Model generalization: bootstrapping and cross-validation

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Lisa Stuart
Data Scientist

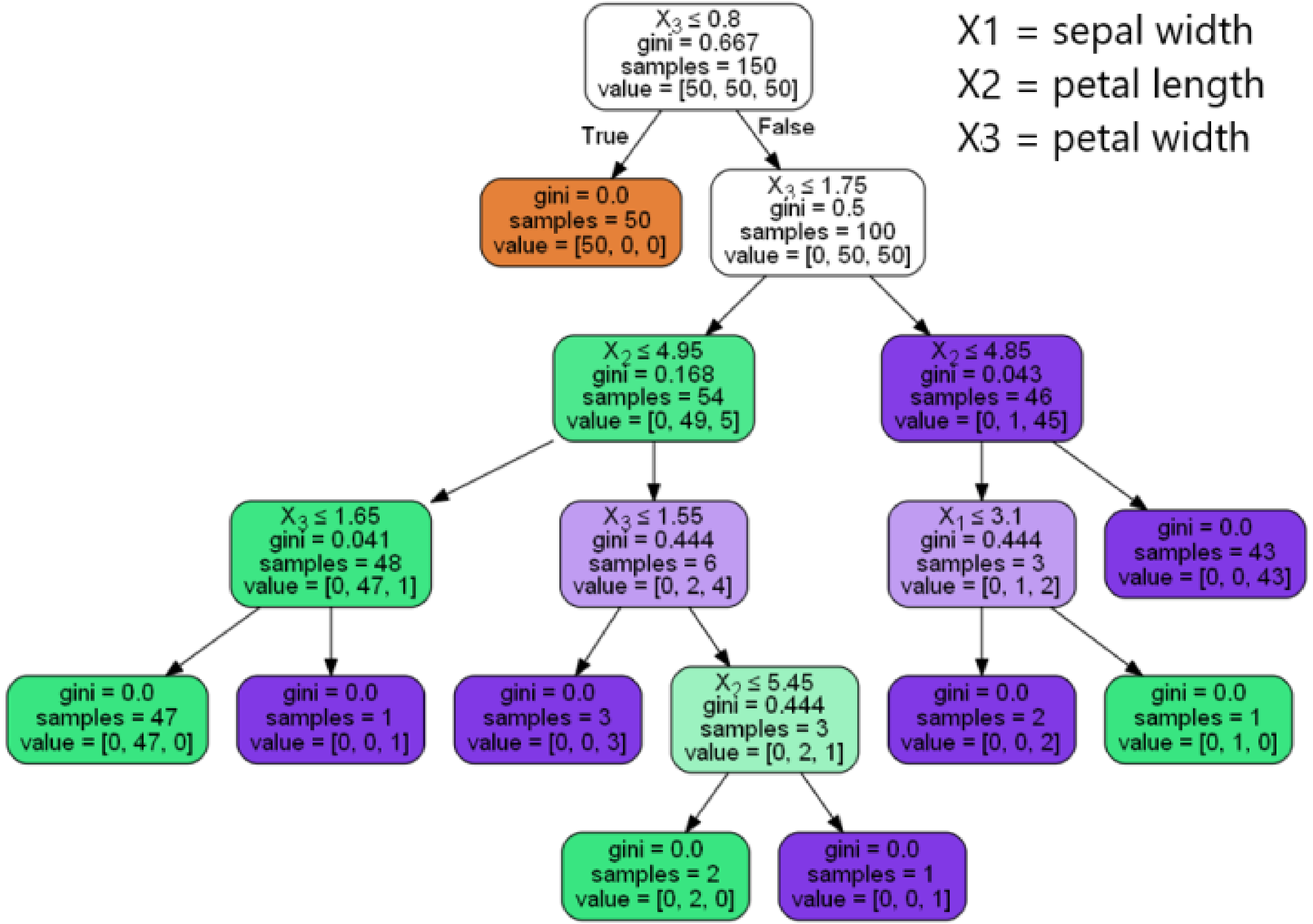
Chapter 4 overview

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

Model generalization

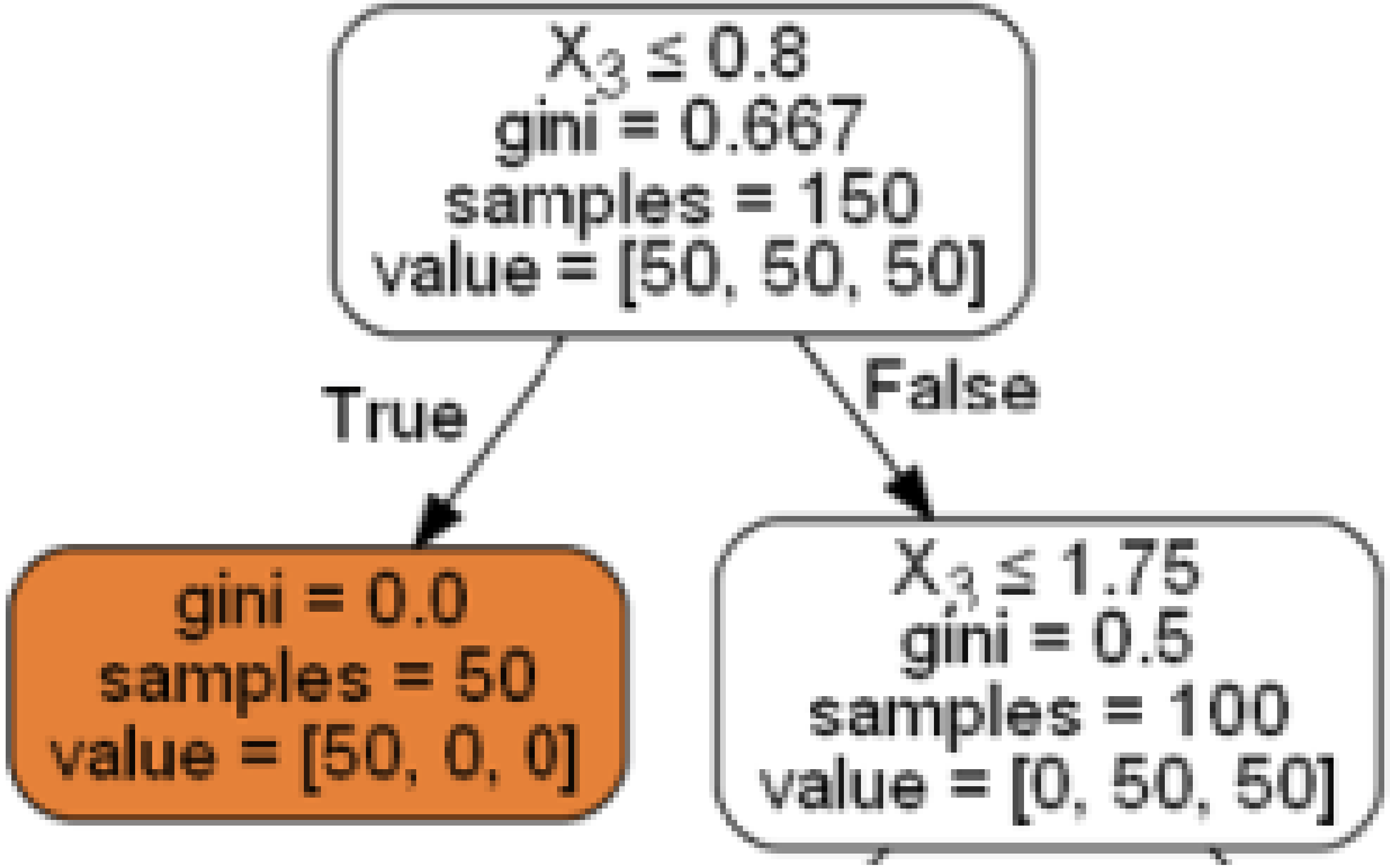
- A ML model's ability to perform well on unseen data
  - test dataset
  - future data
- Train metrics $\approx$ test metrics
- Overfit models do not generalize

Decision tree

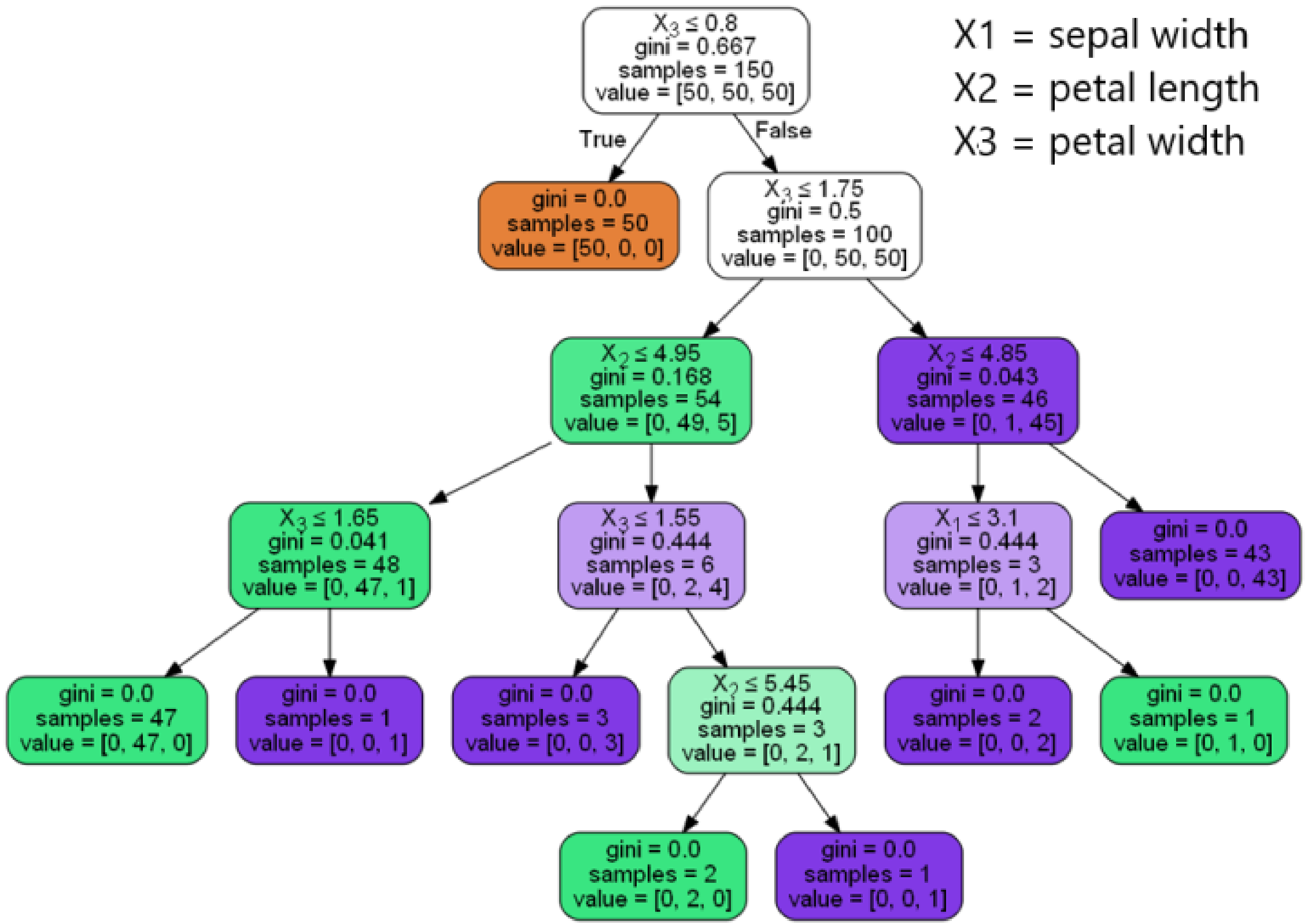


[1] https://medium.com/@rnbrown/creating-and-visualizing-decision-trees-with-python-f8e8fa394176

Decision tree nodes



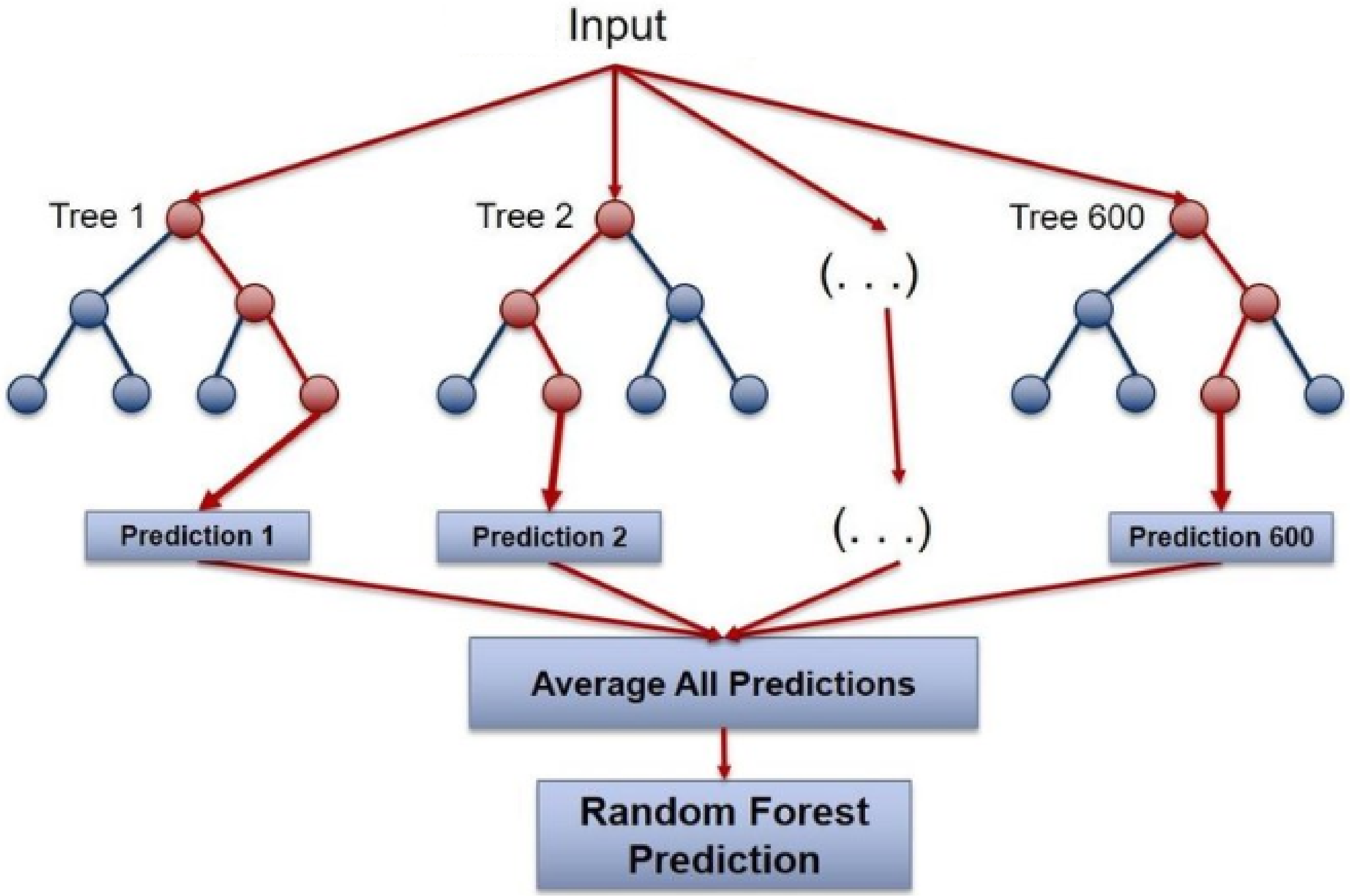
Advantages vs disadvantages



X1 = sepal width
X2 = petal length
X3 = petal width

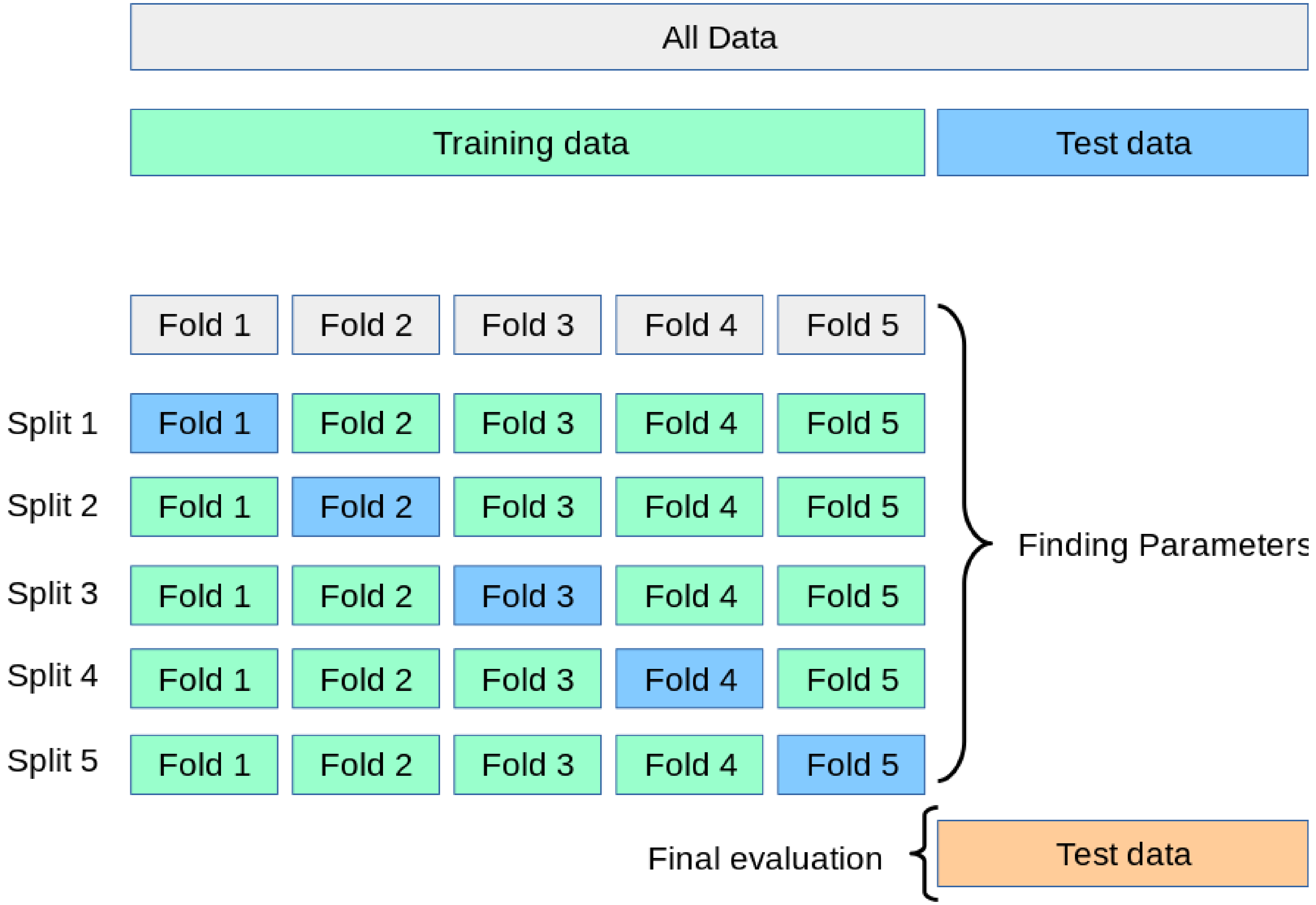
- Advantages:
  - Easy to understand
  - Easy to visualize
- Disadvantages:
  - Easily overfit
  - Considered greedy
  - Biased in cases of class imbalance

Random Forest



[1] https://www.researchgate.net/figure/Random-Forest-visualization_fig11_326560291

K-fold cross-validation



All Data

Training data | Test data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Finding Parameters

Final evaluation | Test data

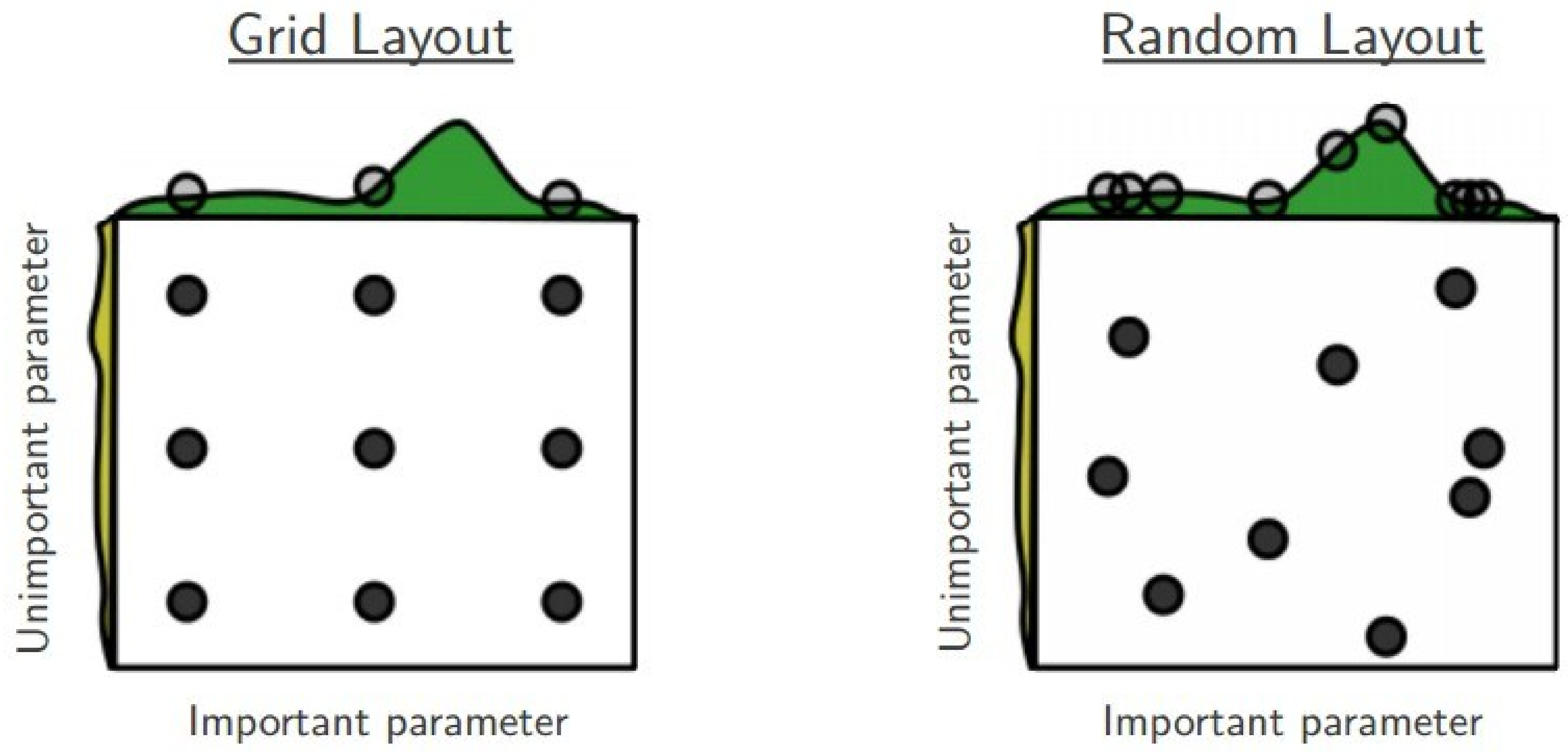
[1] https://scikit-learn.org/stable/modules/cross_validation.html

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

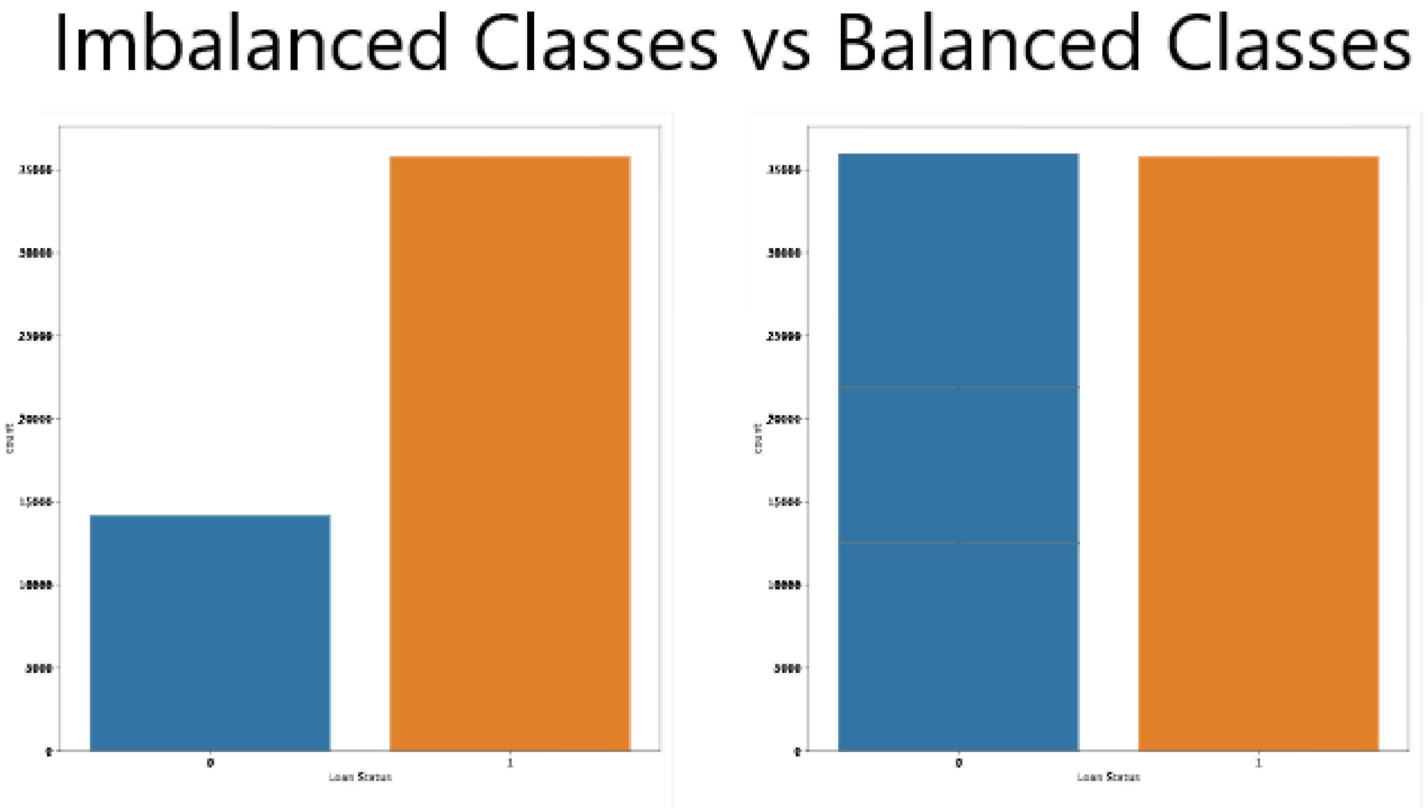
Model evaluation: imbalanced classification models

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Lisa Stuart
Data Scientist

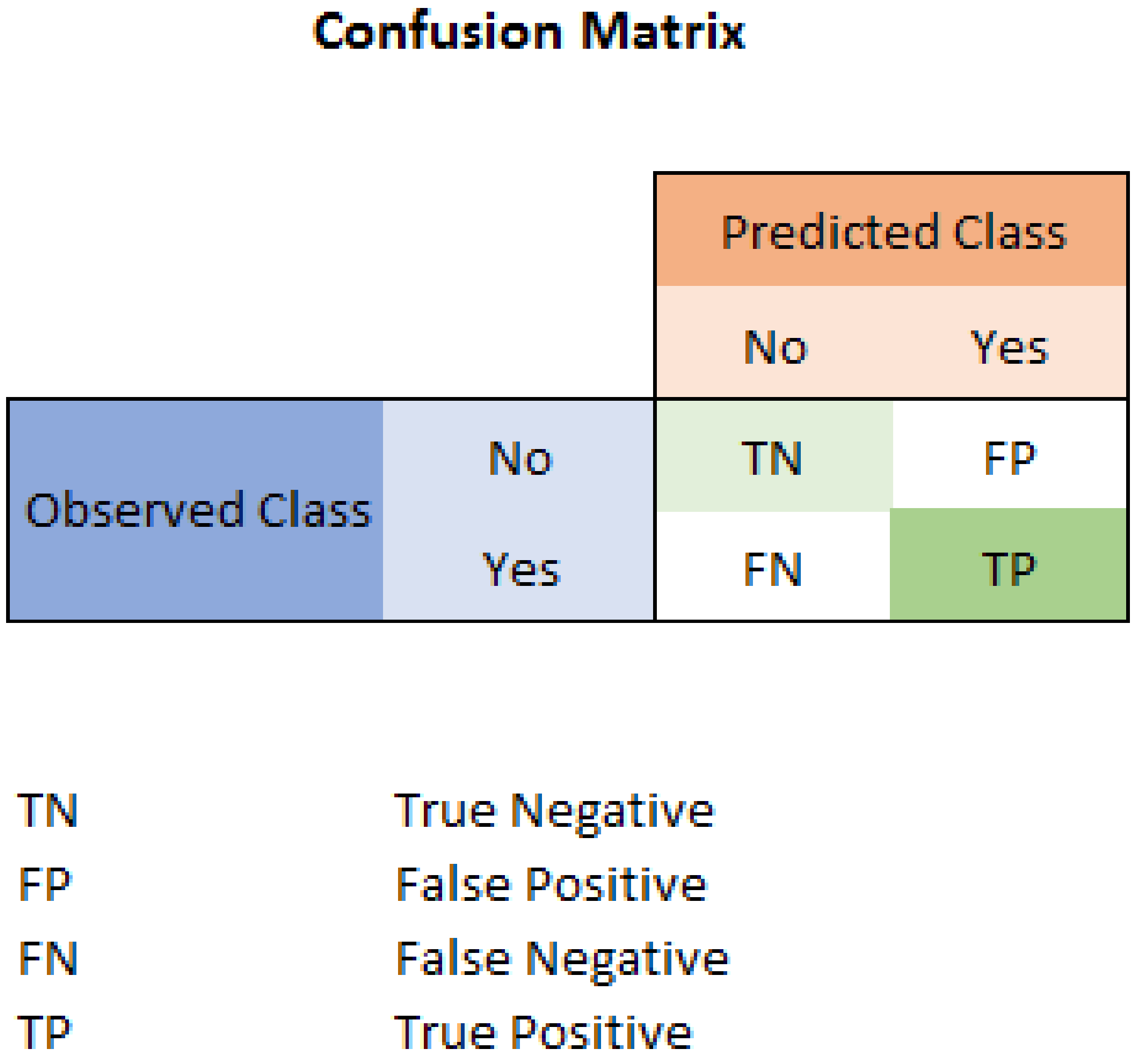
Class imbalance

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



Imbalanced Classes vs Balanced Classes

Confusion matrix



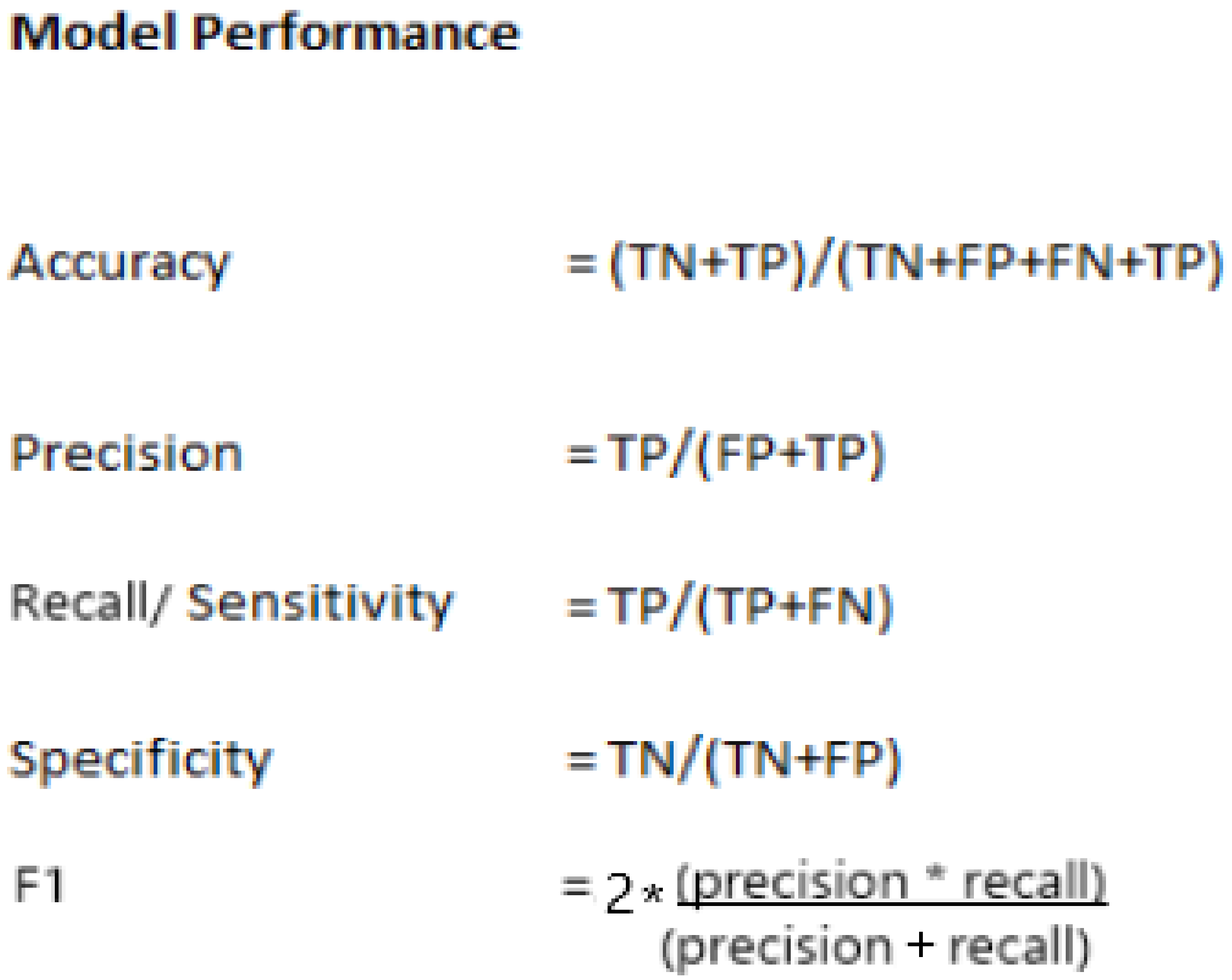
Confusion Matrix

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

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

[1] https://scaryscientist.blogspot.com/2016/03/confusion-matrix.html

Performance metrics

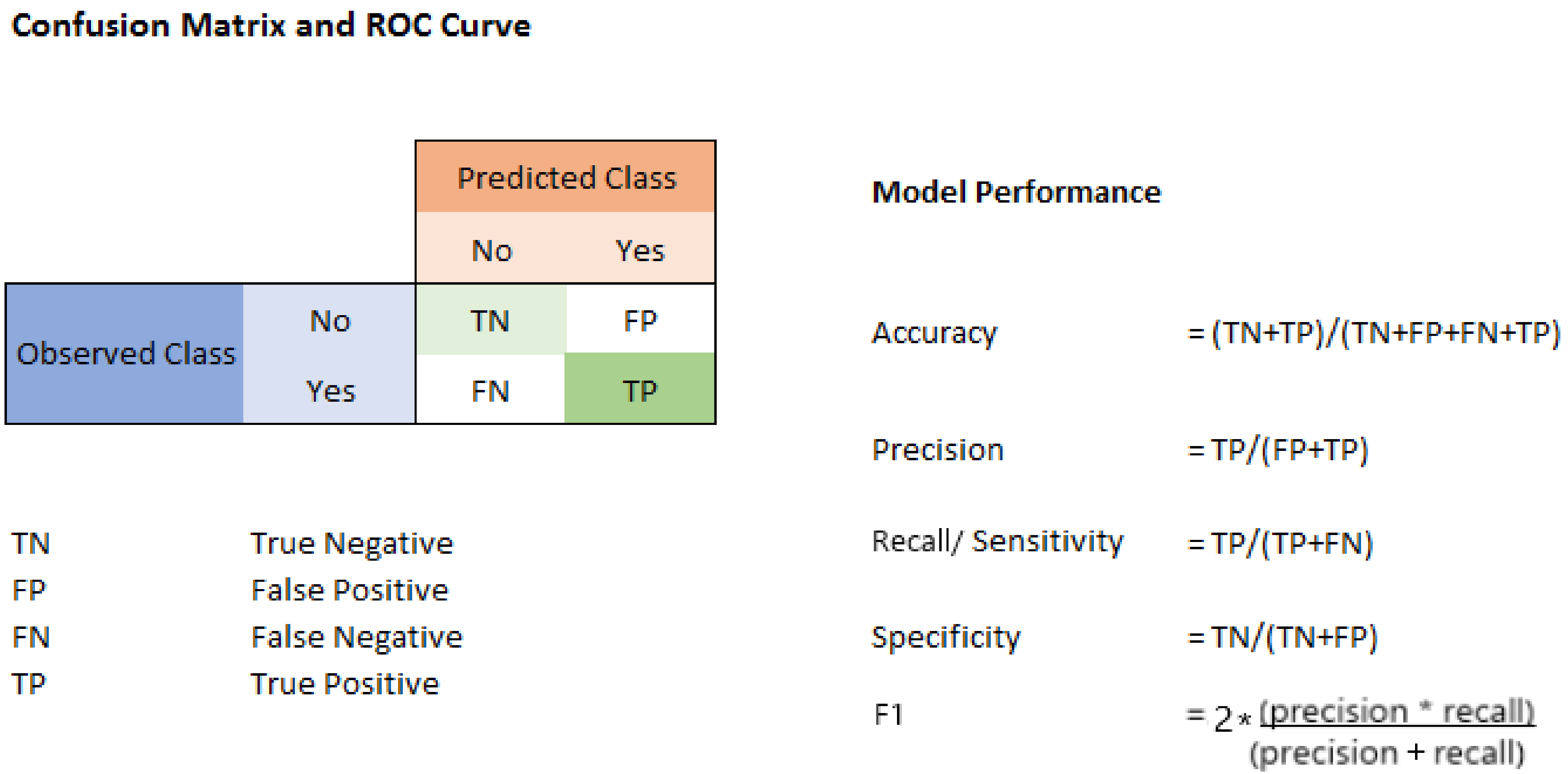


Model Performance

Accuracy	= (TN+TP)/(TN+FP+FN+TP)
Precision	= TP/(FP+TP)
Recall/ Sensitivity	= TP/(TP+FN)
Specificity	= TN/(TN+FP)
F1	$= 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}$

[1] 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

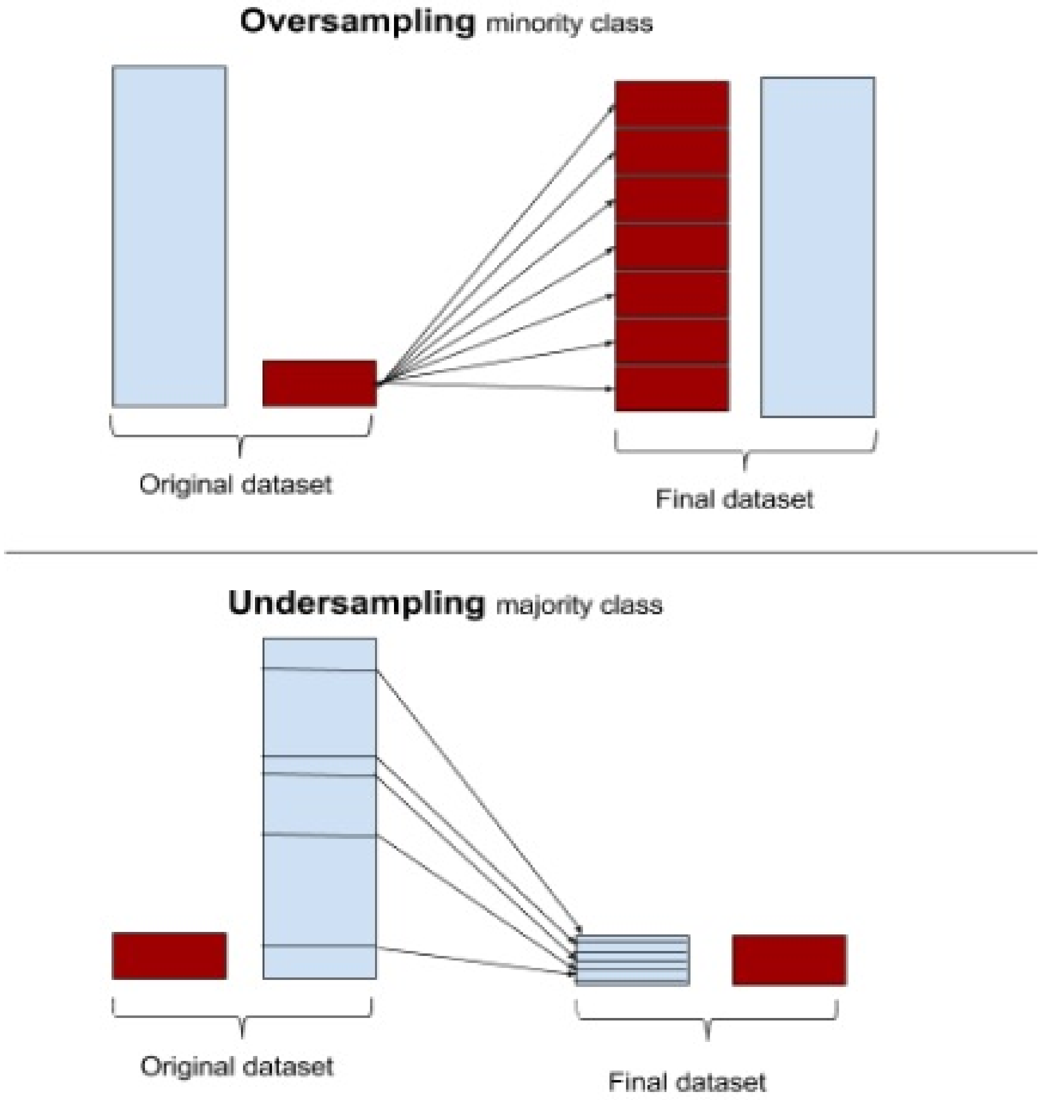
Model Performance

Accuracy	= (TN+TP)/(TN+FP+FN+TP)
Precision	= TP/(FP+TP)
Recall/ Sensitivity	= TP/(TP+FN)
Specificity	= TN/(TN+FP)
F1	$= 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}$

[1] 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!



[1] https://www.svds.com/learning-imbalanced-classes/

Functions

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

Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

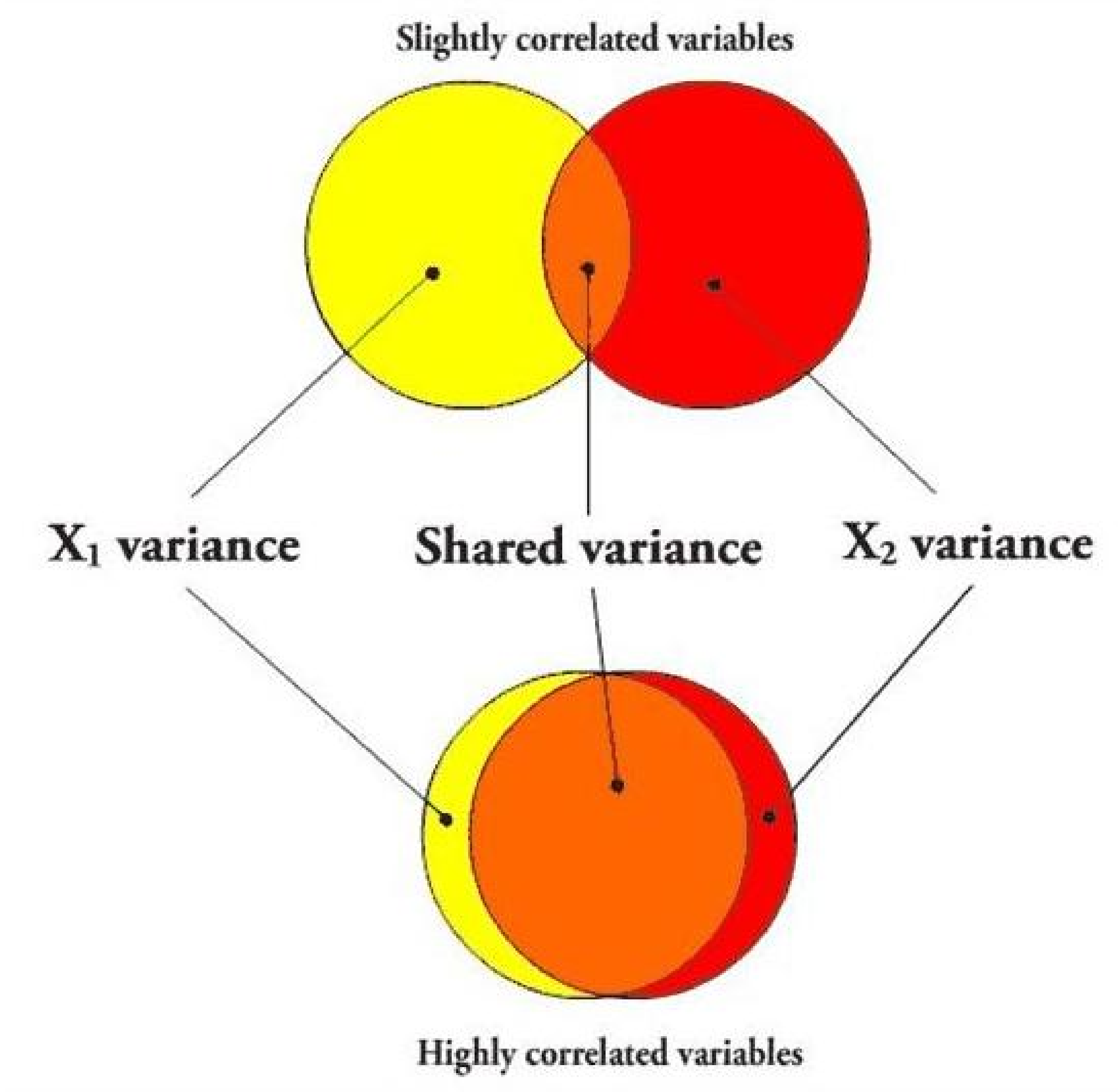
Model selection: regression models

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



[1] https://eigenblogger.com/2010/03/26/post1426/

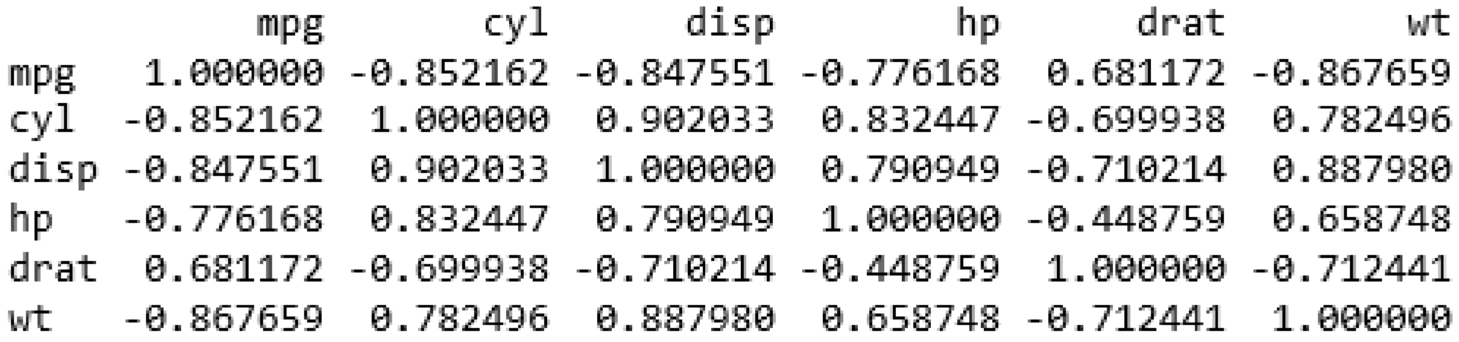
Effects of multicollinearity

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

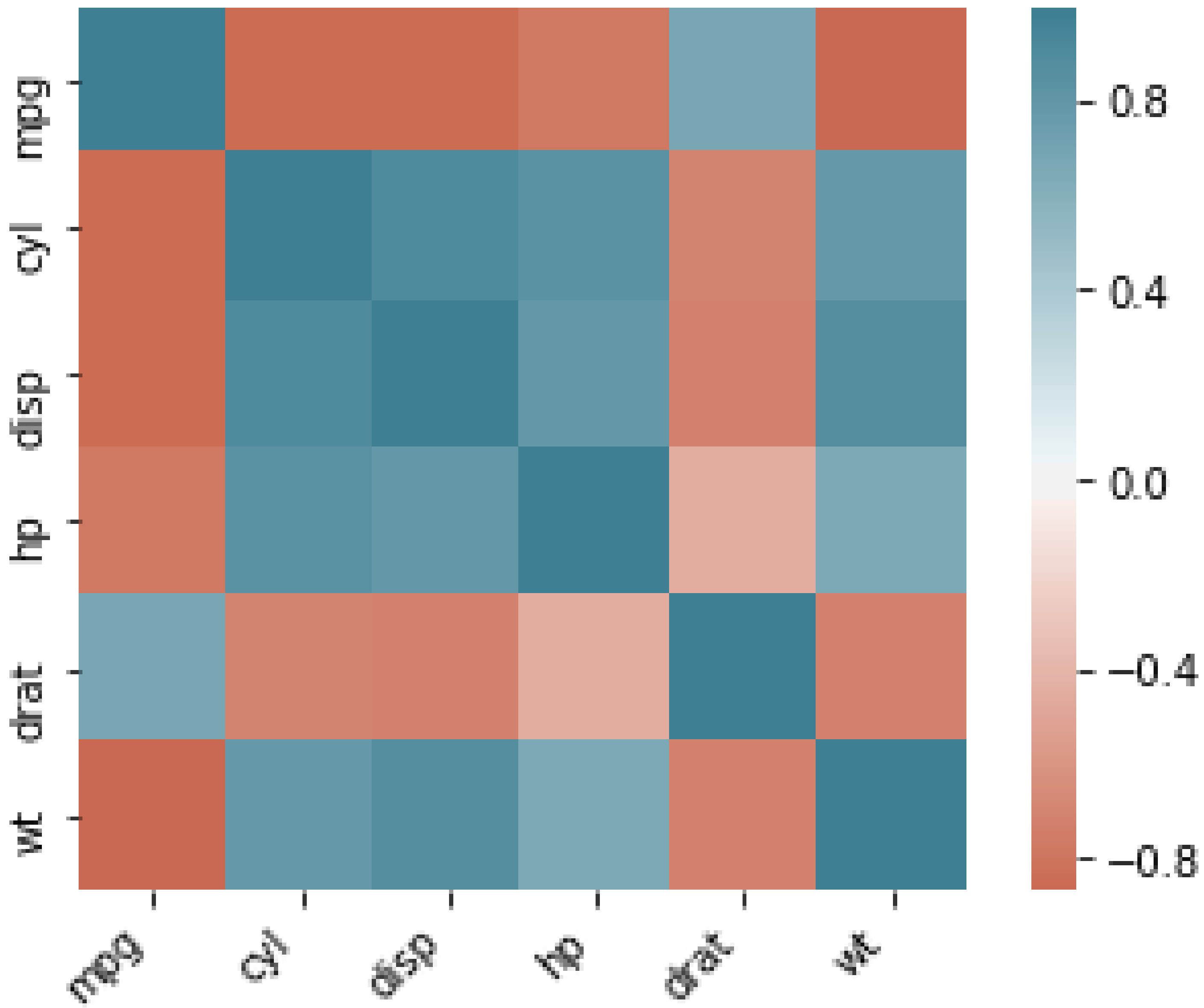
Techniques to address multicollinearity

- 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
<= 1	no
> 1	yes, but can ignore
> 5	yes, need to address

Functions

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

Let's practice!

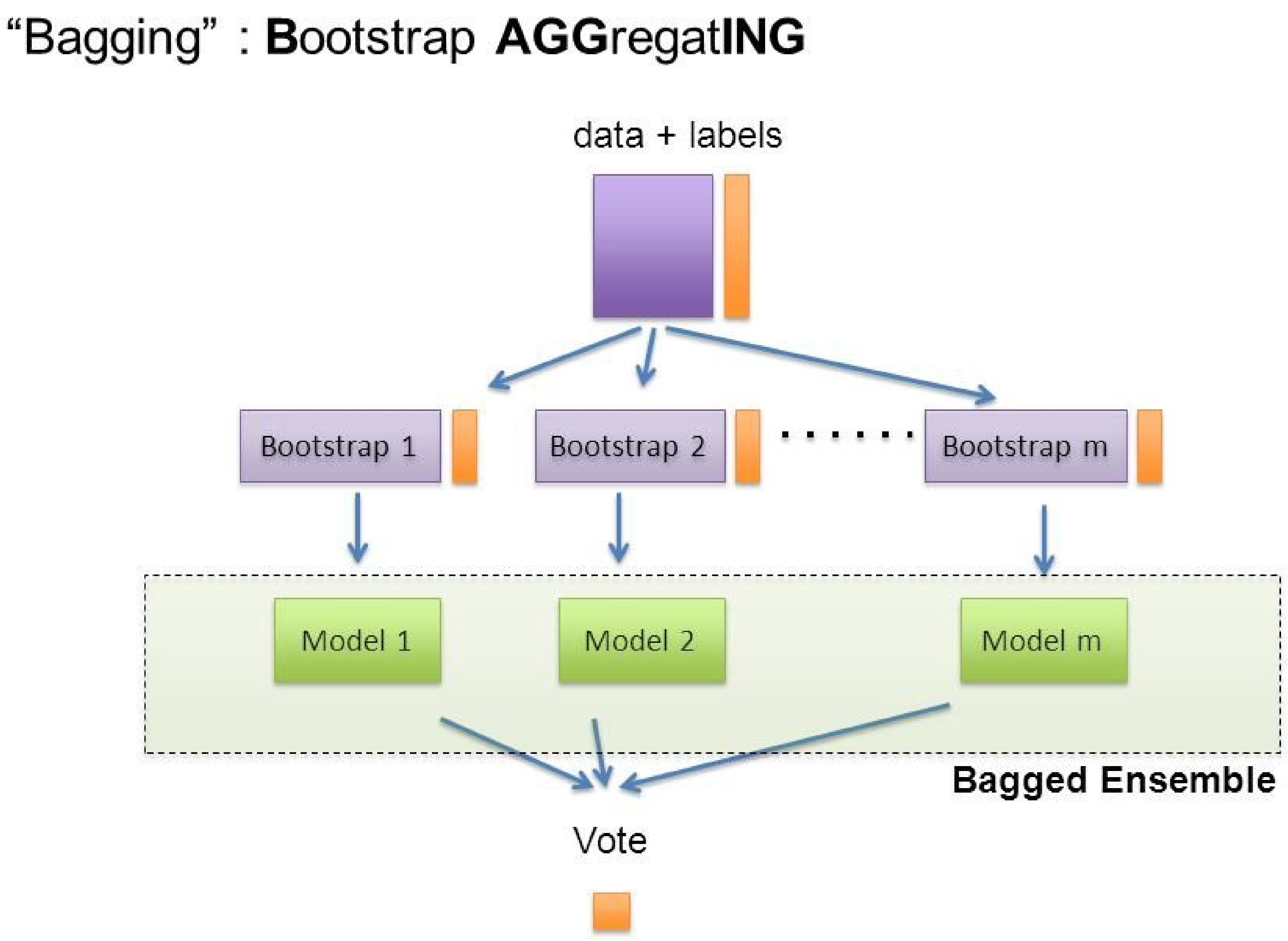
PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Model selection: ensemble models

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Lisa Stuart
Data Scientist

Bootstrapping

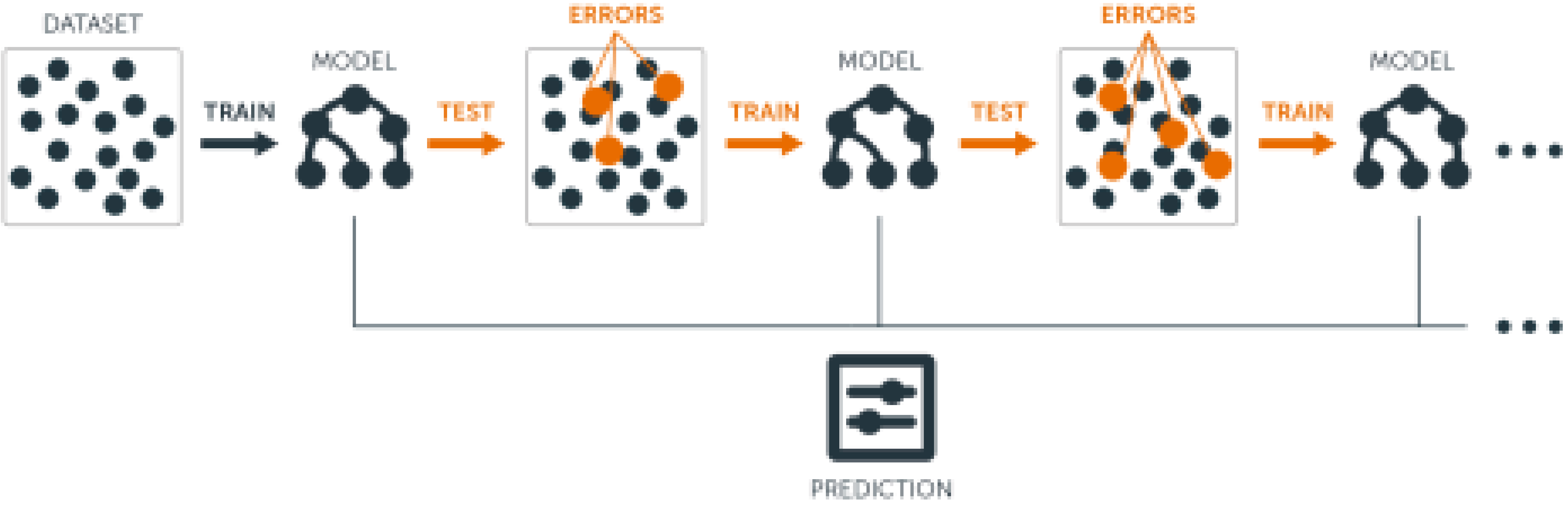


[1] https://medium.com/@rrfd/boosting-bagging-and-stacking-ensemble-methods-with-sklearn-and-mlens-a455c0c982de

Random forest

[1] https://www.sca.com/en/about-us/our-forest/

Gradient Boosting



[1] 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`

[1] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble

Functions

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

Let's practice!

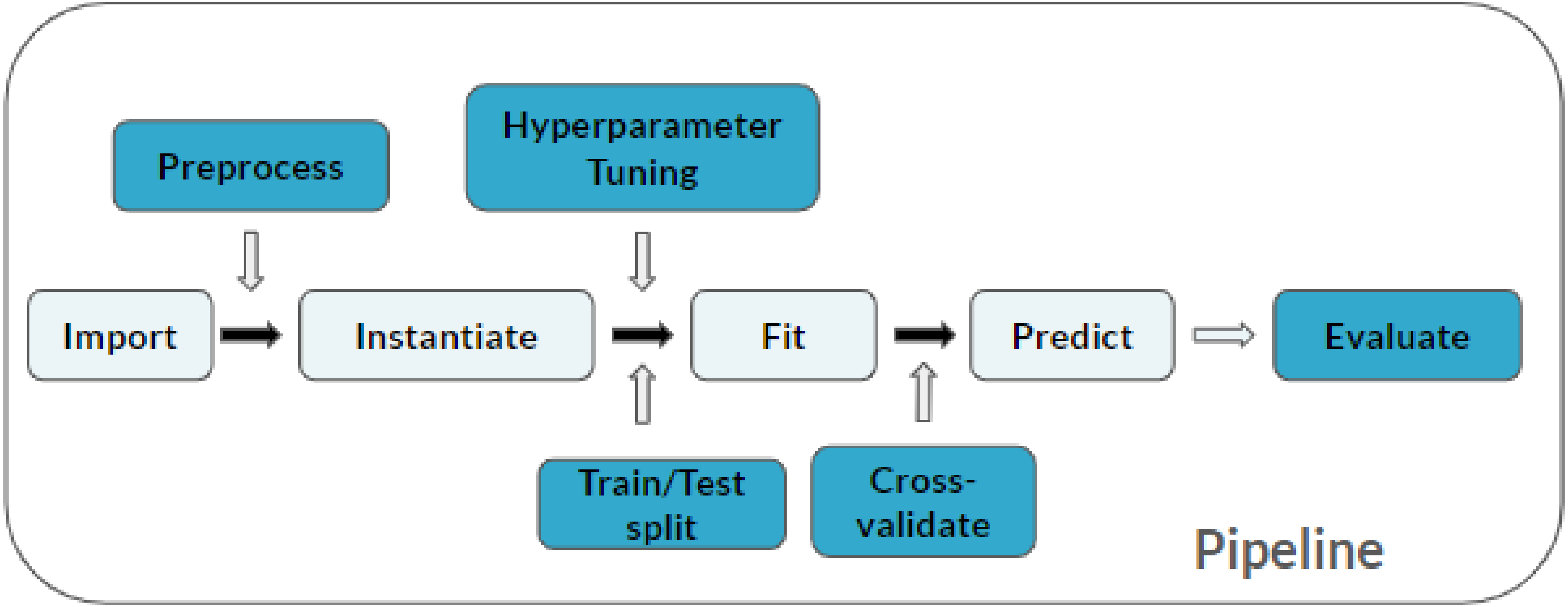
PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Wrap-Up

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Lisa Stuart
Data Scientist

Machine Learning Pipeline



Preprocess

Hyperparameter Tuning

Import

Instantiate

Fit

Predict

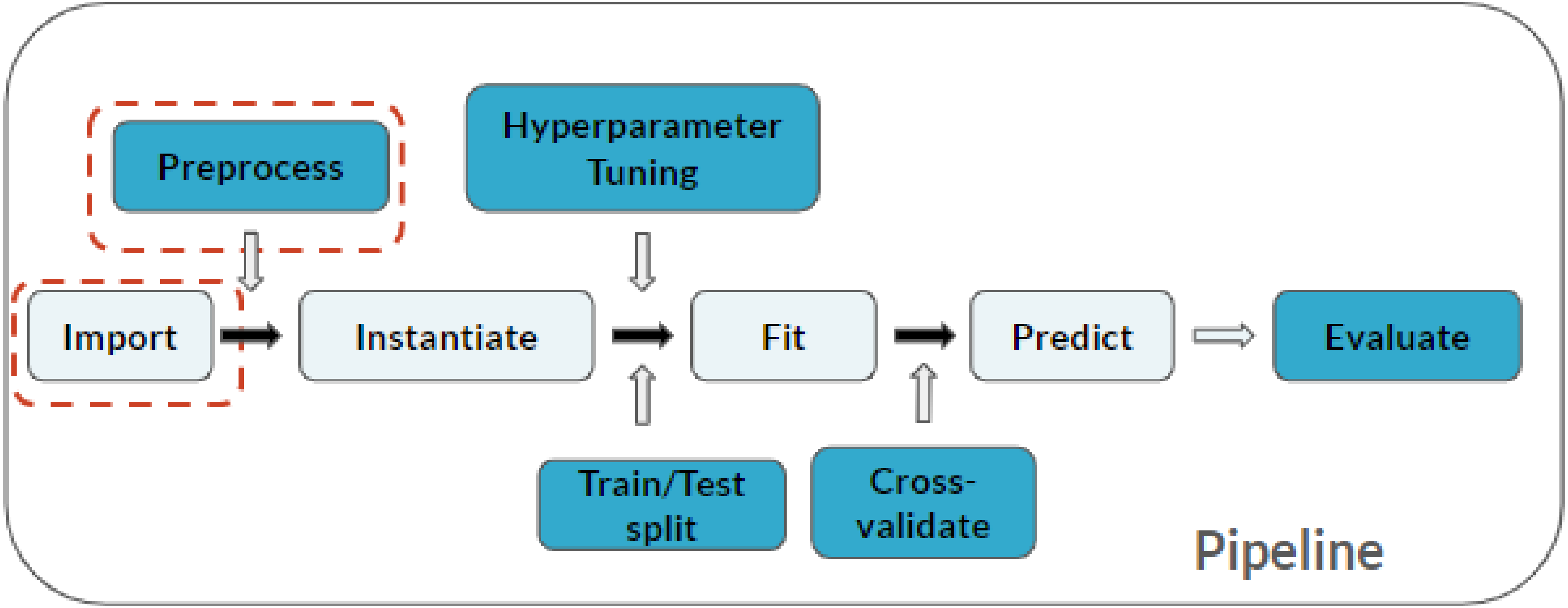
Evaluate

Train/Test split

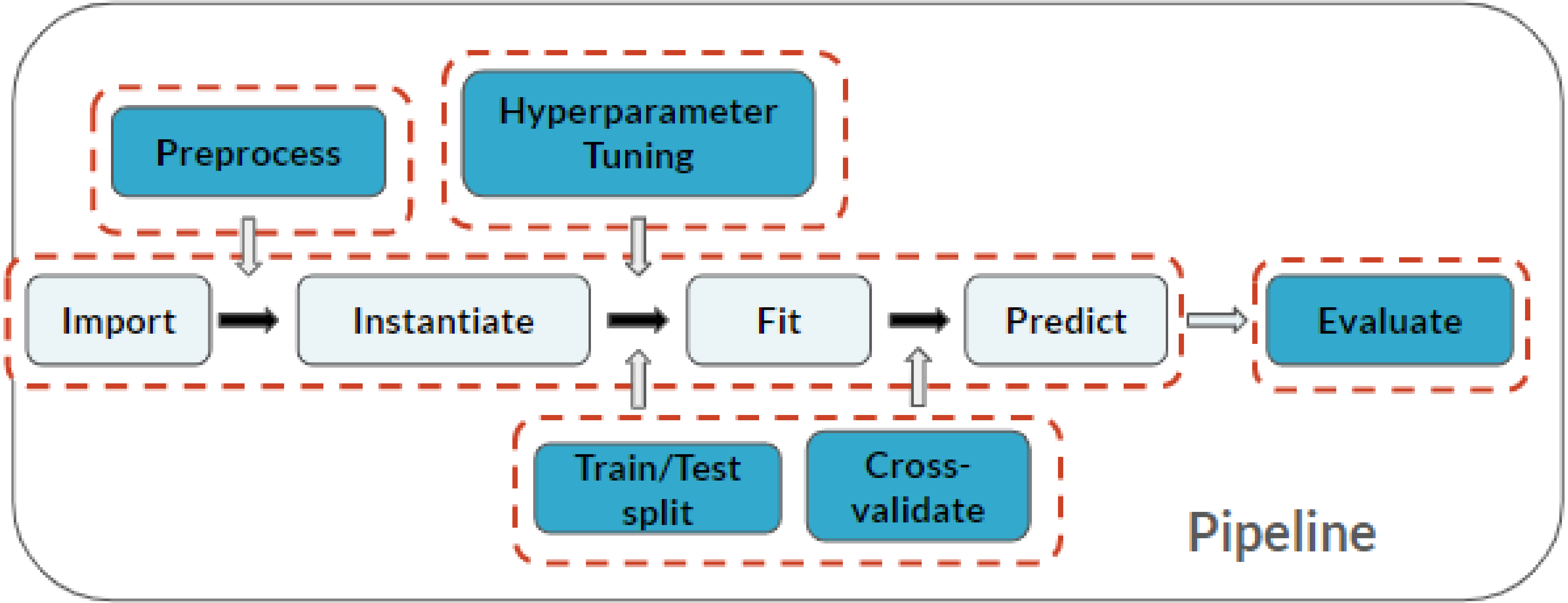
Cross-validate

Pipeline

Machine Learning Pipeline



Machine Learning Pipeline



CONGRATULATIONS!!!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON