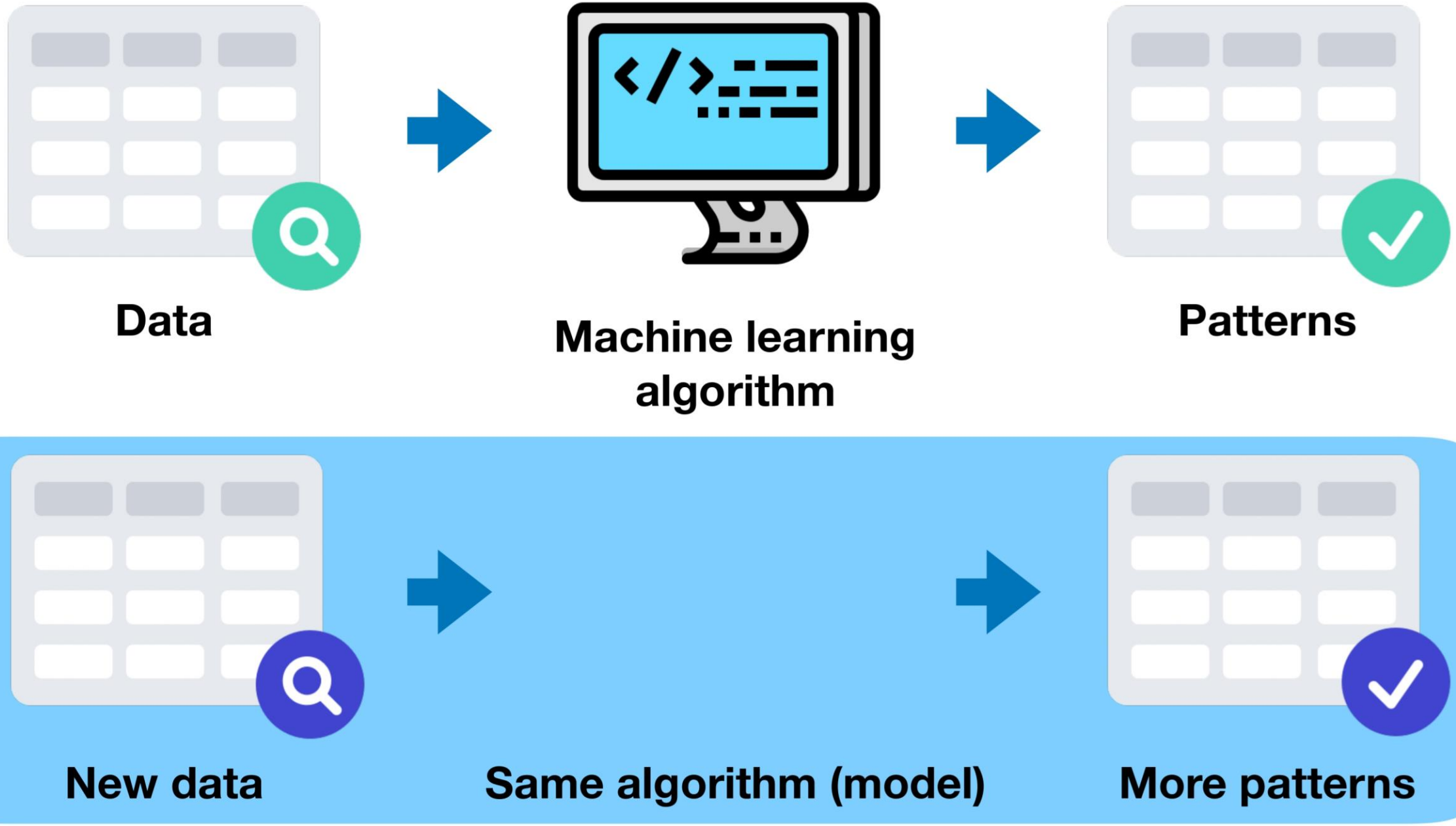
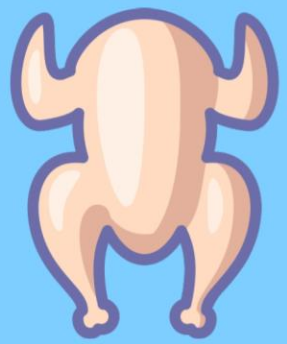
What is machine learning?


Data
Machine learning algorithm
Patterns
Future
New data
Same algorithm (model)
More patterns

Normal algorithm



1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

Starts with

Makes

Machine learning algorithm

Inputs

Output

Starts with

1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

Figures out

Attempt

1

1. Cut vegetables
2. Season chicken with lots of spice
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

2

1. Cut vegetables
2. Season chicken with extra spice
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

3

1. Cut vegetables
2. Season chicken a lil' extra spice
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

4

1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

⋮

1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

100

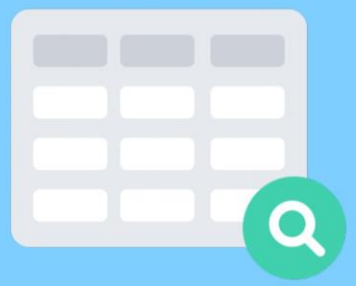
1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

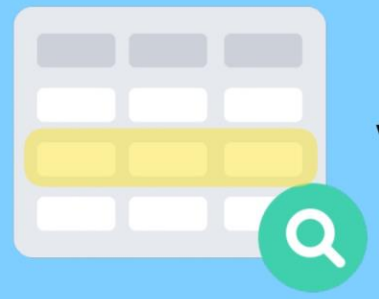
Machine learning algorithms may try 1000's of times to find the right instructions.

Data science

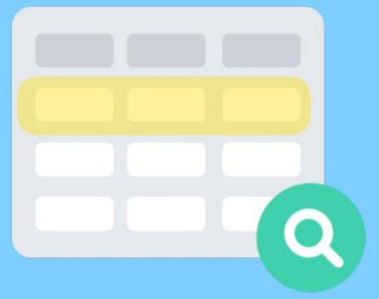
Data analysis

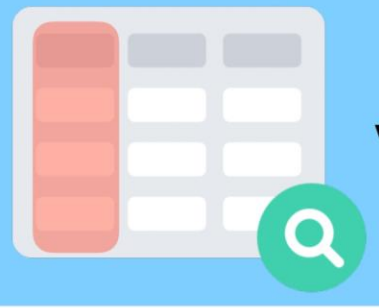
Data



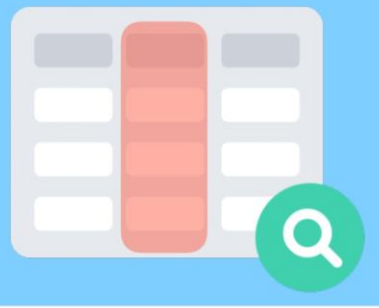


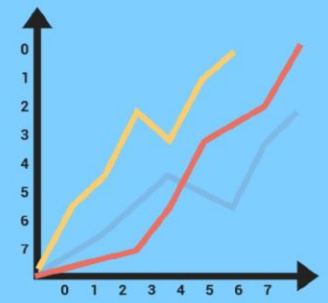
vs.



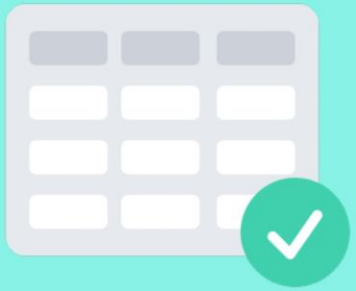


vs.





Machine learning



1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-minutes
5. Add vegetables

What we're going to cover (and what you'll finish with)

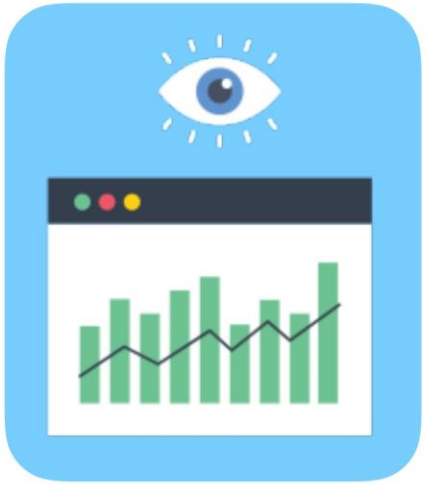
We're focused on...

- Practical solutions
- Writing machine learning code

Steps in a full machine learning project

1. Problem definition

“What problem are we trying to solve?”



Supervised

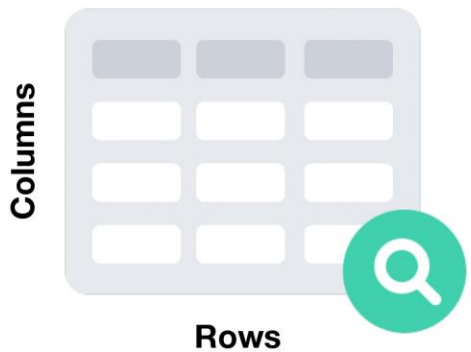
Unsupervised

Classification

Regression

2. Data

“What kind of data do we have?”

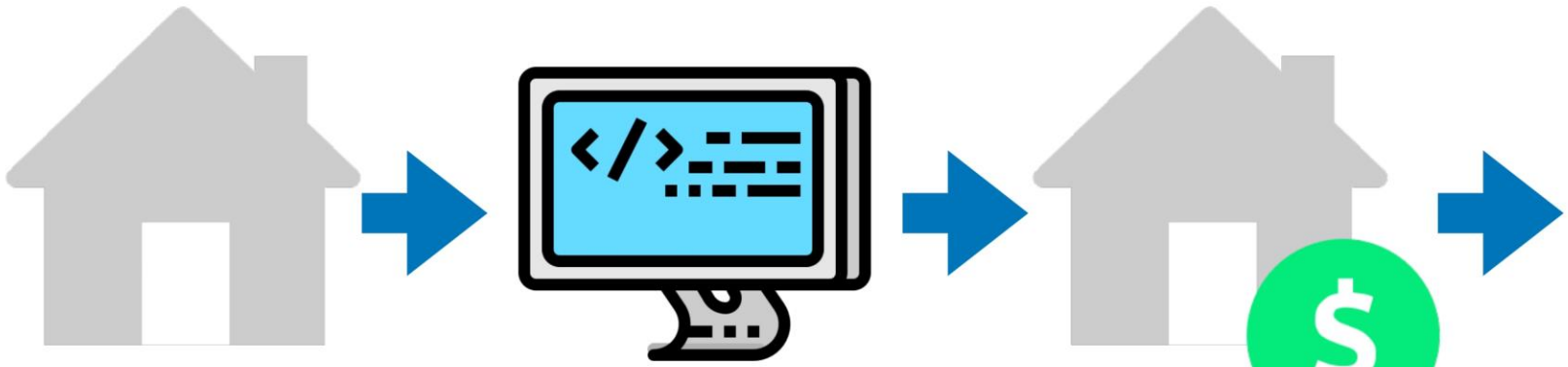


Structured

Unstructured

3. Evaluation

"What defines success for us?"



House data

Machine learning model

House price

Predicted price

$497,890

Actual price

$500,000

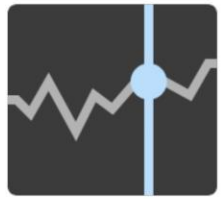
4. Features

“What do we already know about the data?”

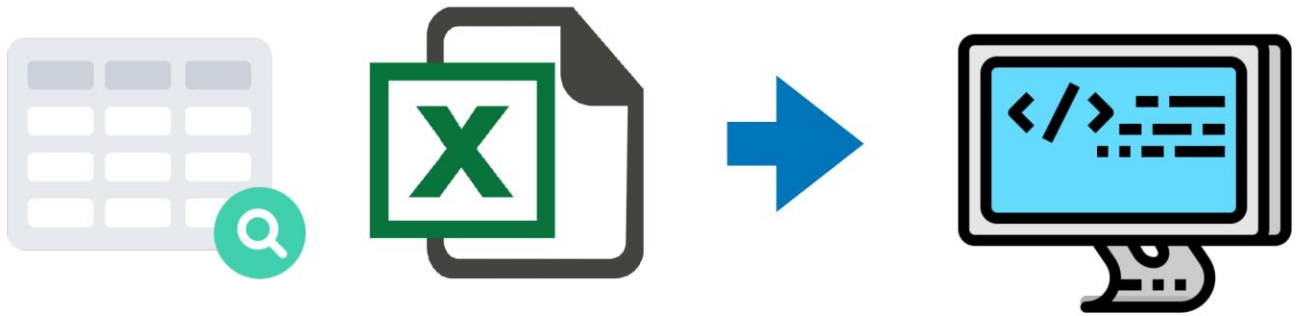
ID	Weight	Sex	Blood Pressure	Chest pain	Heart disease?
4326	110kg	M	120/80	4	YES
5681	64kg	F	130/90	1	NO
7911	81kg	M	130/80	0	NO

Table 1.0: Patient records

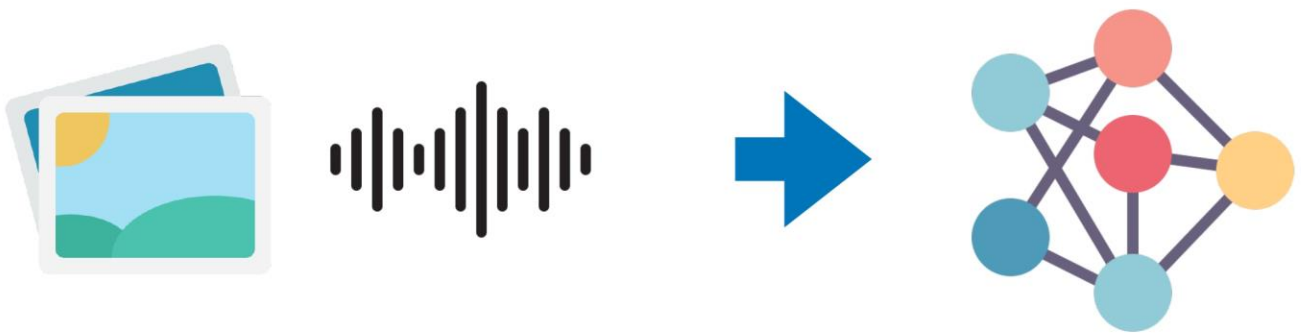
5. Modelling



“Based on our problem and data, what model should we use?”



Problem 1 **Model 1**



Problem 2 **Model 2**

6. Experimentation

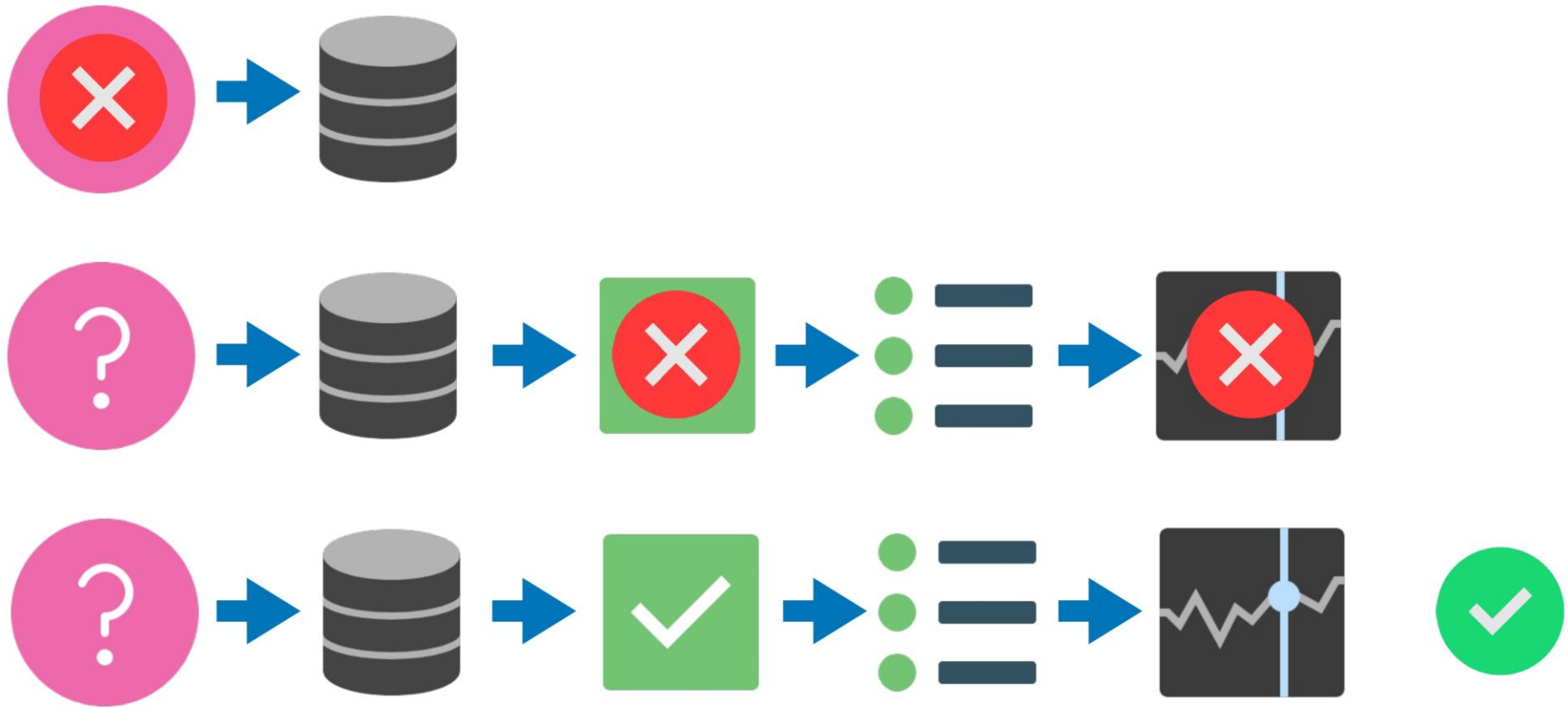
"How could we improve/what can we try next?"

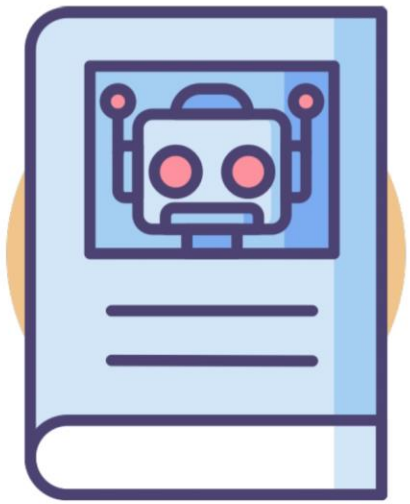
Attempt

1

2

3





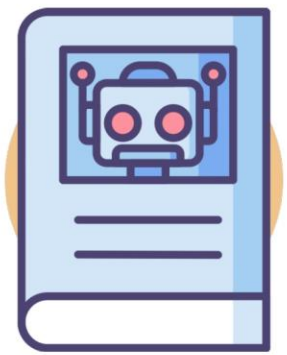
1. Create a framework

2. Match to data science and machine learning tools

3. Learn by doing

Yes	No
Write code	Overthink the process
Make mistakes	Try make things perfect
Build projects	Build things from scratch
Learn what matters	

The framework we'll be using



Your computer

ANACONDA

jupyter

matplotlib

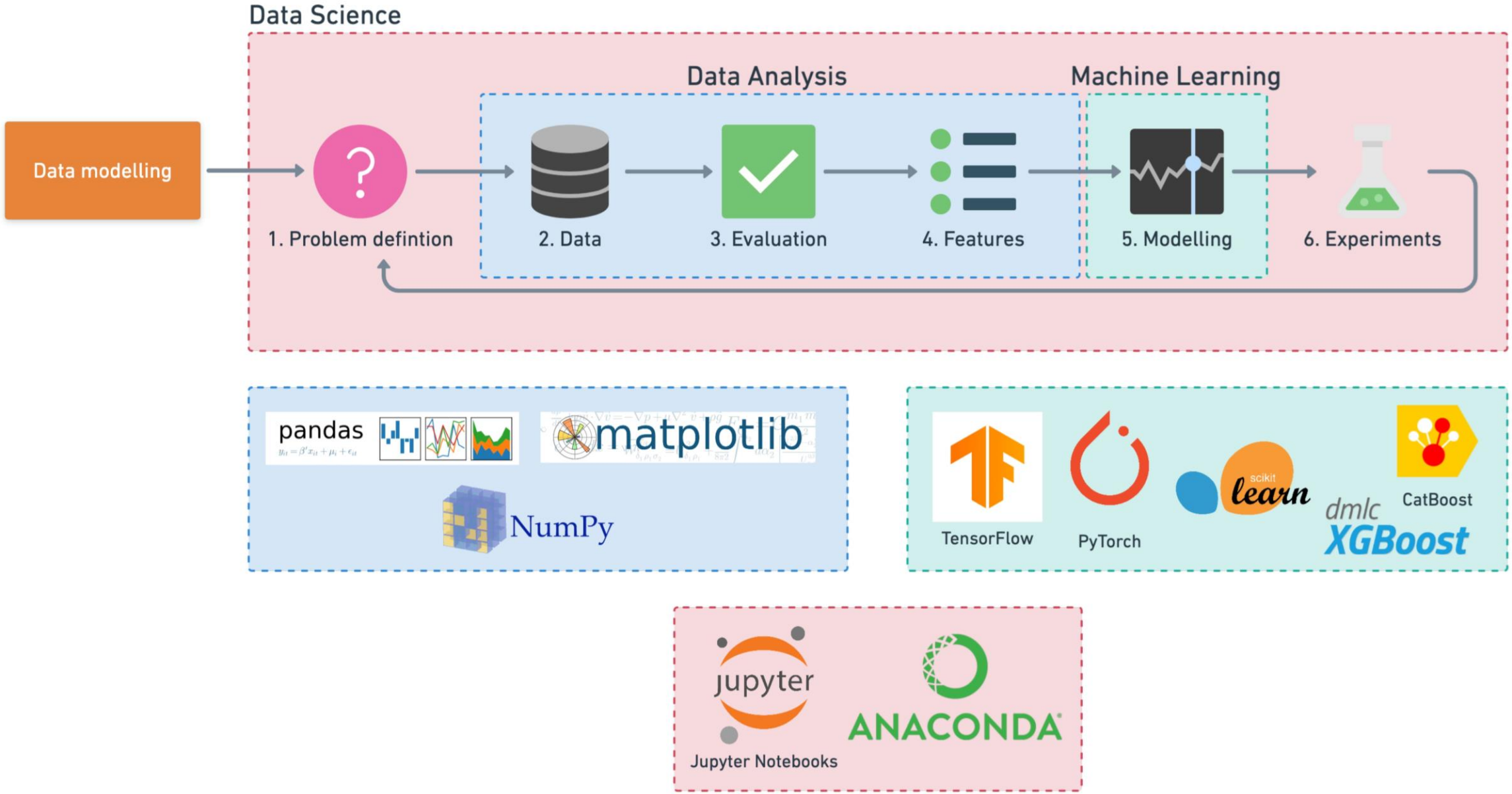
NumPy

pandas

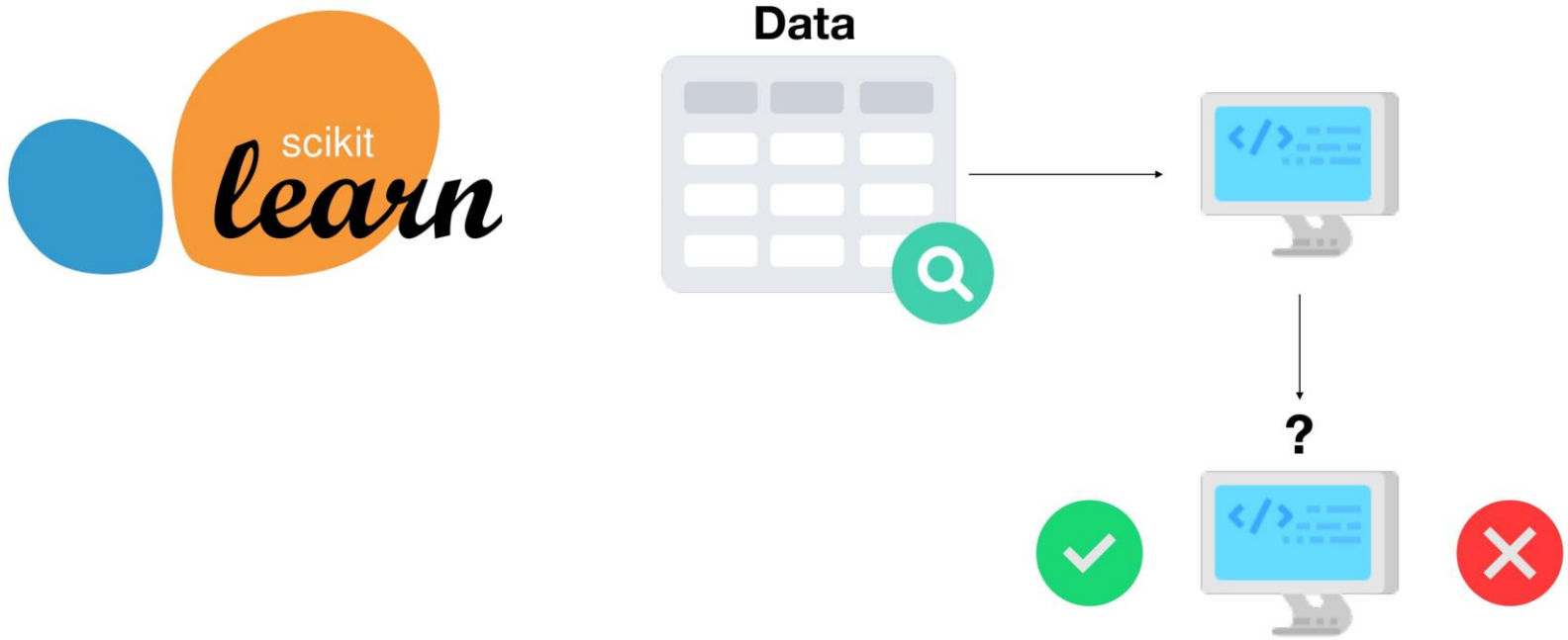
scikit learn

dmlc XGBoost

Tools you can use



What is Scikit-Learn (sklearn)?



Why Scikit-Learn?

- Built on NumPy and Matplotlib (and Python)
- Has many in-built machine learning models
- Methods to evaluate your machine learning models
- Very well-designed API

What are we going to cover?

A Scikit-Learn workflow

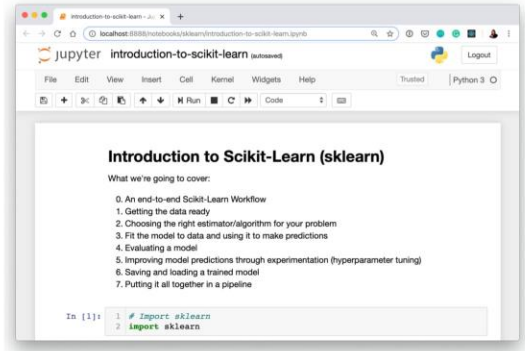
1. Get data ready
2. Pick a model (to suit your problem)
3. Fit the model to the data and make a prediction
4. Evaluate the model
5. Improve through experimentation
6. Save and reload your trained model

What are we going to cover?

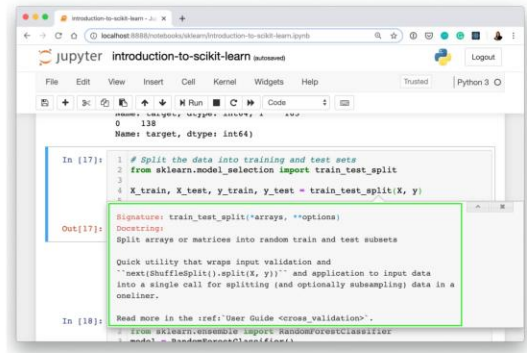
- An end-to-end Scikit-Learn workflow
- Getting data ready (to be used with machine learning models)
- Choosing a machine learning model
- Fitting a model to the data (learning patterns)
- Making predictions with a model (using patterns)
- Evaluating model predictions
- Improving model predictions
- Saving and loading models

Where can you get help?

- Follow along with the code

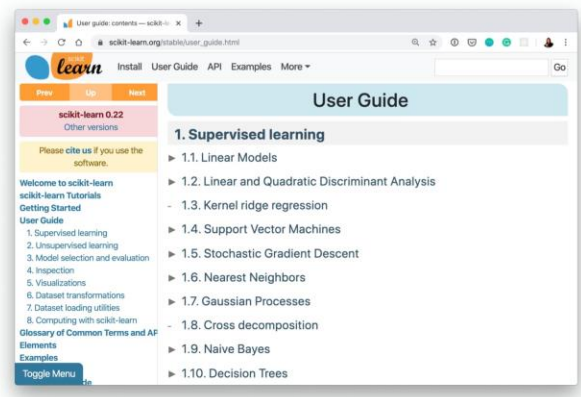


- Try it for yourself
- Press SHIFT + TAB to read the docstring



- Search for it

- Try again
- Ask



Let's model!

Supervised learning

Classification

- "Is this example one thing or another?"
- Binary classification = two options
- Multi-class classification = more than two options

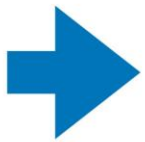
Regression

- "How much will this house sell for?"
- "How many people will buy this app?"

One Hot Encoding

A process used to turn categories into numbers.

Car	Colour
0	Red
1	Green
2	Blue
3	Red



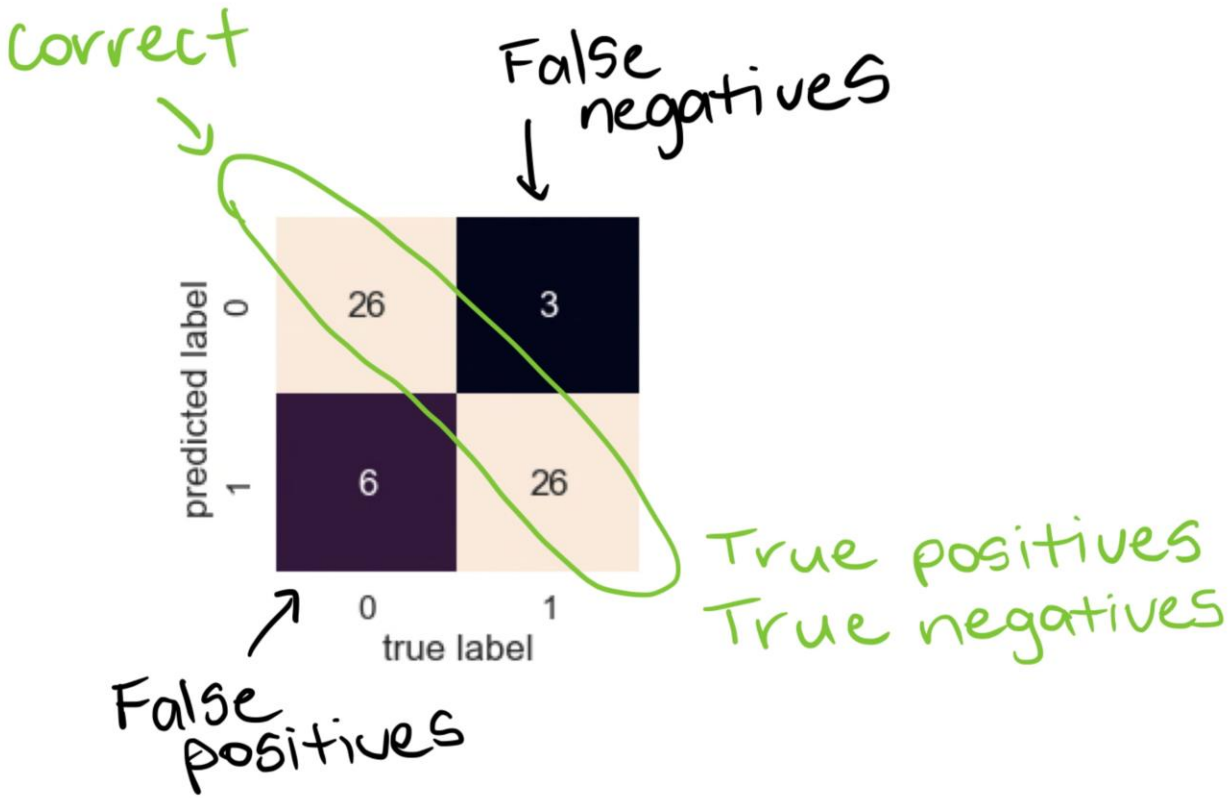
Car	Red	Green	Blue
0	1	0	0
1	0	1	0
2	0	0	1
3	1	0	0

Classification and Regression metrics

Classification	Regression
Accuracy	**R^2 (r-squared)**
Precision	Mean absolute error (MAE)
Recall	Mean squared error (MSE)
F1	Root mean squared error (RMSE)

Bold = default evaluation in Scikit-Learn

Confusion matrix anatomy



- True positive = model predicts 1 when truth is 1
- False positive = model predicts 1 when truth is 0
- True negative = model predicts 0 when truth is 0
- False negative = model predicts 0 when truth is 1

Classification report anatomy

```python
from sklearn.metrics import classification_report

print(classification_report(y_test, y_preds))
```

```
              precision    recall  f1-score   support

           0       0.81      0.90      0.85        29
           1       0.90      0.81      0.85        32

    accuracy                           0.85        61
   macro avg       0.85      0.85      0.85        61
weighted avg       0.86      0.85      0.85        61
```

- **Precision** - Indicates the proportion of positive identifications (model predicted class 1) which were actually correct. A model which produces no false positives has a precision of 1.0.
- **Recall** - Indicates the proportion of actual positives which were correctly classified. A model which produces no false negatives has a recall of 1.0.
- **F1 score** - A combination of precision and recall. A perfect model achieves an F1 score of 1.0.
- **Support** - The number of samples each metric was calculated on.
- **Accuracy** - The accuracy of the model in decimal form. Perfect accuracy is equal to 1.0.
- **Macro avg** - Short for macro average, the average precision, recall and F1 score between classes. Macro avg doesn't class imbalance into effort, so if you do have class imbalances, pay attention to this metric.
- **Weighted avg** - Short for weighted average, the weighted average precision, recall and F1 score between classes. Weighted means each metric is calculated with respect to how many samples there are in each class. This metric will favour the majority class (e.g. will give a high value when one class out performs another due to having more samples).

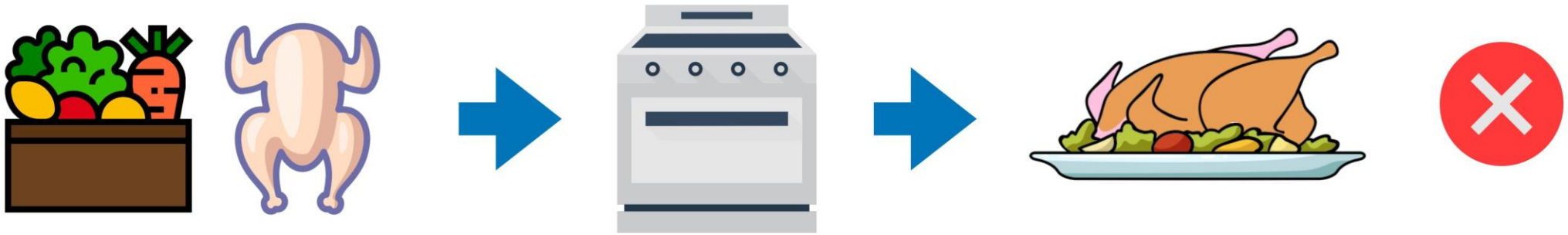
Which classification metric should you use?

- **Accuracy** is a good measure to start with if all classes are balanced (e.g. same amount of samples which are labelled with 0 or 1).
- **Precision** and **recall** become more important when classes are imbalanced.
- If false positive predictions are worse than false negatives, aim for higher precision.
- If false negative predictions are worse than false positives, aim for higher recall.
- **F1-score** is a combination of precision and recall.

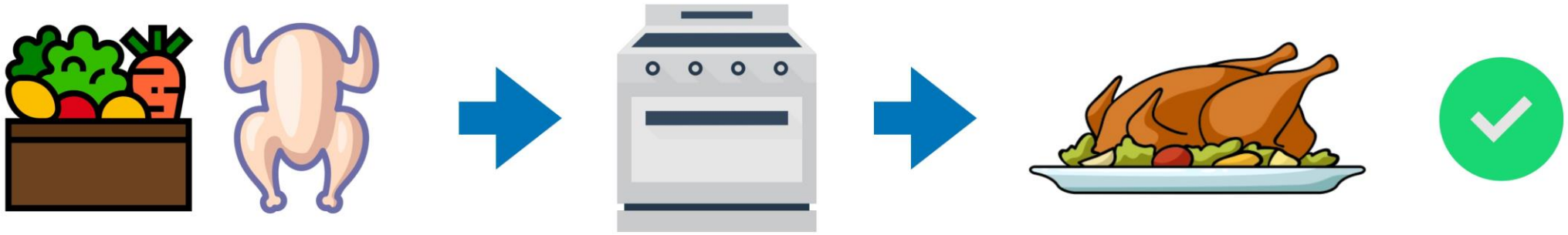
Which regression metric should you use?

- **R^2** is similar to accuracy. It gives you a quick indication of how well your model might be doing. Generally, the closer your **R^2** value is to 1.0, the better the model. But it doesn't really tell exactly how wrong your model is in terms of how far off each prediction is.
- **MAE** gives a better indication of how far off each of your model's predictions are on average.
- As for **MAE** or **MSE**, because of the way MSE is calculated, squaring the differences between predicted values and actual values, it amplifies larger differences. Let's say we're predicting the value of houses (which we are).
  - Pay more attention to MAE: When being $10,000 off is ***twice*** as bad as being $5,000 off.
  - Pay more attention to MSE: When being $10,000 off is ***more than twice*** as bad as being $5,000 off.

Improving a model (via hyperparameter tuning)

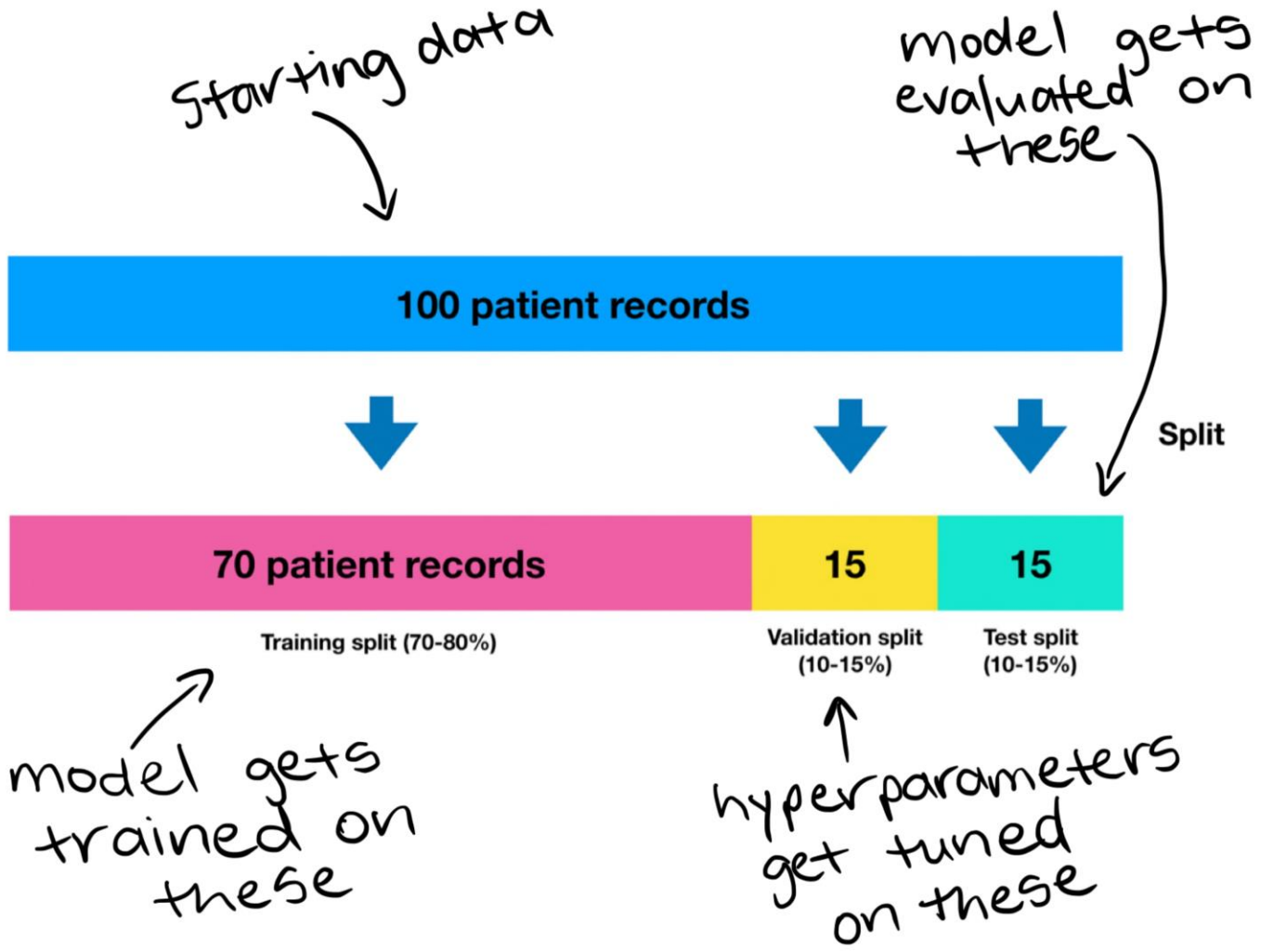


Cooking time: 1 hour
Temperature: 180°C



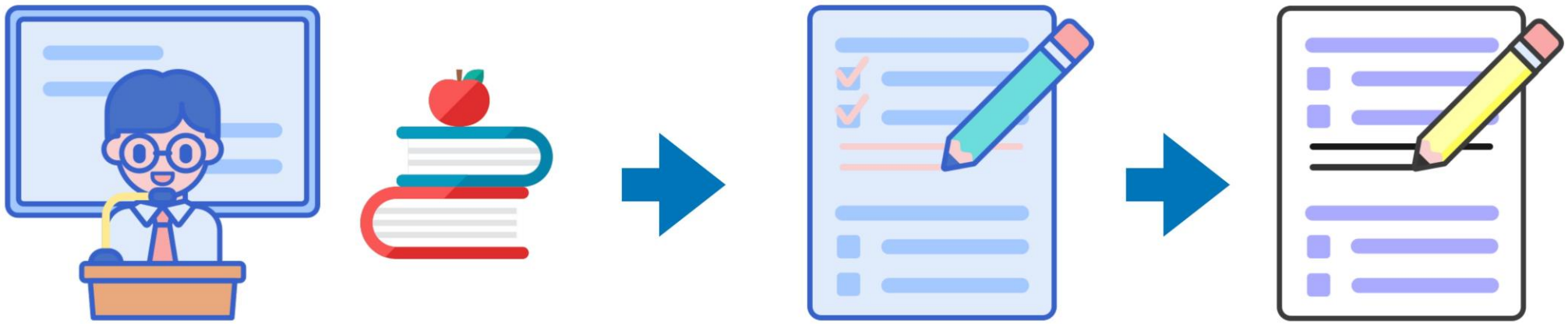
Cooking time: 1 hour
Temperature: 200°C

Tuning Hyperparameters by Hand



The most important concept in machine learning

(the 3 sets)



Course materials (training set)

Practice exam (validation set)

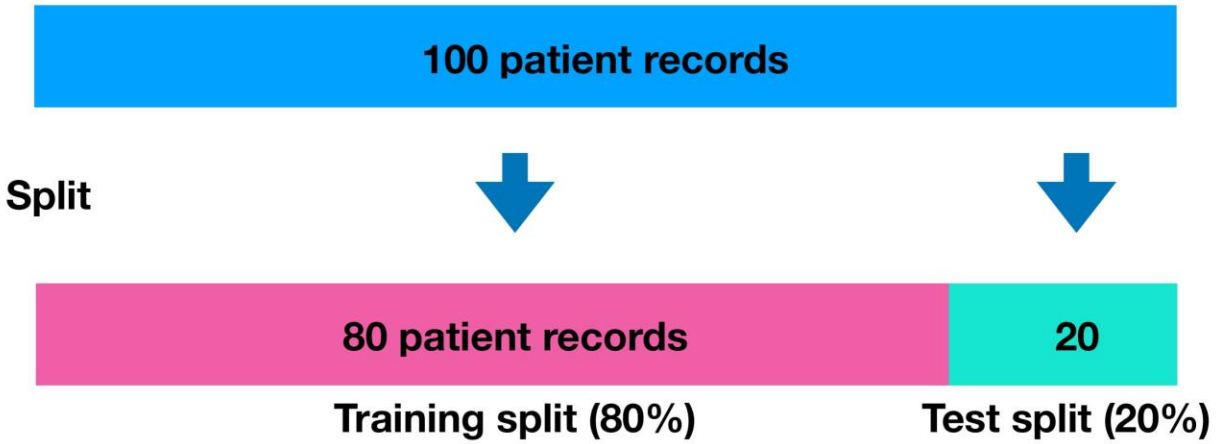
Final exam (test set)

Generalization

The ability for a machine learning model to perform well on data it hasn't seen before.

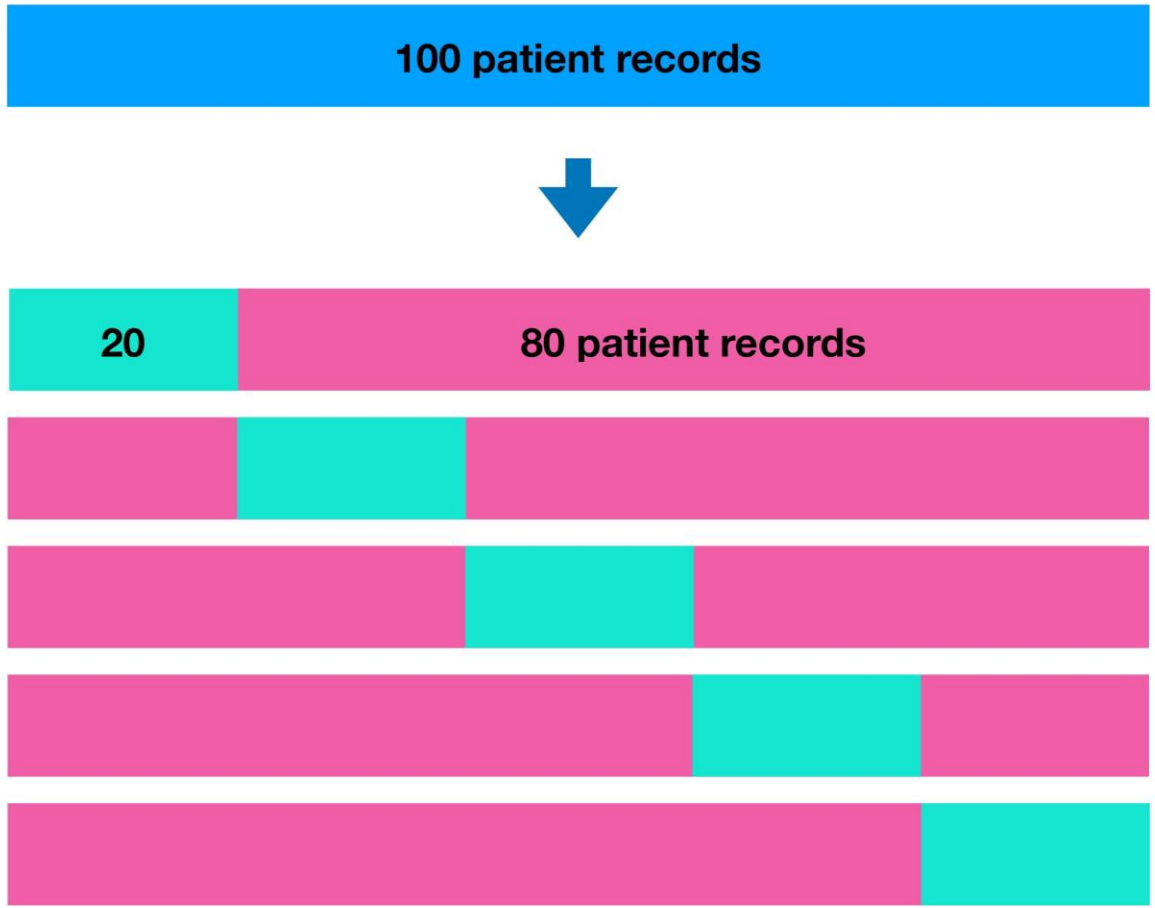
Cross-validation

Normal Train & Test Split



Model is trained on training data, and evaluated on the test data.

5-fold Cross-validation



Model is trained on 5 different versions of training data, and evaluated on 5 different versions of the test data.

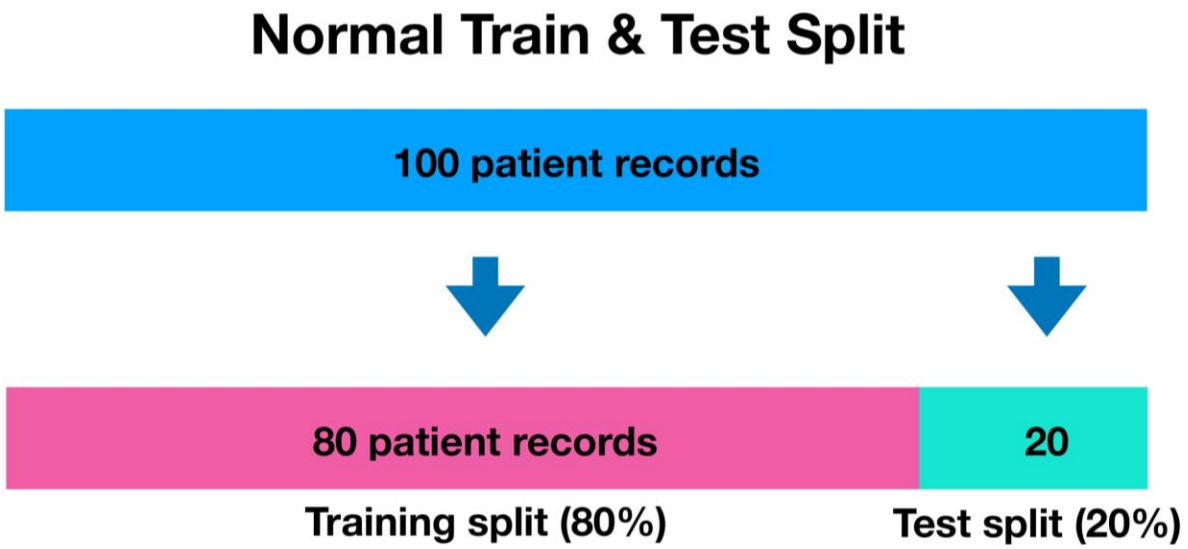


Figure 1.0: Model is trained on training data, and evaluated on the test data.

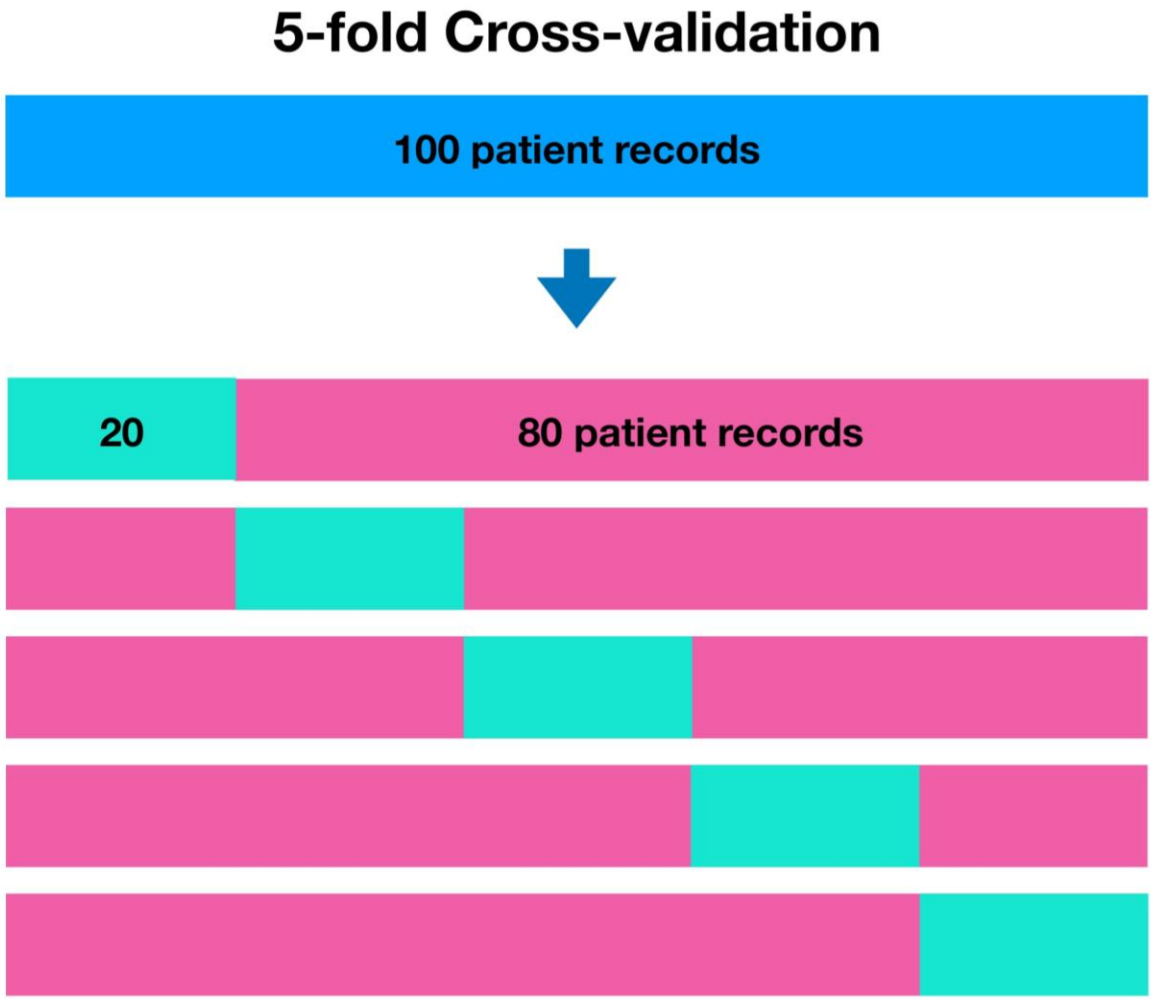


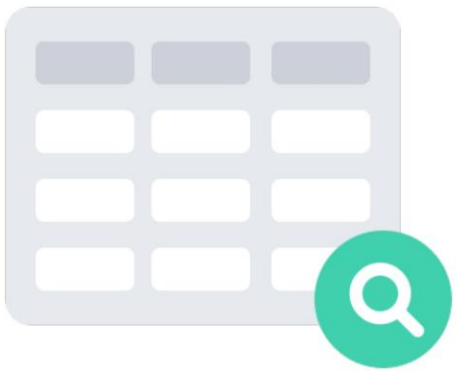
Figure 2.0: Model is trained on training data, and evaluated on the test data.

Things to remember

- All data should be numerical
- There should be no missing values
- Manipulate the test set the same as the training set
- Never test on data you've trained on
- Tune hyperparameters on validation set OR use cross-validation
- One best performance metric doesn't mean the best model

What is structured data?

Data

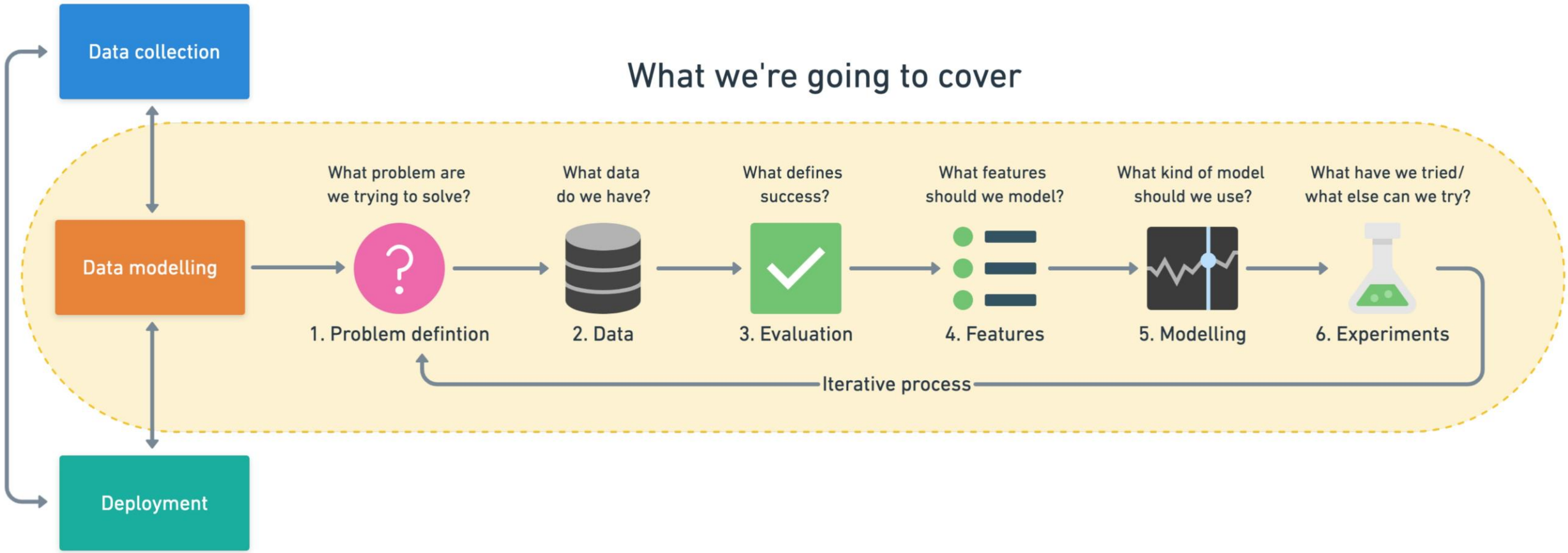


Feature variables | Target

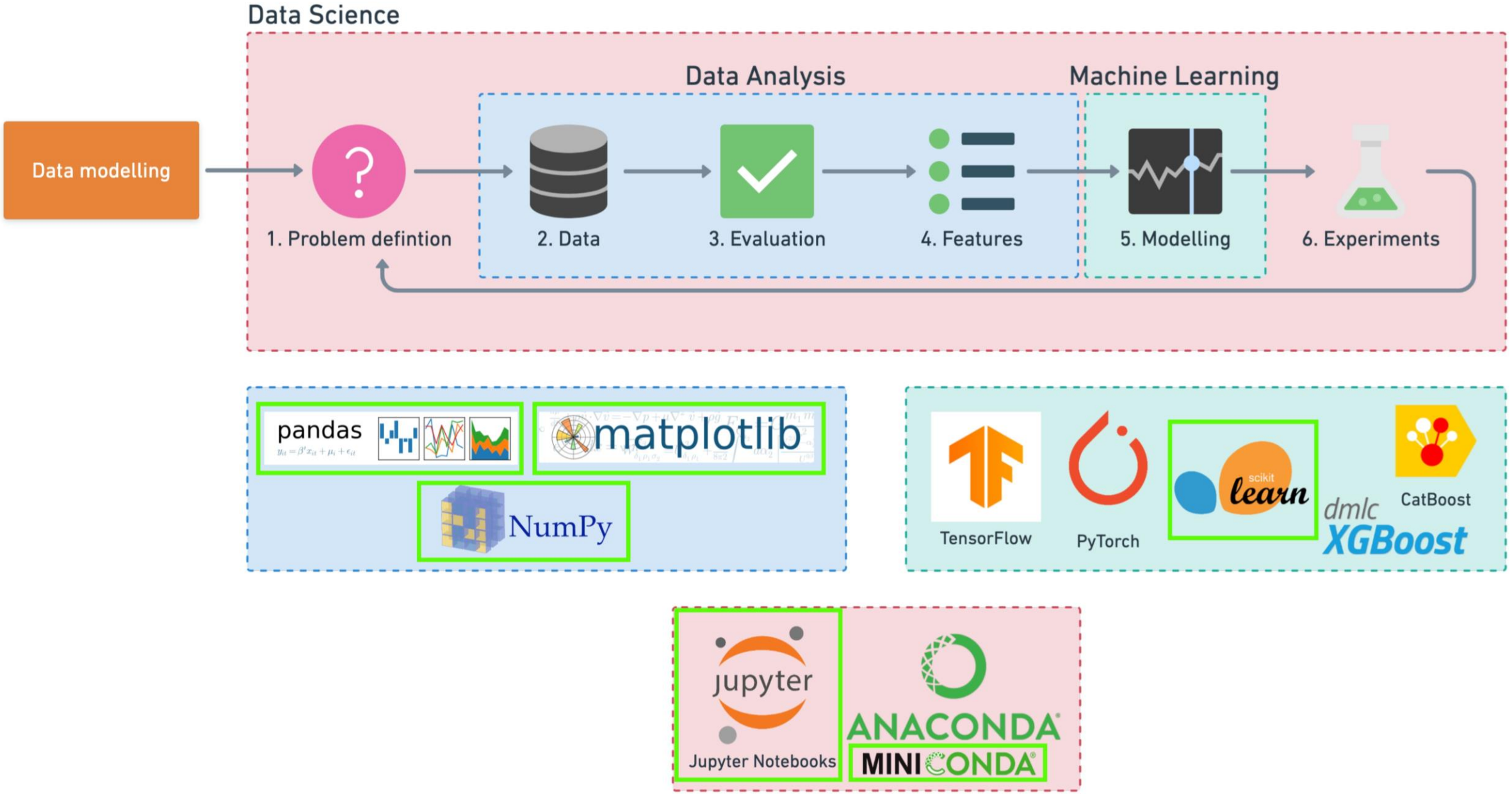
ID	Weight	Sex	Heart Rate	Chest pain	Heart disease?
4326	110kg	M	81	4	YES
5681	64kg	F	61	1	NO
7911	81kg	M	57	0	NO

Table 1.0: Patient records

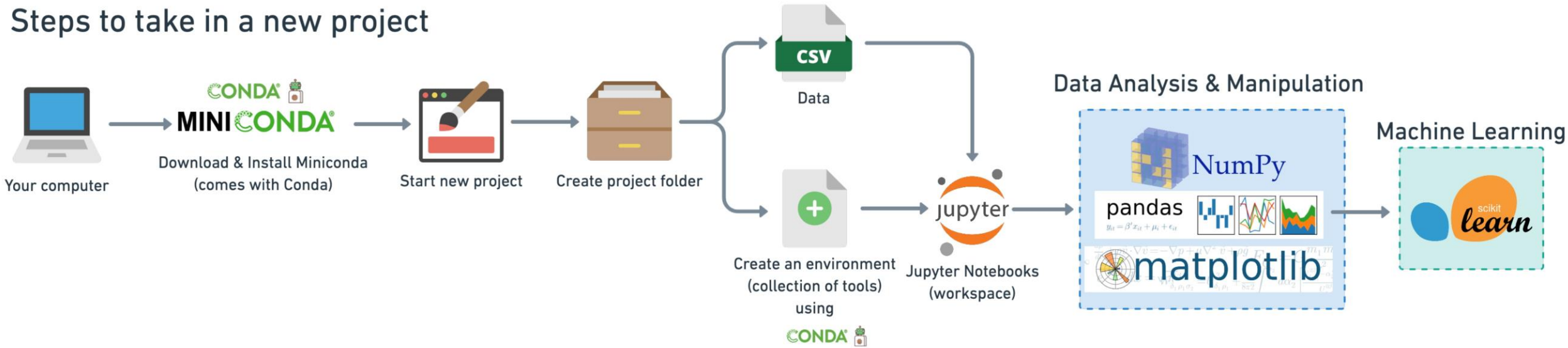
Steps in a full machine learning project



Tools you can use

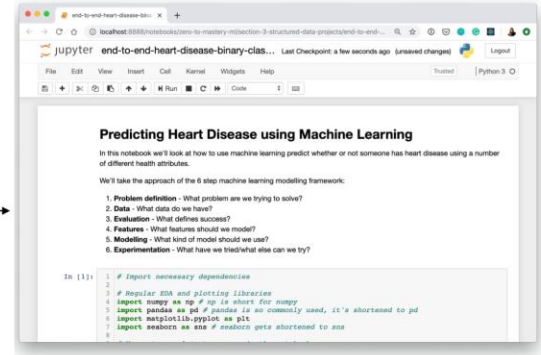


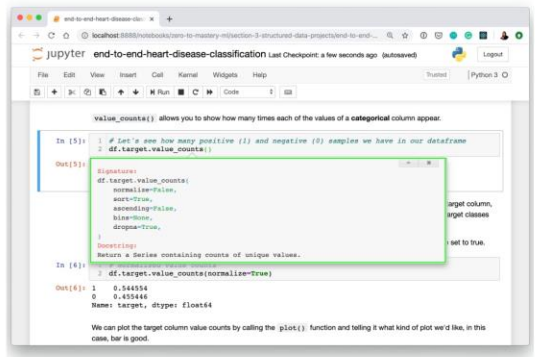
Steps to take in a new project

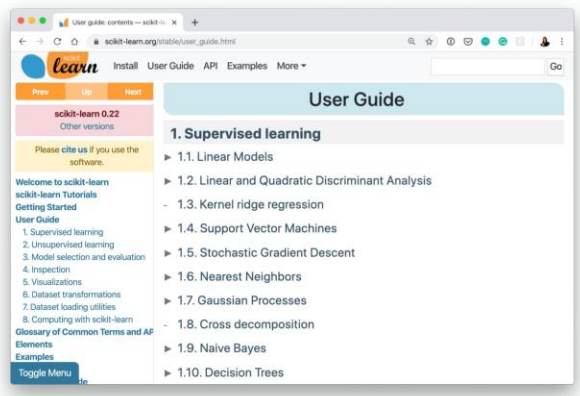


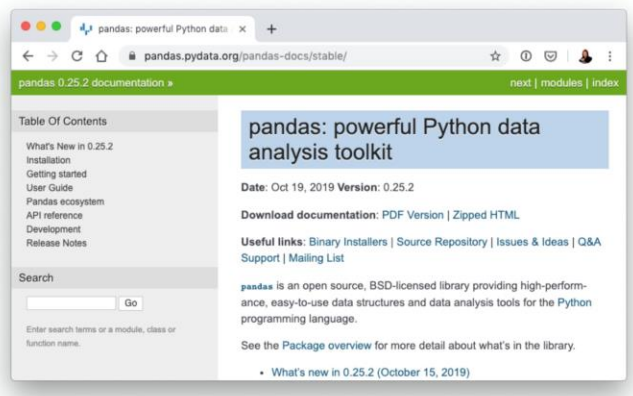
Where can you get help?

- Follow along with the code
- Try it for yourself
- Press SHIFT + TAB to read the docstring
- Search for it
- Try again
- Ask



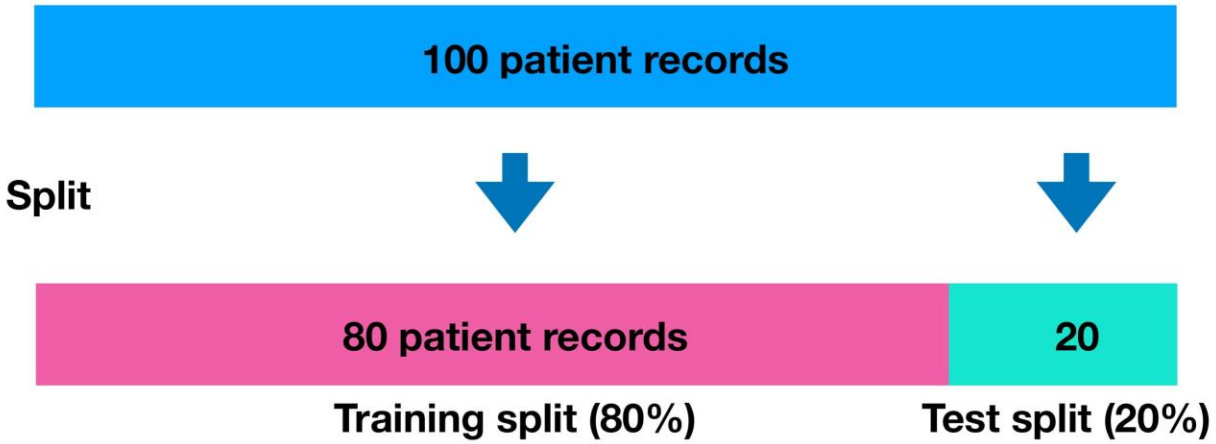






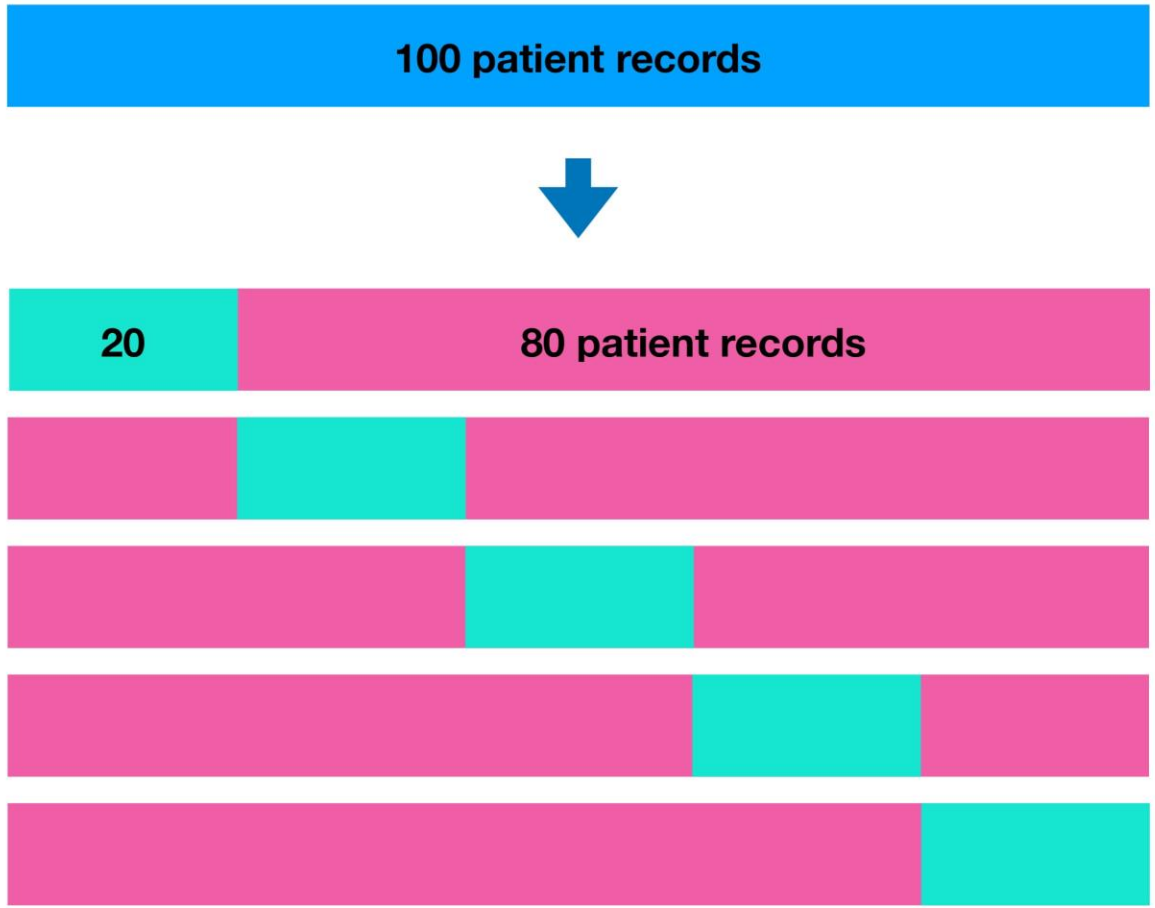
Cross-validation

Normal Train & Test Split



Model is trained on training data, and evaluated on the test data.

5-fold Cross-validation



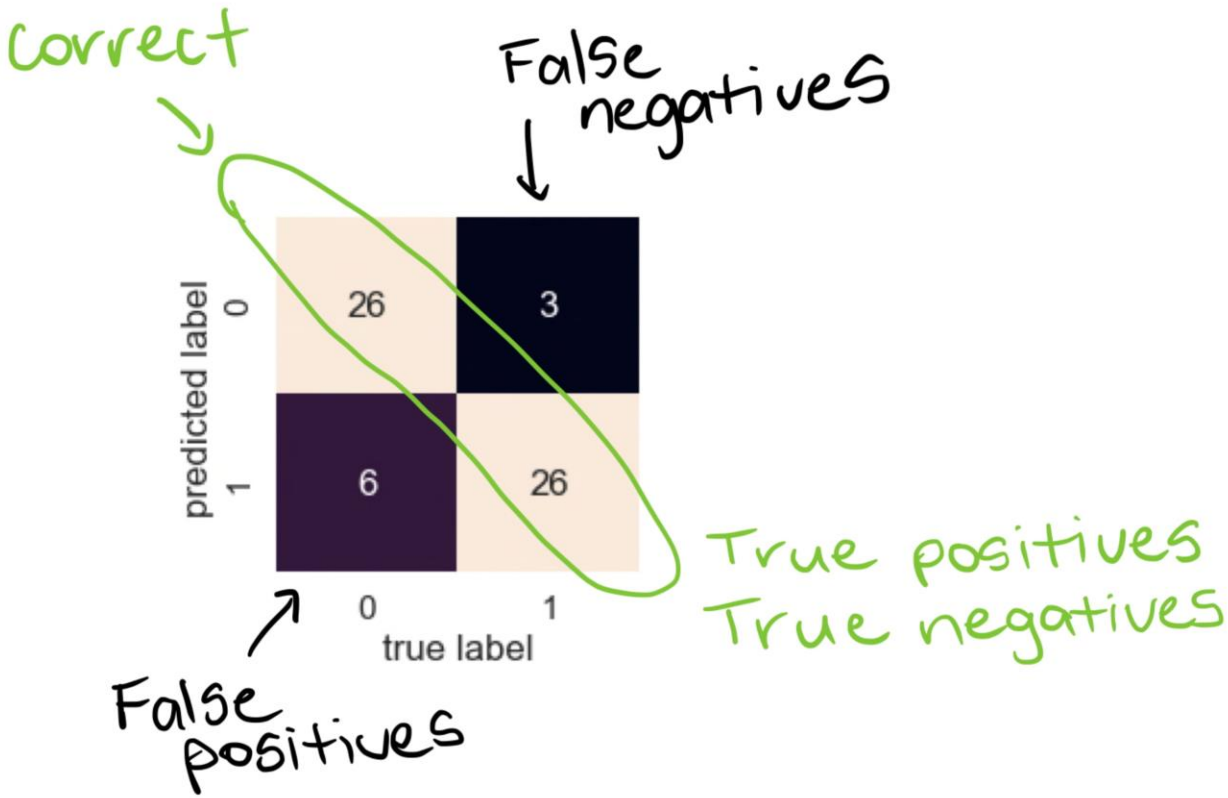
Model is trained on 5 different versions of training data, and evaluated on 5 different versions of the test data.

Classification and Regression metrics

Classification	Regression
Accuracy	**R^2 (r-squared)**
Precision	Mean absolute error (MAE)
Recall	Mean squared error (MSE)
F1	Root mean squared error (RMSE)

Bold = default evaluation in Scikit-Learn

Confusion matrix anatomy



- True positive = model predicts 1 when truth is 1
- False positive = model predicts 1 when truth is 0
- True negative = model predicts 0 when truth is 0
- False negative = model predicts 0 when truth is 1

Classification report anatomy

```python
from sklearn.metrics import classification_report

print(classification_report(y_test, y_preds))
```

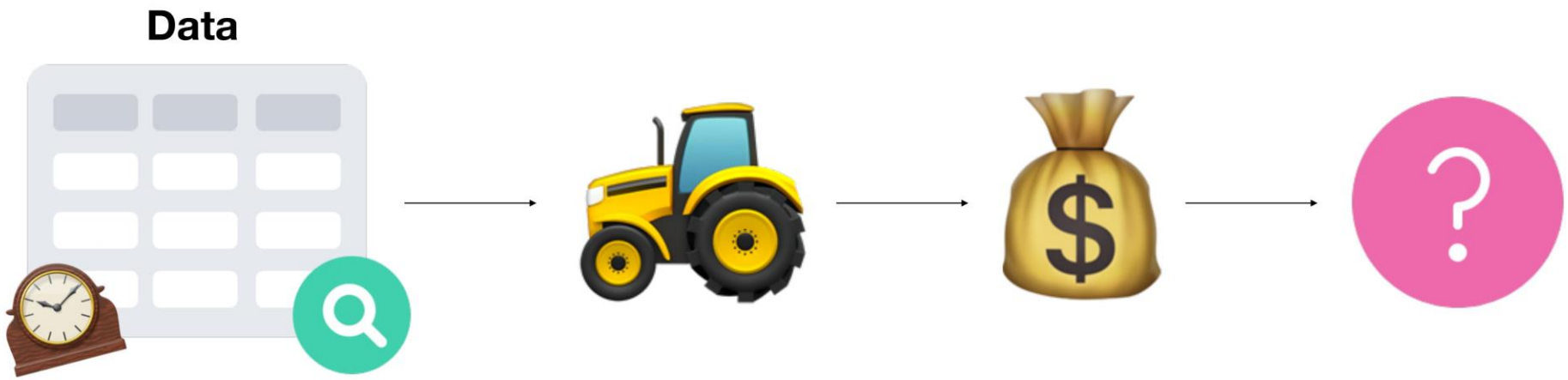
```
              precision    recall  f1-score   support

           0       0.81      0.90      0.85        29
           1       0.90      0.81      0.85        32

    accuracy                           0.85        61
   macro avg       0.85      0.85      0.85        61
weighted avg       0.86      0.85      0.85        61
```

- **Precision** - Indicates the proportion of positive identifications (model predicted class 1) which were actually correct. A model which produces no false positives has a precision of 1.0.
- **Recall** - Indicates the proportion of actual positives which were correctly classified. A model which produces no false negatives has a recall of 1.0.
- **F1 score** - A combination of precision and recall. A perfect model achieves an F1 score of 1.0.
- **Support** - The number of samples each metric was calculated on.
- **Accuracy** - The accuracy of the model in decimal form. Perfect accuracy is equal to 1.0.
- **Macro avg** - Short for macro average, the average precision, recall and F1 score between classes. Macro avg doesn't class imbalance into effort, so if you do have class imbalances, pay attention to this metric.
- **Weighted avg** - Short for weighted average, the weighted average precision, recall and F1 score between classes. Weighted means each metric is calculated with respect to how many samples there are in each class. This metric will favour the majority class (e.g. will give a high value when one class out performs another due to having more samples).

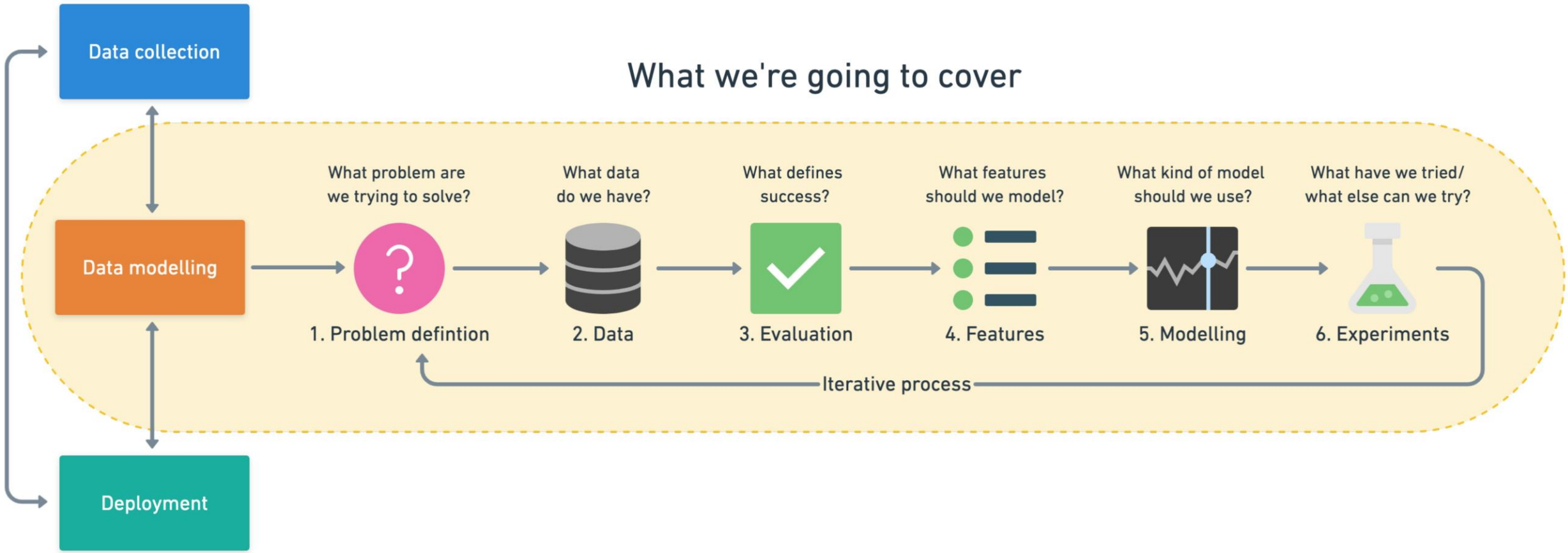
Which classification metric should you use?

- **Accuracy** is a good measure to start with if all classes are balanced (e.g. same amount of samples which are labelled with 0 or 1).
- **Precision** and **recall** become more important when classes are imbalanced.
- If false positive predictions are worse than false negatives, aim for higher precision.
- If false negative predictions are worse than false positives, aim for higher recall.
- **F1-score** is a combination of precision and recall.

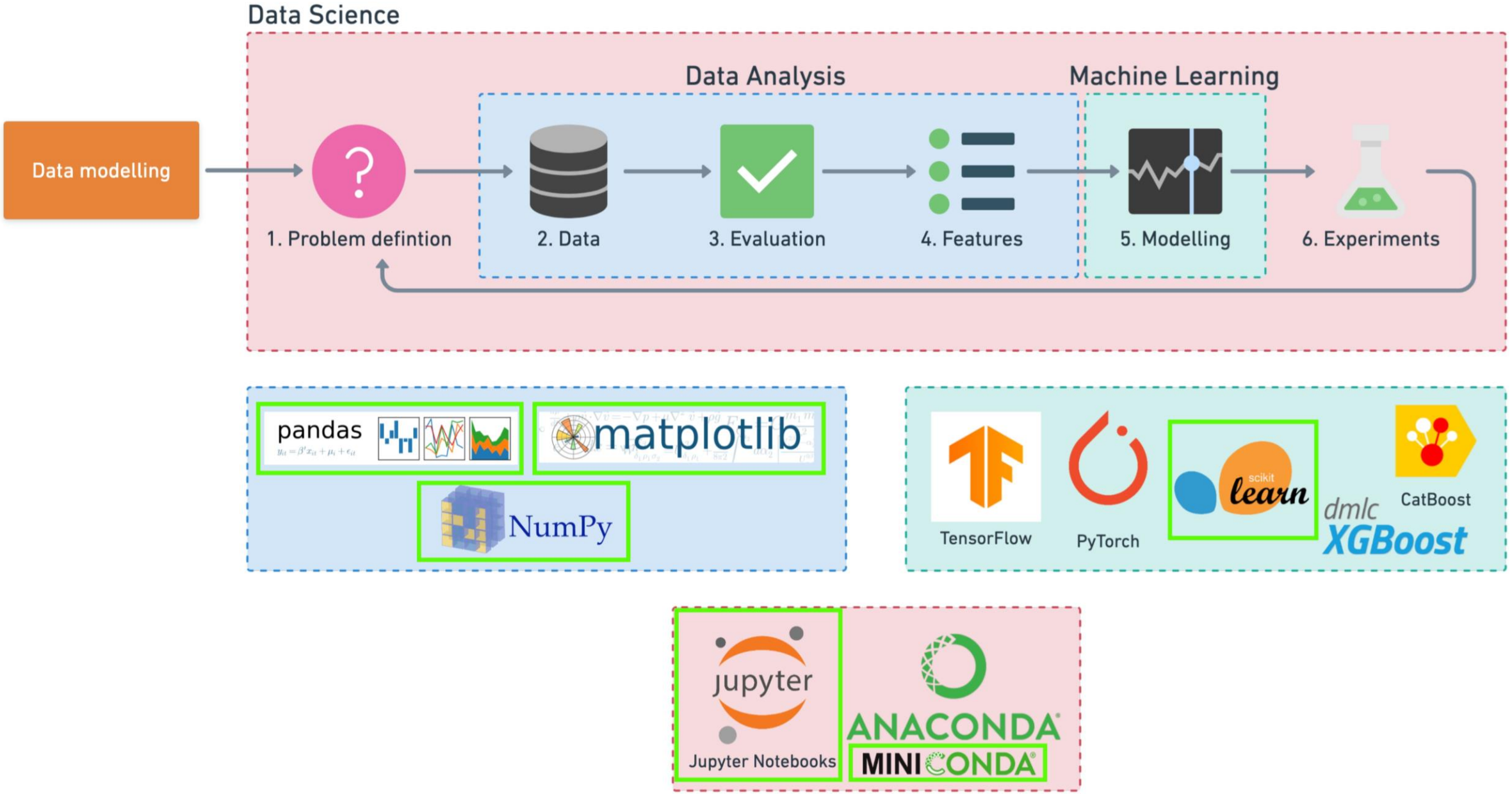
Structured Data Project 2: Predicting the sale price of Bulldozers (regression)



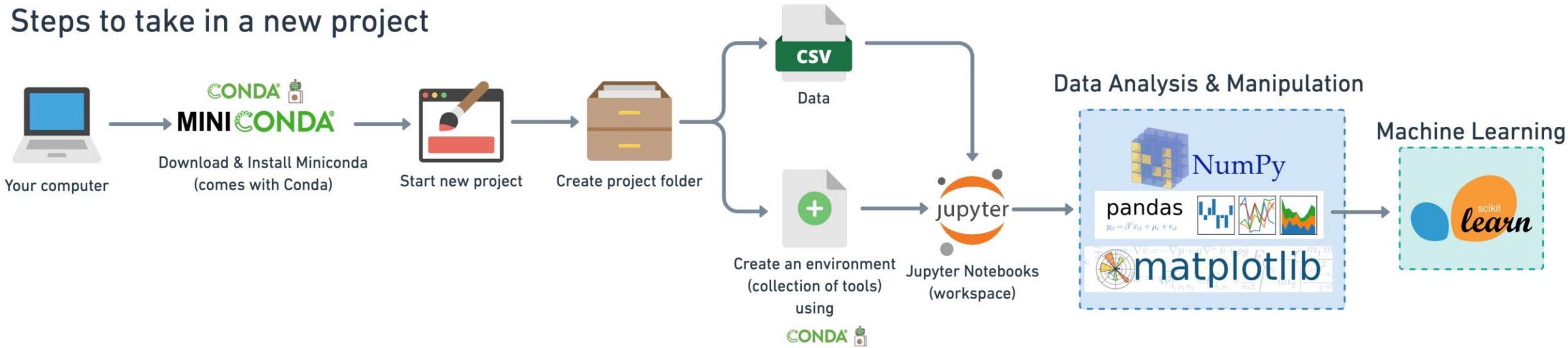
Steps in a full machine learning project



Tools you can use

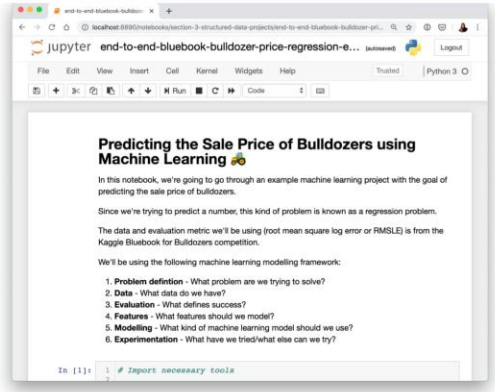


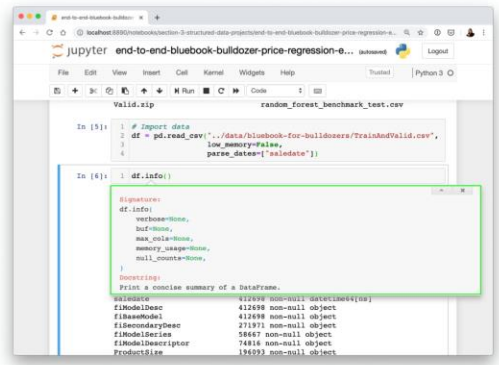
Steps to take in a new project

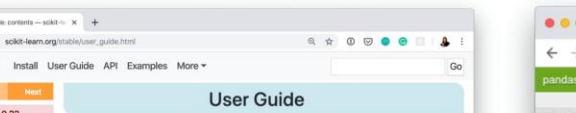


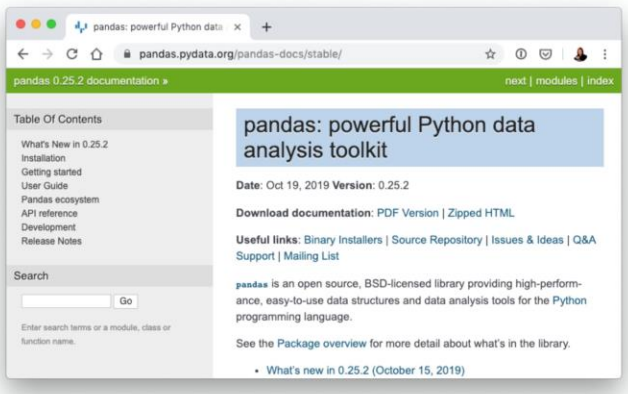
Where can you get help?

- Follow along with the code
- Try it for yourself
- Press SHIFT + TAB to read the docstring
- Search for it
- Try again
- Ask



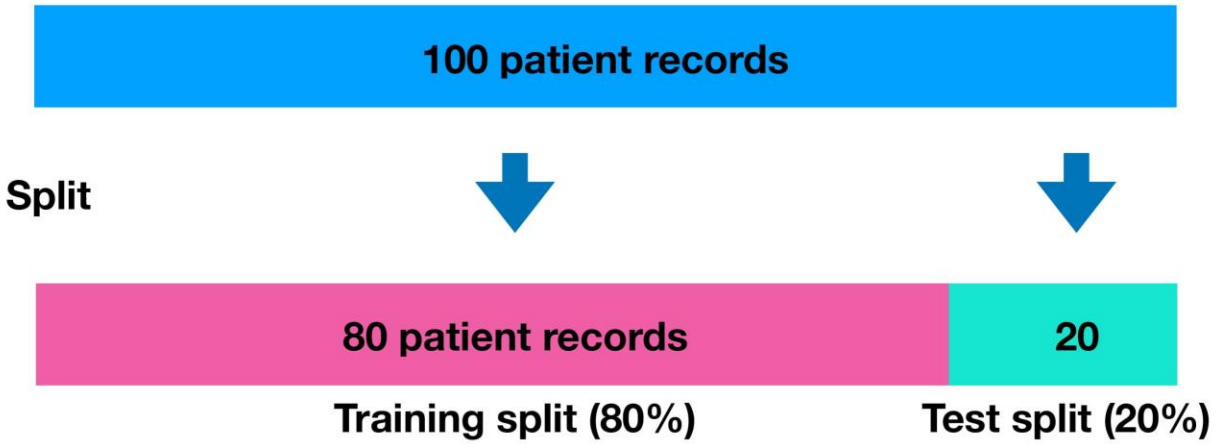






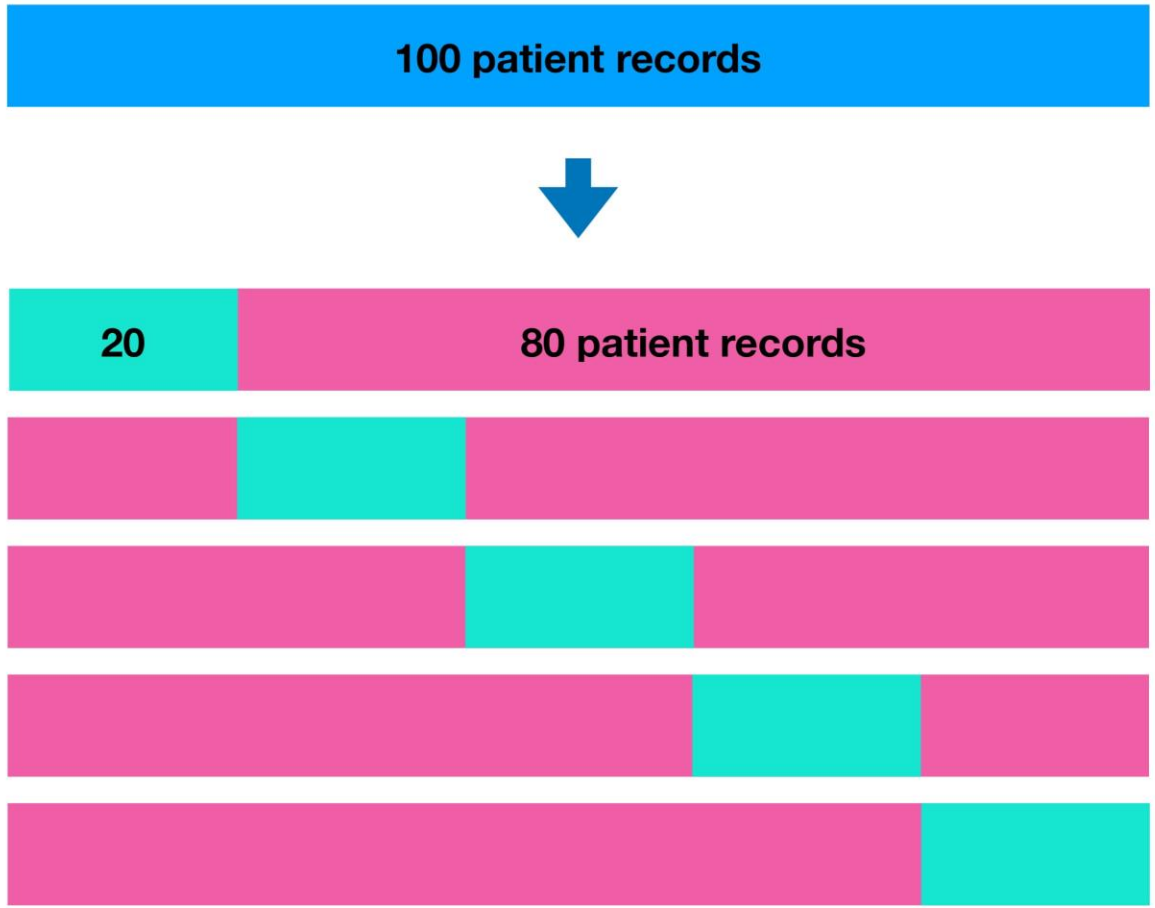
Cross-validation

Normal Train & Test Split



Model is trained on training data, and evaluated on the test data.

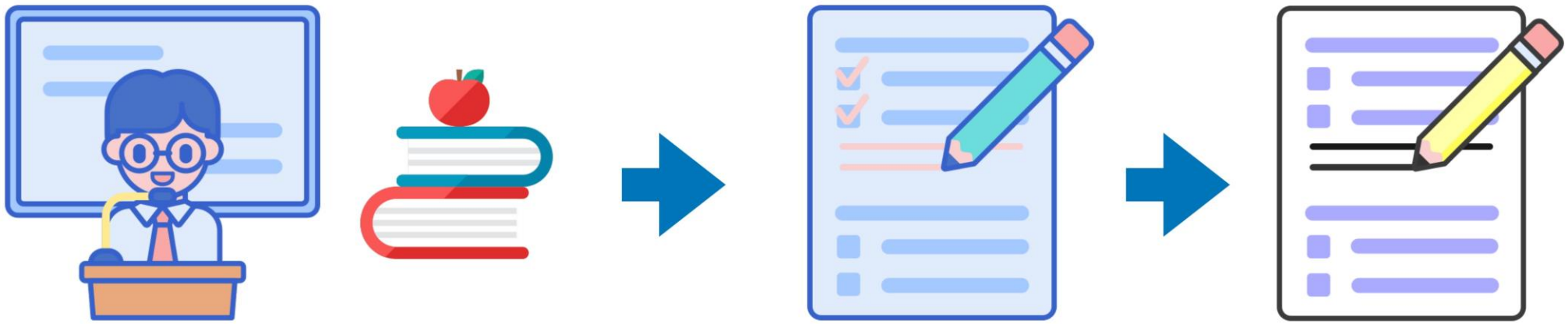
5-fold Cross-validation



Model is trained on 5 different versions of training data, and evaluated on 5 different versions of the test data.

The most important concept in machine learning

(the 3 sets)



Course materials (training set)

Practice exam (validation set)

Final exam (test set)

Generalization

The ability for a machine learning model to perform well on data it hasn't seen before.

Classification and Regression metrics

Classification	Regression
Accuracy	**R^2 (r-squared)**
Precision	Mean absolute error (MAE)
Recall	Mean squared error (MSE)
F1	Root mean squared error (RMSE)

Bold = default evaluation in Scikit-Learn

Which regression metric should you use?

- **R^2** is similar to accuracy. It gives you a quick indication of how well your model might be doing. Generally, the closer your **R^2** value is to 1.0, the better the model. But it doesn't really tell exactly how wrong your model is in terms of how far off each prediction is.
- **MAE** gives a better indication of how far off each of your model's predictions are on average.
- As for **MAE** or **MSE**, because of the way MSE is calculated, squaring the differences between predicted values and actual values, it amplifies larger differences. Let's say we're predicting the value of houses (which we are).
  - Pay more attention to MAE: When being $10,000 off is ***twice*** as bad as being $5,000 off.
  - Pay more attention to MSE: When being $10,000 off is ***more than twice*** as bad as being $5,000 off.