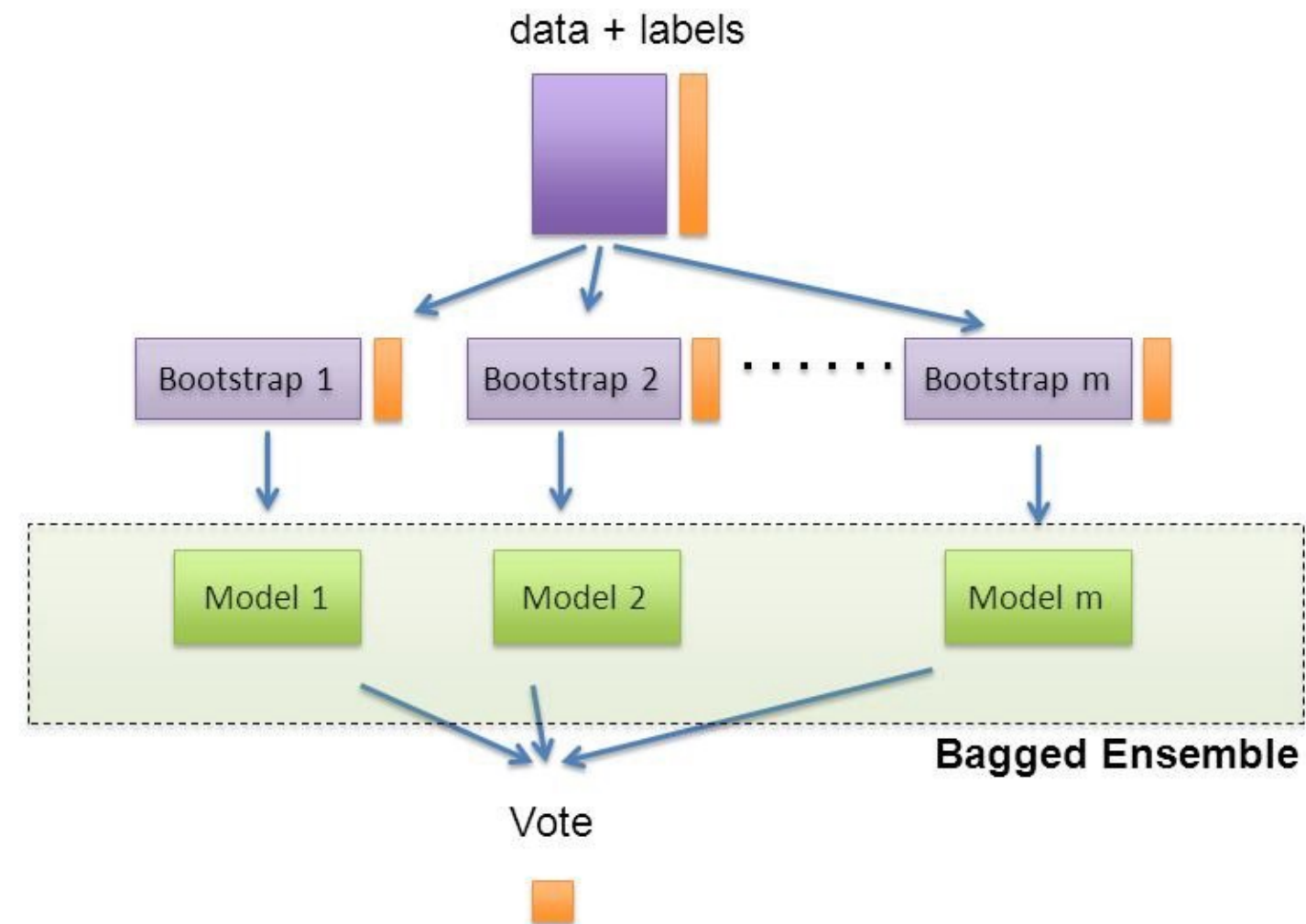
PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON



**Lisa Stuart**  
Data Scientist

# Bootstrapping

“Bagging” : **B**ootstrap **AGG**regating



<sup>1</sup> <https://medium.com/@rrfd/boosting-bagging-and-stacking-ensemble-methods-with-sklearn-and-mlens-a455c0c982de>



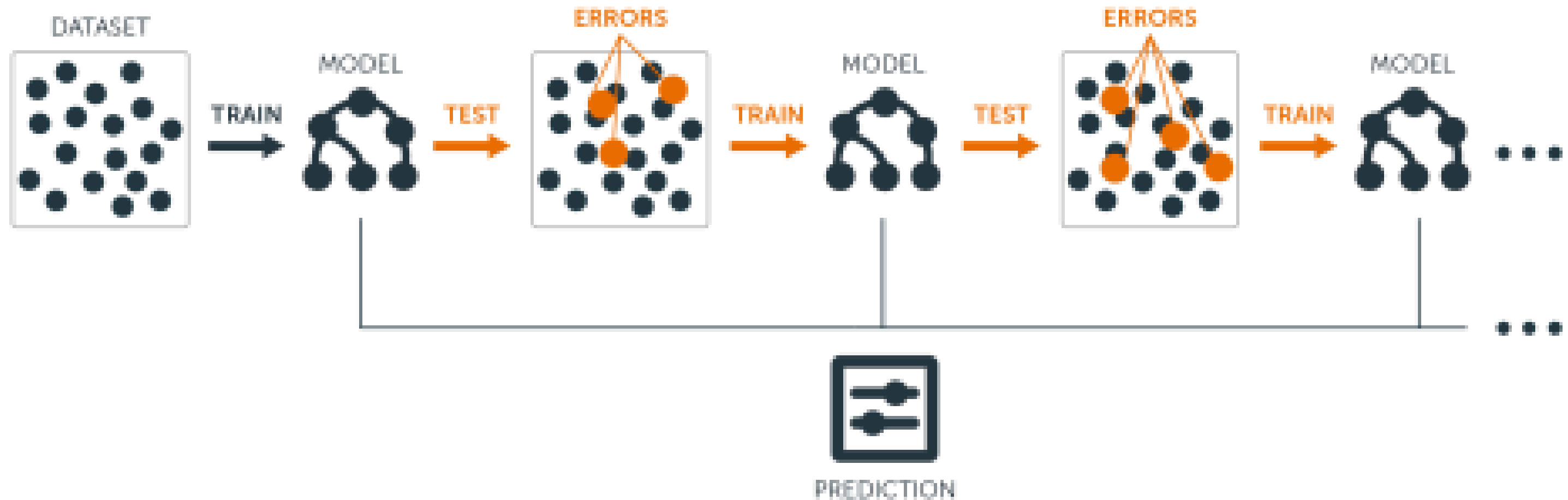
# Random forest



<sup>1</sup> <https://www.sca.com/en/about-us/our-forest/>



# Gradient Boosting



<sup>1</sup> <https://blog.bigml.com/2017/03/14/introduction-to-boosted-trees/>

# RF vs GB

parameter	Random Forest	Gradient Boosting
n_estimators	10	100
criterion	gini (or entropy )	friedman_mse
max_depth	None	3
learning_rate	N/A	0.1

<sup>1</sup> <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble>



# Functions

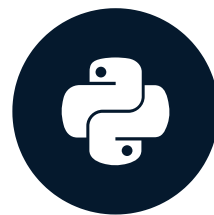
Function	returns
<code>sklearn.ensemble.RandomForestClassifier</code>	Random Forest
<code>sklearn.ensemble.GradientBoostingClassifier</code>	Gradient Boosted Model
<code>sklearn.metrics.accuracy_score</code>	trained model accuracy
<code>sklearn.metrics.confusion_matrix(y_test, y_pred)</code>	confusion matrix
<code>sklearn.metrics.precision_score(y_test, y_pred)</code>	precision
<code>sklearn.metrics.recall_score(y_test, y_pred)</code>	recall
<code>sklearn.metrics.f1_score(y_test, y_pred)</code>	f1 score

# Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

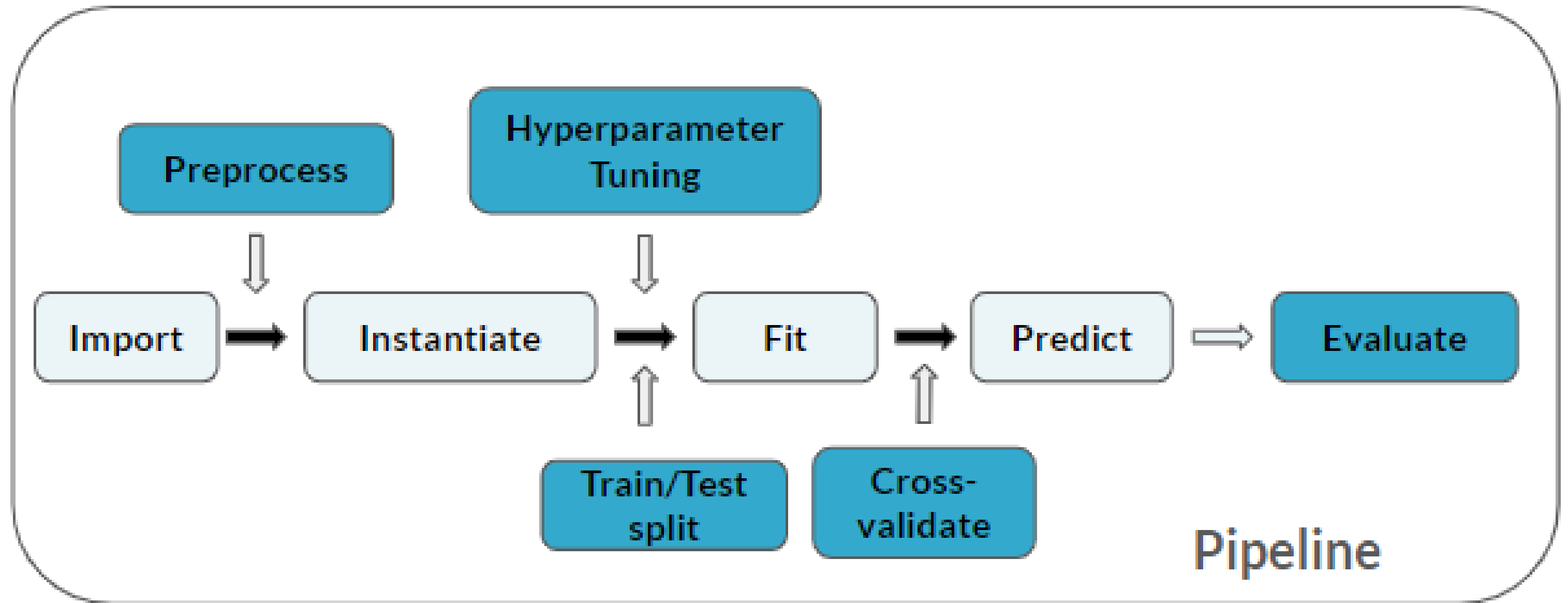
# Wrap-Up

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

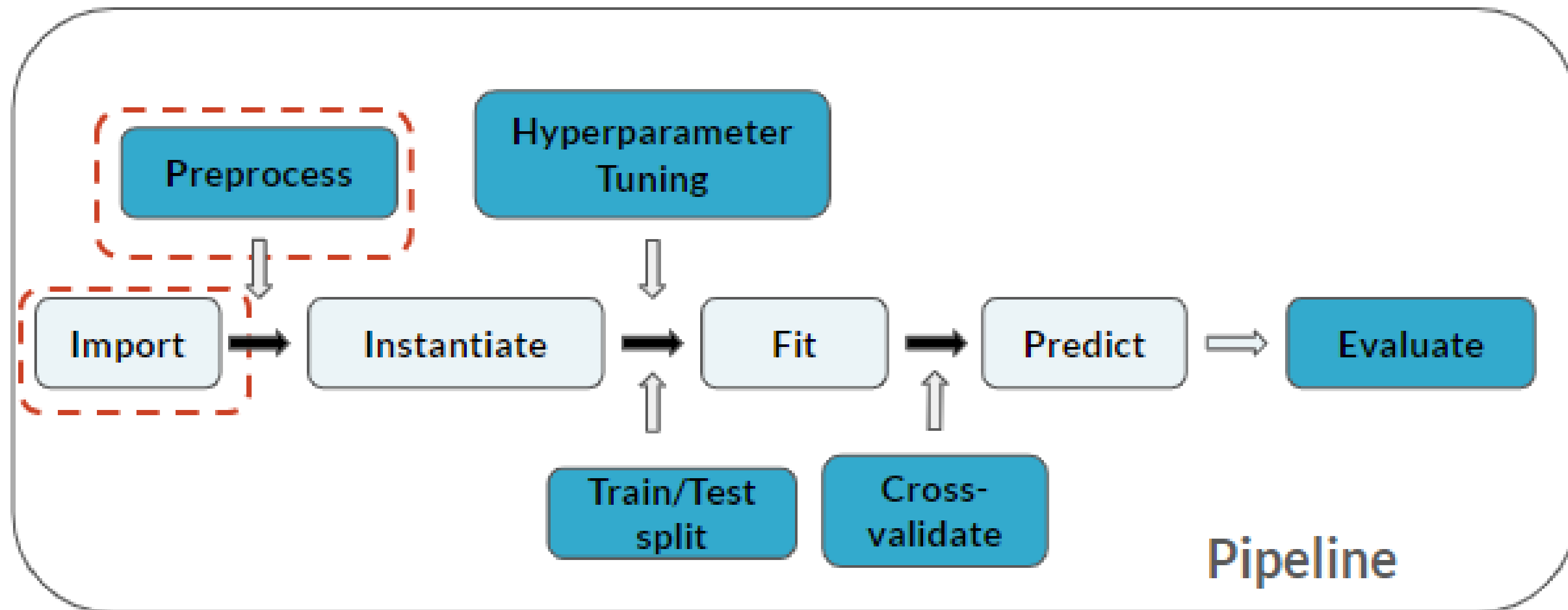


**Lisa Stuart**  
Data Scientist

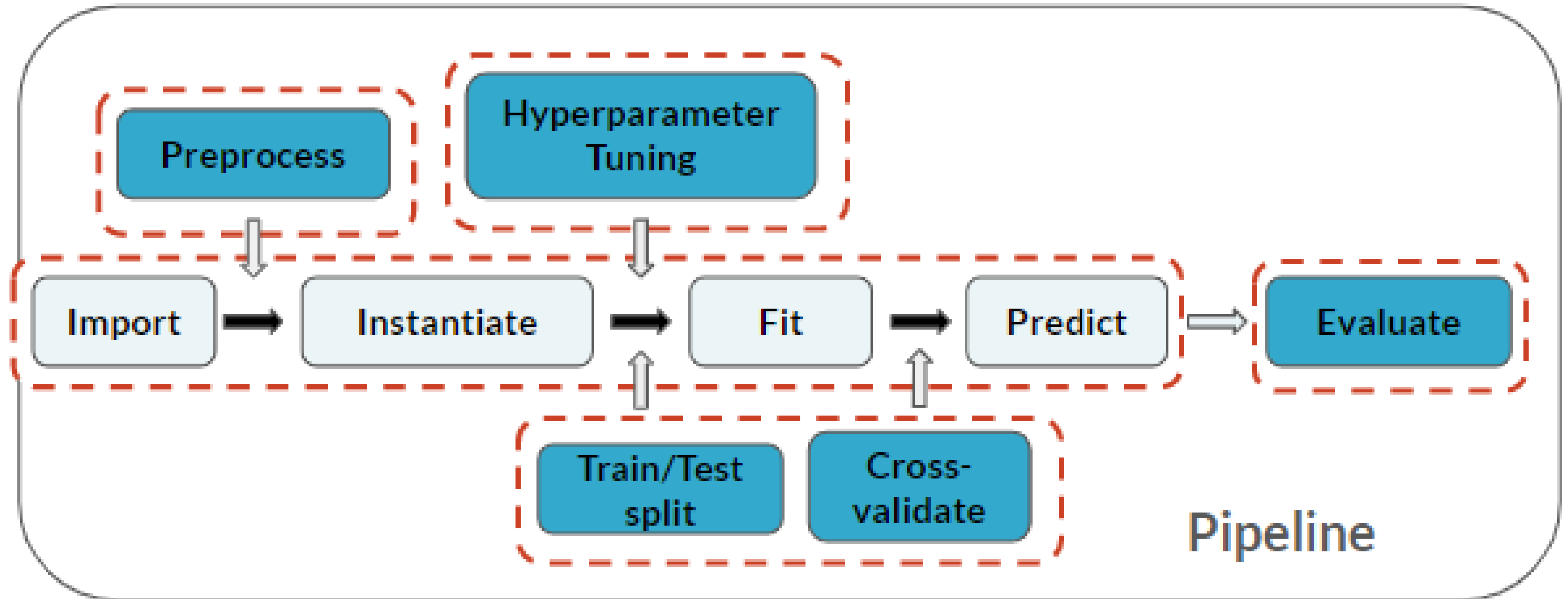
# Machine Learning Pipeline



# Machine Learning Pipeline



# Machine Learning Pipeline



# CONGRATULATIONS!!!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON