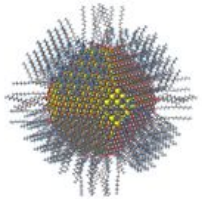
Hot Pots for Good Dots

Technology Review

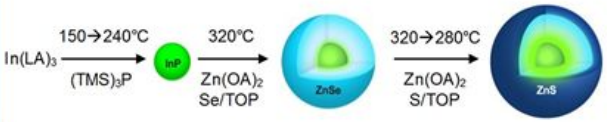
Benedicte Diakubama
Florence Dou
Hao Nguyen
Harrison Sarsito

Targeted properties of Quantum Dots via Machine Learning

- Quantum dots are used in optoelectronics, photovoltaics, catalysis...

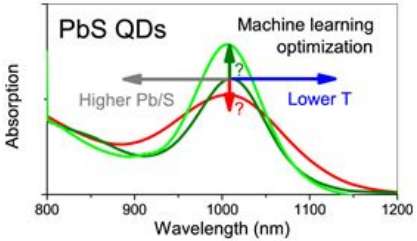


- Syntheses to achieve desired properties are typically ***trial and error*** based



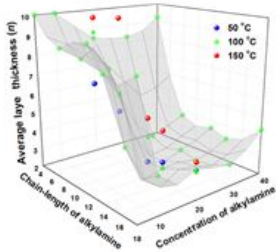
How do we get a good dataset when everyone reports their results differently?

- Mine old lab notebook data



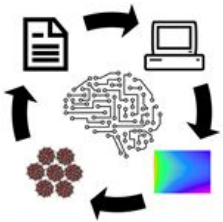
ACS Nano **2019**, *13*, *10*, 11122–11128

- Make 70+ samples



Chem. Mater. **2019**, *31*, 9, 3281–3292

- Use existing dataset from *Santos et al.*



- Manually extracted conditions and properties from literature
- Focus on CdSe, expand dataset
- Feature engineering
- Improve upon models in Python

J. Phys. Chem. C **2020**, *124*, 44, 24298–24305

Considered Machine Learning Libraries

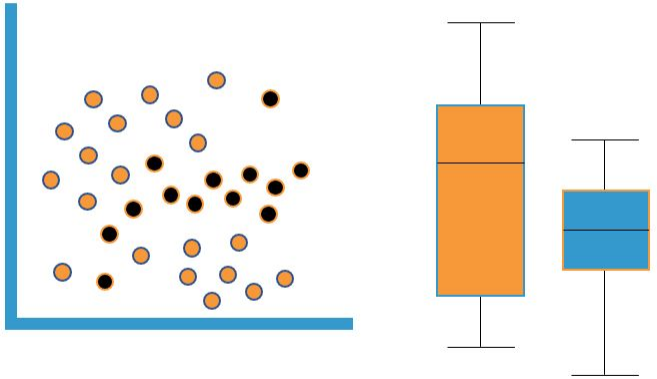
It's a simple and efficient tools for predictive data. It features classification, regression, and clustering algorithm. Eg: Random Forest

PYTORCH

It's like Numpy with strong GPU helps facilitates building deep learning projects. It's based on the torch library, and it uses graphics processing units.

TensorFlow

It's an open-source library for fast numerical computing use for deep learning. Eg: Linear Regression

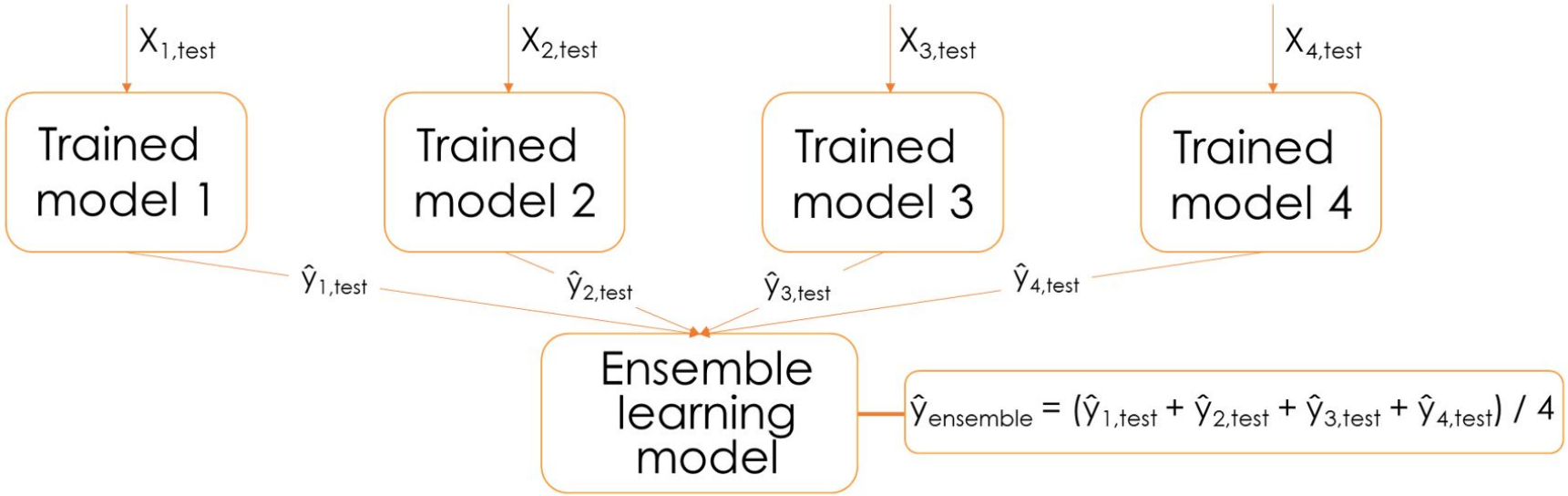


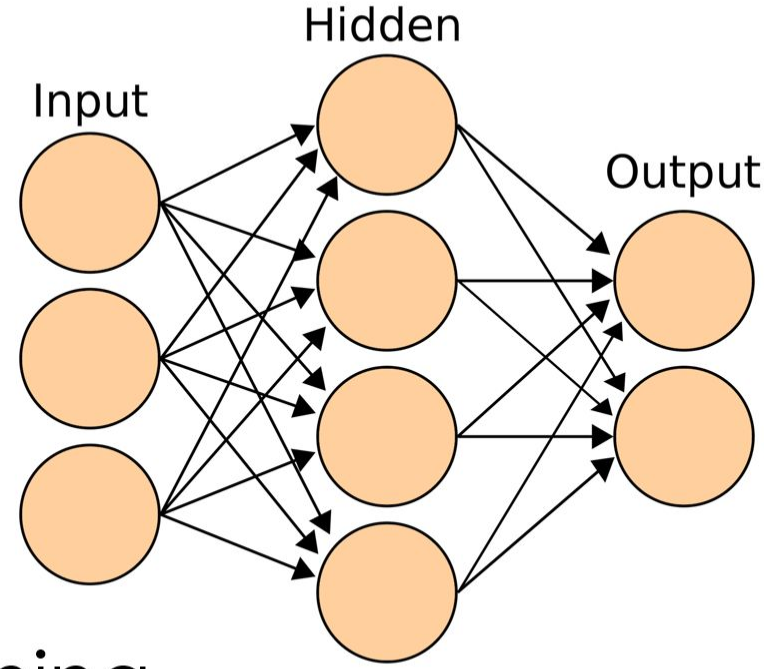
- It is free & easy to use
- Scikit-learn does what we need
- It is backed and updated by many authors

scikit Bagging

Bagging (Bootstrap Aggregation) as a Predictive Model

- **S** (**N** number of input conditions and outputs) →Bootstrap samples $\mathbf{Z_1}$, $\mathbf{Z_2}$, $\mathbf{Z_3}$ … $\mathbf{Z_M}$
- Each bootstrap sample's size is equal to **N** / **M**

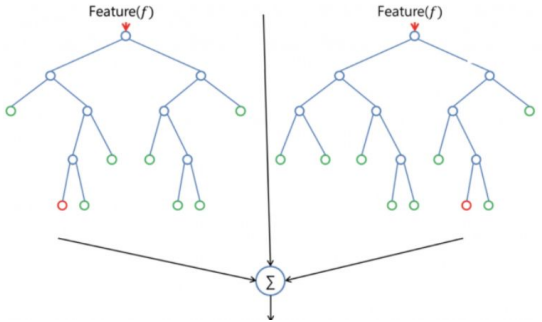




- Simplification
- Fixed parameters
- Bad for unsupervised learning and deep learning

Random Forest

- What is it?:
  - Builds multiple decision trees and merges them together to get a more accurate and stable prediction. It overcomes the overfitting challenge decision tree faces

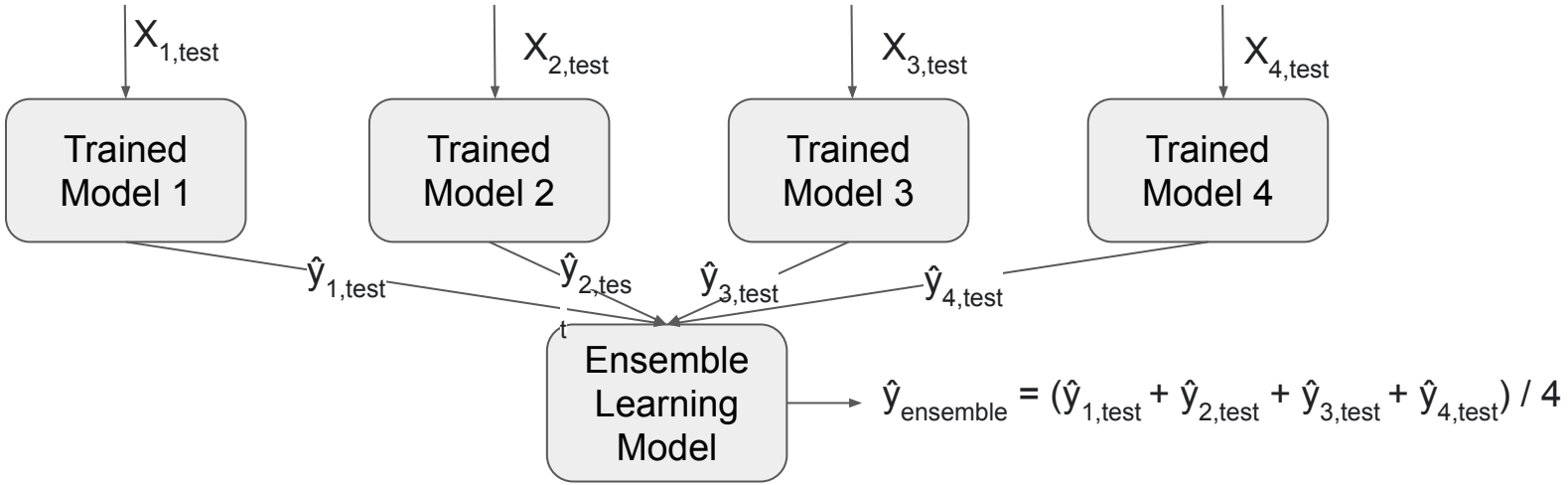


- How it works?:
  - Data set will be divided into training and testing set
  - The training set is trained on the model using `classifier.fit()` then import `RandomForestClassifier`
  - `classifier.predict()` is used to predict test set

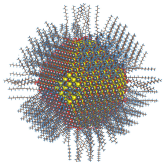
Bagging (Bootstrap Aggregation) as a Predictive Model

S (**N** number of input conditions and outputs) →Bootstrap samples $\mathbf{Z_1}$, $\mathbf{Z_2}$, $\mathbf{Z_3}$ … $\mathbf{Z_M}$

Each bootstrap sample's size is equal to **N** / **M**



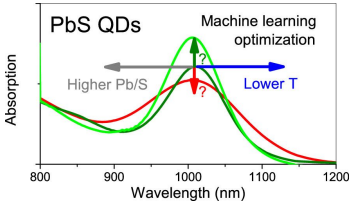
Targeted properties of Quantum Dots via Machine Learning



Quantum dots are used in optoelectronics, photovoltaics, catalysis...

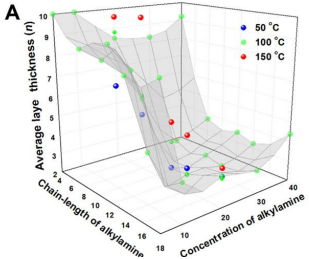
How do we get a good dataset when everyone reports their results differently?

1. Mine old lab notebook data



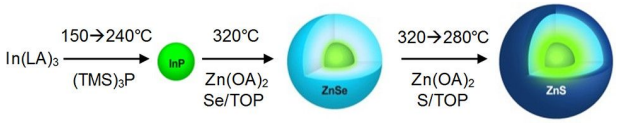
ACS Nano 2019, 13, 10, 11122–11128

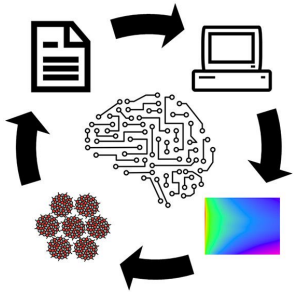
2. Make 70+ samples



Chem. Mater. 2019, 31, 9, 3281–3292

Syntheses to achieve desired properties are typically trial and error based





3. Use existing dataset from Santos *et al*

- Manually extracted conditions and properties from literature
- Focus on CdSe, expand dataset
- Feature engineering
- Improve upon models in Python

J. Phys. Chem. C 2020, 124, 44, 24298–24305

Bagging (Bootstrap Aggregation) as a Predictive Model

Bagging is a type of **Ensemble Method** in Machine Learning.

Ensembling is the usage of **multiple learning algorithms** to obtain **better predictive performance** than could be obtained from any of its constituent learning algorithms alone[1].

Join **several "weak learners"** to provide a **"strong learning"** collaborative result.

1. Opitz, D.; Maclin, R. (1999). "Popular ensemble methods: An empirical study". *Journal of Artificial Intelligence Research*. **11**: 169–198. doi:10.1613/jair.614.