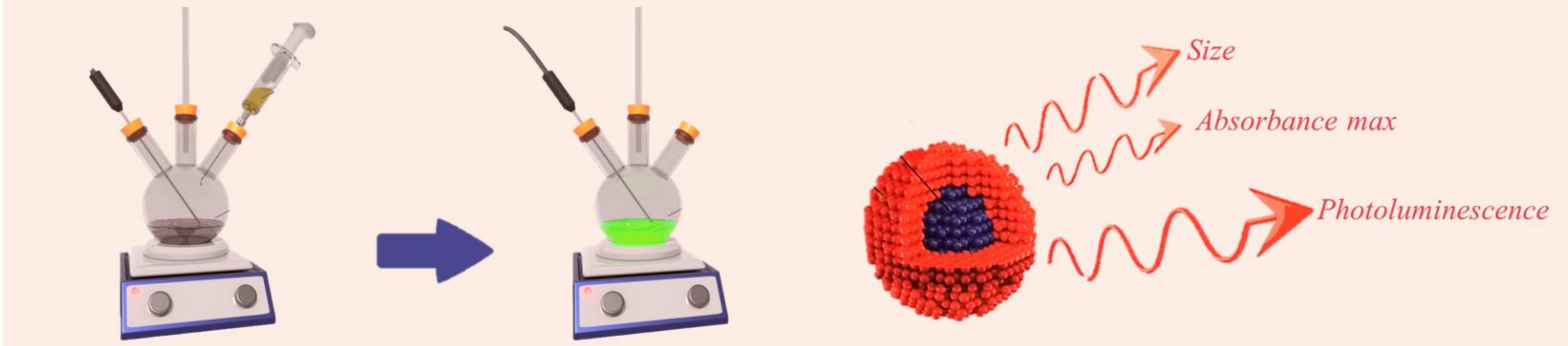
Hot Pots for Good Dots

Using machine learning to predict physical properties of CdSe quantum dots



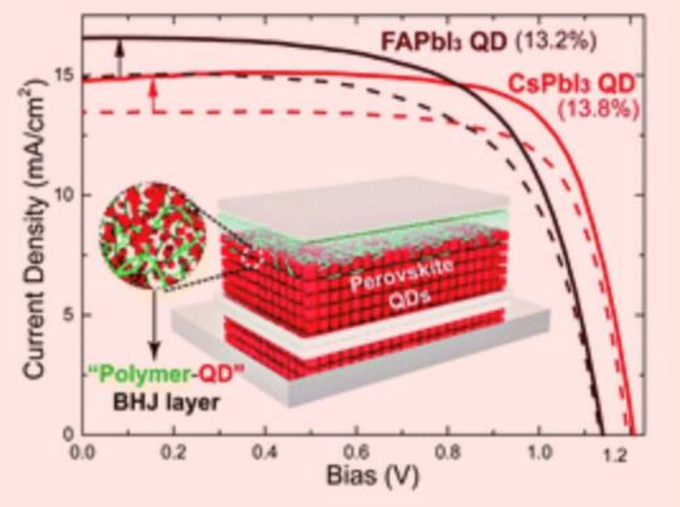
By: Benedicte **Diakubama**, Florence **Dou**, Hao **Nguyen**, Harrison **Sarsito**

Narrative:

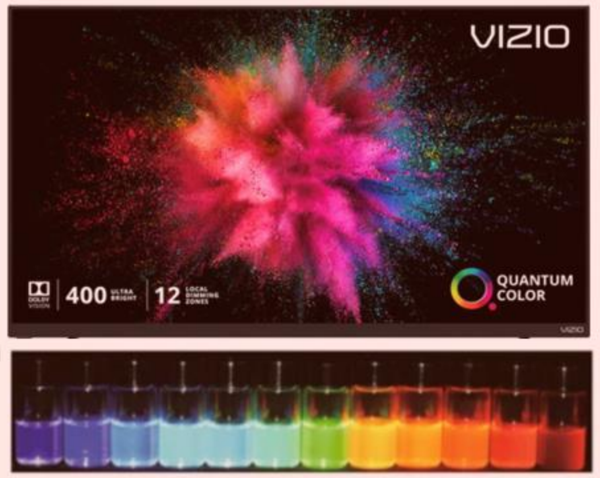
Hello everyone, what's cookin'?

My name is Harrison, and I'm joined here today with my teammates Benedicte, Florence and Hao, and we're really excited to share with you all what we've learned in the past weeks at the intersection of machine learning and quantum dots.

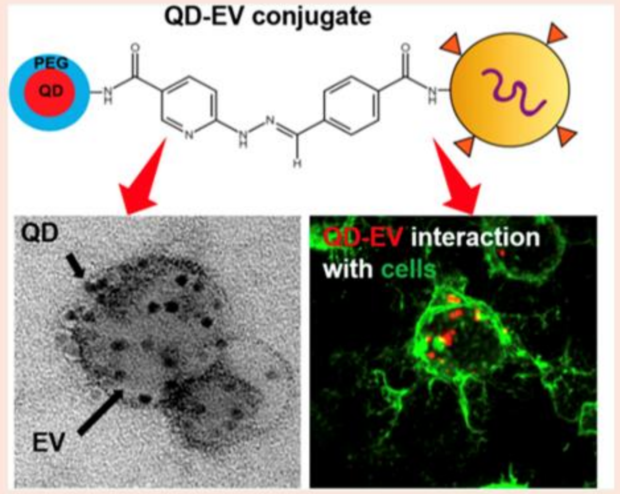
Synthesis of quantum dots is highly empirical



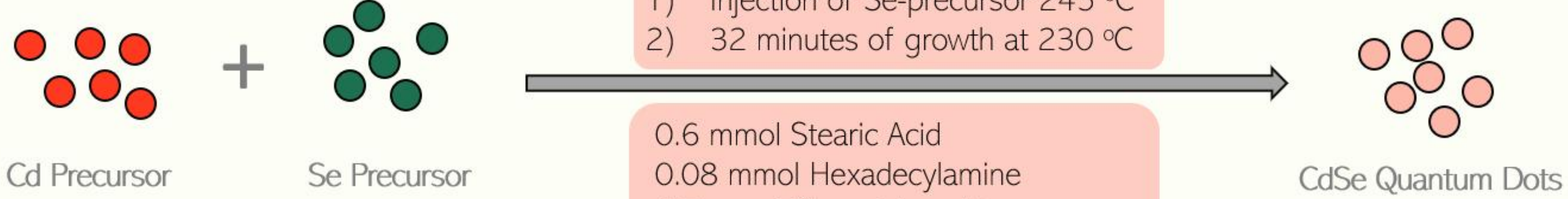
Ji, et al. *J. Mater. Chem. A*, **2020**, *8*, 8104-8112



Liu, et al. *Light Sci Appl*, **2020**, *9*, 83



Zhang, et al. *ACS Appl. Nano Mater.* **2020**, *3*, 7211-7222



1) Injection of Se-precursor 245 ºC
2) 32 minutes of growth at 230 ºC

0.6 mmol Stearic Acid
0.08 mmol Hexadecylamine
4.5 mmol Trioctylphosphine
7.89 mmol Octadecene

Čapek, et al. *Chem. Mater.* **2009**, 21, 8, 1743–1749

Goal: Develop a machine learning model that would predict how different **experimental conditions** would affect the **optoelectronic properties** of CdSe Quantum Dots synthesized via hot injection

Narrative:

In the recent decade, nanoscale semiconductors have received much attention in the scientific community due to their ability to absorb lights at different wavelengths, which make them suited for a variety of applications that include

Increasing the photoconversion efficiency of solar cells . Improving the pictorial "sharpness" of electronic displays such as TVs and computer monitors.
And even Labelling of extracellular vesicles for better probing of their behavior

One of the primary reasons why these materials are of interest for these applications is because they have size dependent physical properties, hence why a primary focus area in this research is to investigate the synthetic pathways to create nanomaterials of specific size and morphology.

The most common synthesis method for these semiconductors is the hot injection method, which in many aspects is very similar to delicious hot pots that we know and love.

Cadmium selenide quantum dots for example are a result of a combination of numerous ingredients put together at the right temperature and "cooked" for a predetermined amount of time.

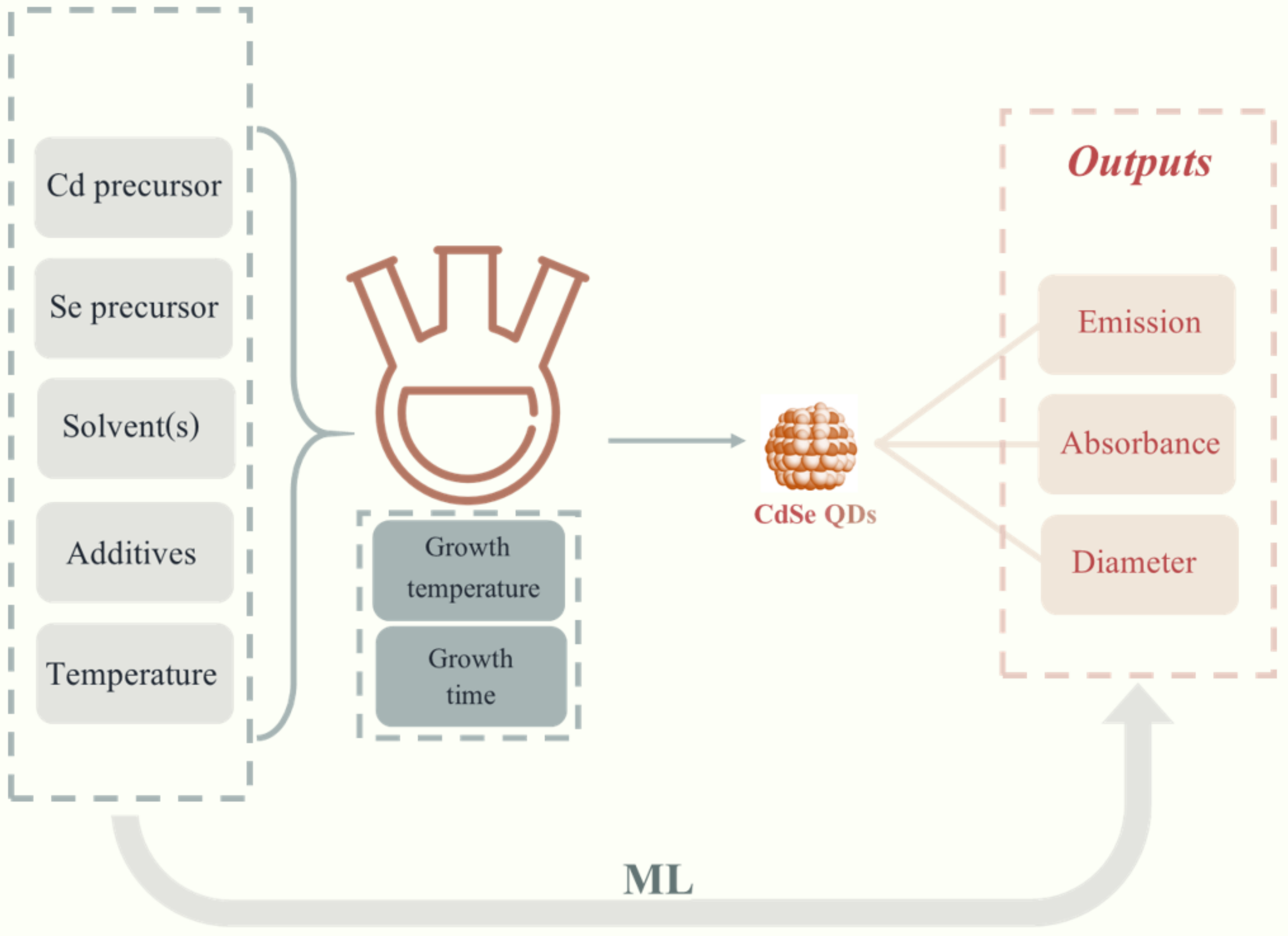
What makes things slightly more complicated is that there are a myriad of experimental conditions that could be used to synthesize the same material of the same size.

What our team decided to do for this project was to do an initial case study on the extensively studied CdSe quantum dot, to see if it's possible to develop…

Use Case 1: Property Prediction

Clay

Enthusiastic Undergrad Michael



Narrative:

Our first use case involves Clay– a 3rd year chemistry PhD student who's studying synthesis of CdSe nanoparticles.

Another thing to know about Clay is that he's frustrated, and here's why:

He is interested in the effects that different solvents and the growth temperature has on the final properties of these dots synthesized through hot injection.

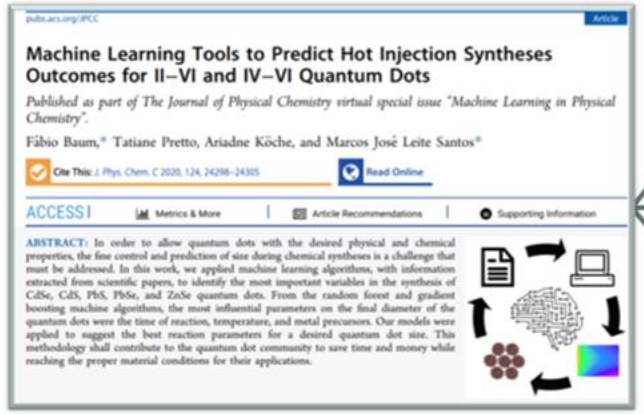
Unfortunately, he and his trusty undergrad sidekick can only go into lab twice a week due to COVID precautions, and his curiosity is killing him.

As my teammate Florence will show you later, Clay could alleviate this curiosity by making use of our predictive model, which is integrated into an online interface that would allow him to enter in quantities of each of these experimental inputs including his inputs of interest the growth temperature and solvent identity, and see its effect on the resulting CdSe quantum dots' diameter, max absorbance and emission.

Use Case 2: ML Model Comparison to *Santos et al*

Zach

Zach's grad students



pubs.acs.org/JPCC

Article

Machine Learning Tools to Predict Hot Injection Syntheses Outcomes for II–VI and IV–VI Quantum Dots

Published as part of The Journal of Physical Chemistry virtual special issue "Machine Learning in Physical Chemistry".

Fábio Baum,* Tatiane Pretto, Ariadne Köche, and Marcos José Leite Santos*

Cite This: J. Phys. Chem. C 2020, 124, 24298–24305

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

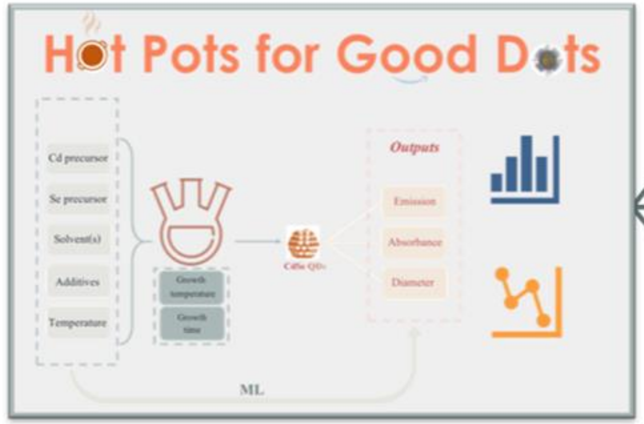
ABSTRACT: In order to allow quantum dots with the desired physical and chemical properties, the fine control and prediction of size during chemical syntheses is a challenge that must be addressed. In this work, we applied machine learning algorithms, with information extracted from scientific papers, to identify the most important variables in the synthesis of CdSe, CdS, PbS, PbSe, and ZnSe quantum dots. From the random forest and gradient boosting machine algorithms, the most influential parameters on the final diameter of the quantum dots were the time of reaction, temperature, and metal precursors. Our models were applied to suggest the best reaction parameters for a desired quantum dot size. This methodology shall contribute to the quantum dot community to save time and money while reaching the proper material conditions for their applications.

Written in R

Tested with 5 models

Predicting one output

Vs.



Written in Python

Tested with 12 models

Predicting three outputs

Narrative:

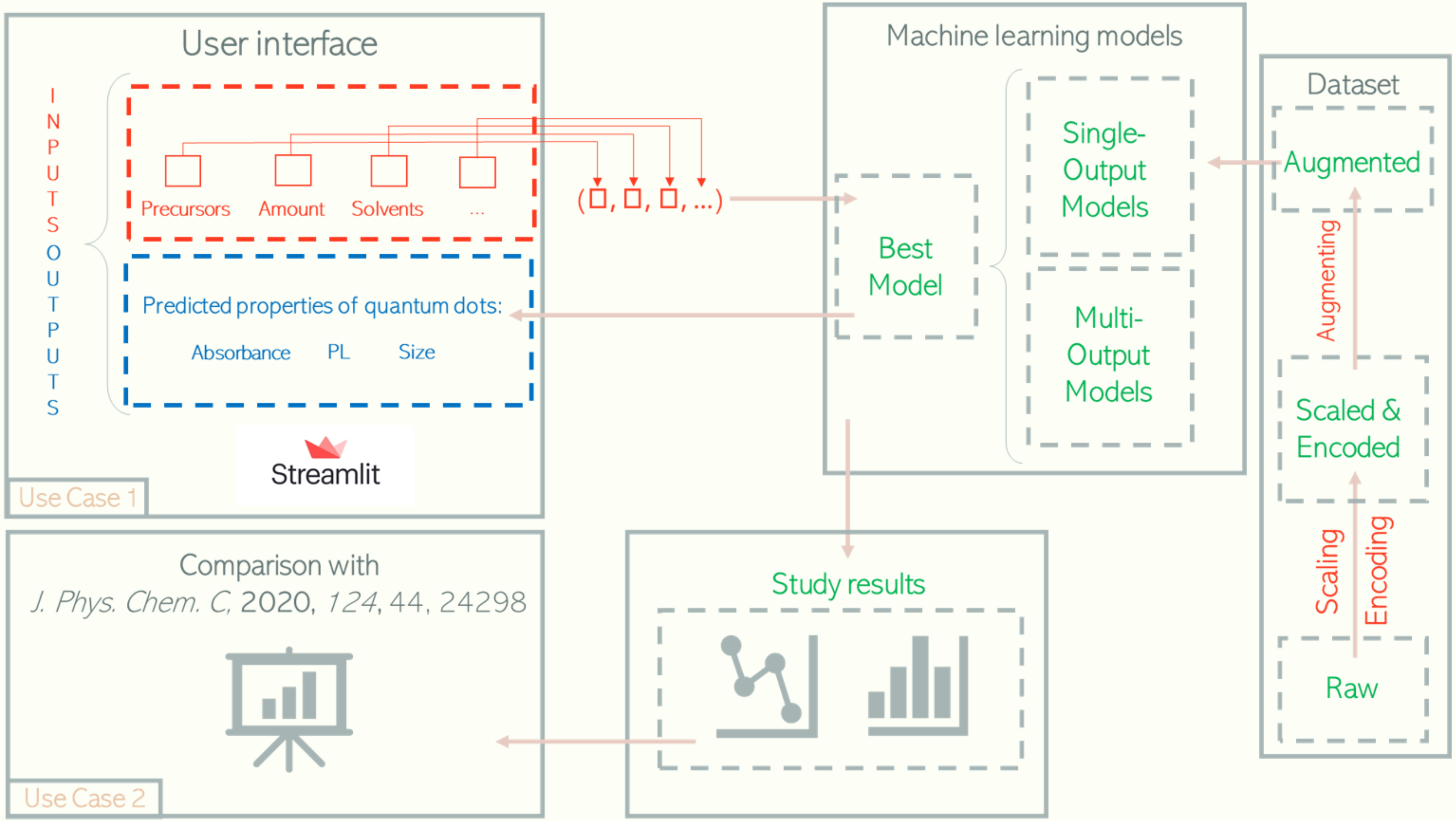
The second use case involves Zach, a notoriously demanding chemistry professor who is interested in synthesizing CdSe quantum dots with very specific properties.

He has read this ML paper published in the JPCC previously about how machine learning can be used to predict diameters of synthesized CdSe dots, but wonders if he should get a second opinion before he tasks his rather posh group of 8 PhD students to grind through some synthesis.

In particular, he is curious if there are other ML models out there that can predict not only the diameter of the quantum dots, but also their max absorbance and photoluminescence at the same time, and if so, how their predictive capabilities compare.

Now I'll pass it on to Hao, who will outline our project workflow.

Workflow Diagram



Narrative:

This is the workflow of our project and also the outline of our presentation
We started with the raw dataset that we obtained from the JPCC paper, then we cleaned up and manipulated the dataset, so it is usable for our goals.

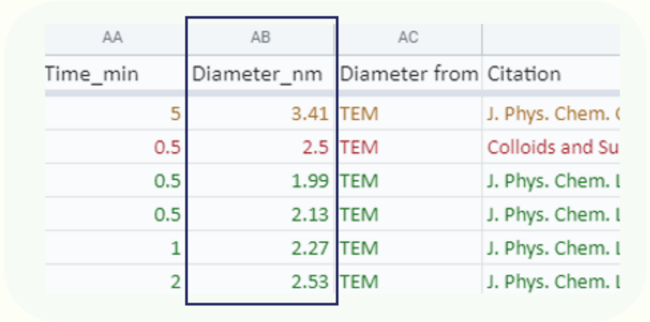
Then we used the dataset to test on different ML models including single and multi outputs.

We analyzed the performance of those models and compare with the JPCC paper which is our use case number 2.

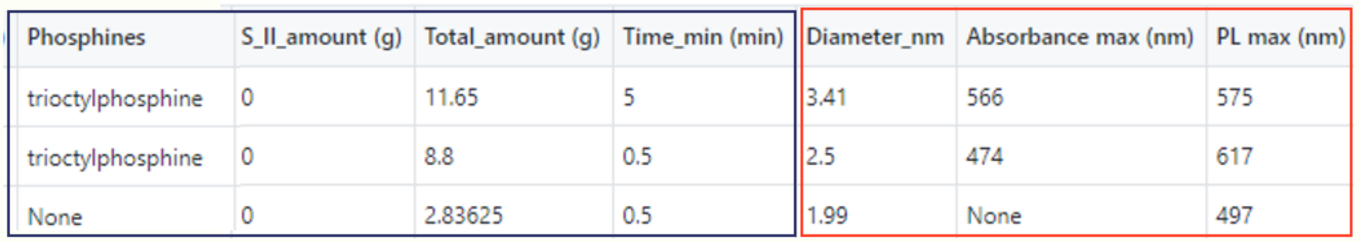
For the model with the best performance, we apply it to our user interface Streamlit as our use case number 1.

And now, my teammate Florence with explain the data manipulation part.

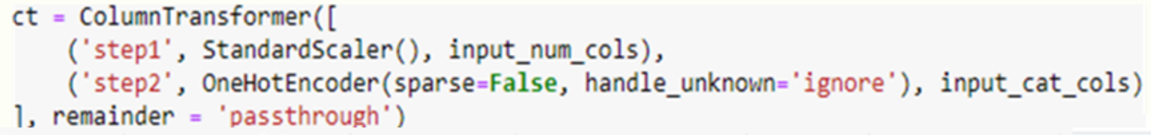
Dataset Manipulation



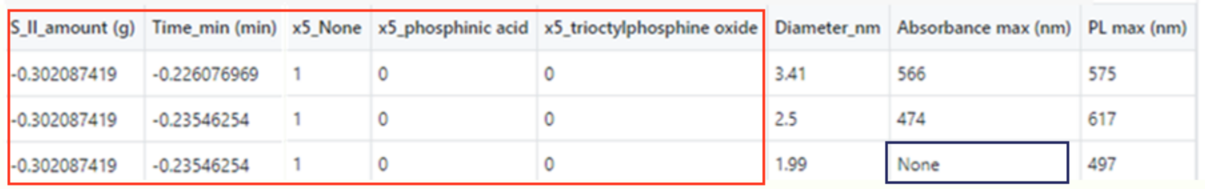
AA	AB	AC	
Time_min	Diameter_nm	Diameter from	Citation
5	3.41	TEM	J. Phys. Chem. (
0.5	2.5	TEM	Colloids and Su
0.5	1.99	TEM	J. Phys. Chem. l
0.5	2.13	TEM	J. Phys. Chem. l
1	2.27	TEM	J. Phys. Chem. l
2	2.53	TEM	J. Phys. Chem. l



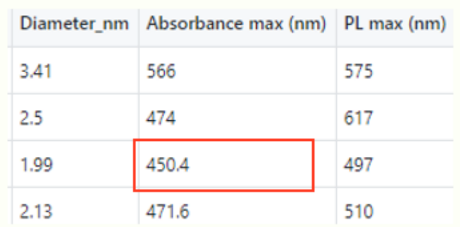
Phosphines	S_II_amount (g)	Total_amount (g)	Time_min (min)	Diameter_nm	Absorbance max (nm)	PL max (nm)
trioctylphosphine	0	11.65	5	3.41	566	575
trioctylphosphine	0	8.8	0.5	2.5	474	617
None	0	2.83625	0.5	1.99	None	497



```
ct = ColumnTransformer([
    ('step1', StandardScaler(), input_num_cols),
    ('step2', OneHotEncoder(sparse=False, handle_unknown='ignore'), input_cat_cols)
], remainder = 'passthrough')
```



S_II_amount (g)	Time_min (min)	x5_None	x5_phosphinic acid	x5_trioctylphosphine oxide	Diameter_nm	Absorbance max (nm)	PL max (nm)
-0.302087419	-0.226076969	1	0	0	3.41	566	575
-0.302087419	-0.23546254	1	0	0	2.5	474	617
-0.302087419	-0.23546254	1	0	0	1.99	None	497



Diameter_nm	Absorbance max (nm)	PL max (nm)
3.41	566	575
2.5	474	617
1.99	450.4	497
2.13	471.6	510

Data Cleaning

Double-checked values from the literatures

Data Expansion

Added absorbance and emission outputs

df.shape() – (234, 33)

Scaling & Encoding

sklearn StandardScaler() scales numerical columns

OneHotEncoder() transforms categorical entries to new columns with 1 or 0 values

Data Augmentation

Used **Random Forest** to predict None values in absorbance and emission outputs for a larger dataset

Narrative:

We spent a lot of time cleaning up the data. We actually went to each citation to make sure all the numbers are correct and fix typos.
We expand the data from only having diameter as an output to having three outputs which are diameter, absorbance max, and photoluminescence.
Because of that, we reduce the dataset length from about eight hundreds to 233 datarows.

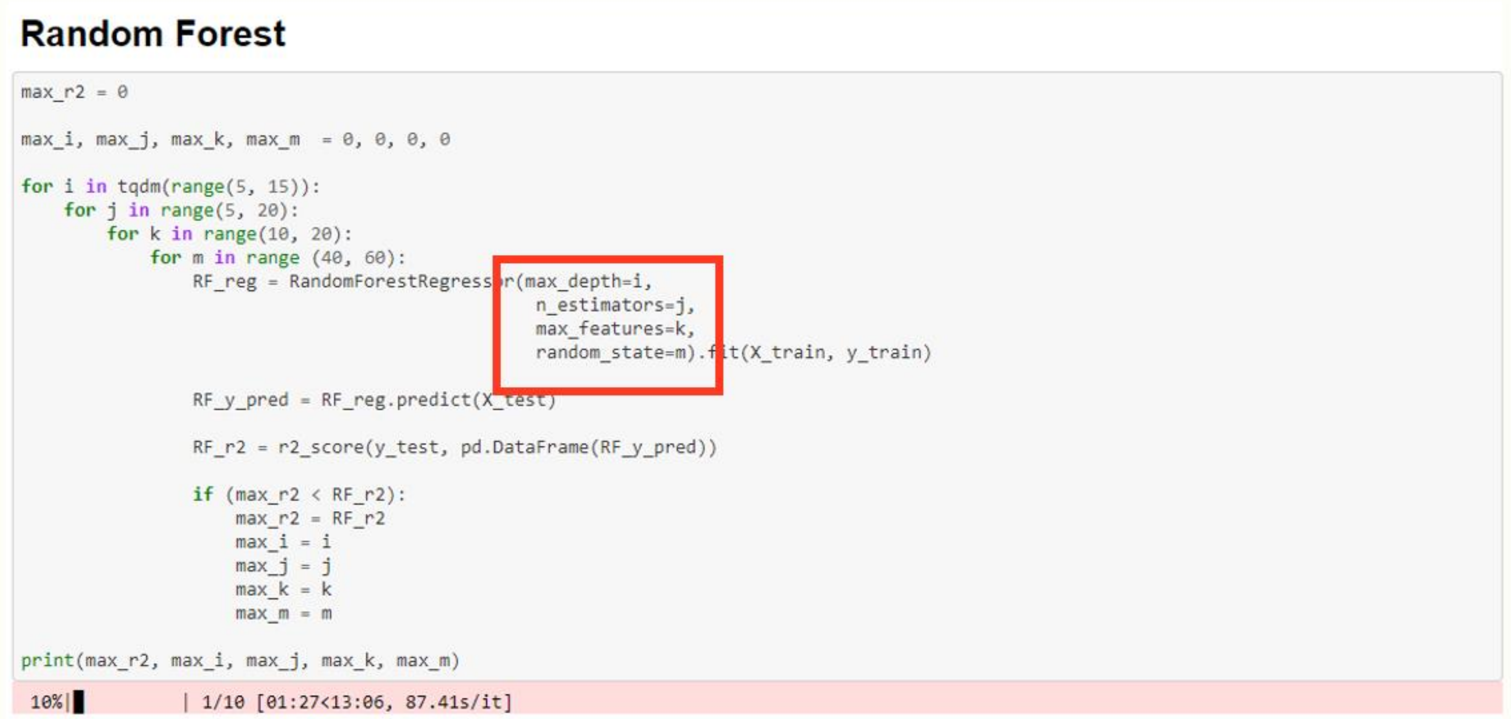
We then scaled and encoded the dataset, meaning that we normalize the numerical entries and turned the categorical entries to new columns with 1 or 0 values.

Finally, to deal with rows that were missing some outputs, we augmented the dataset meaning that we used Random Forest to predict the missing outputs, then added those predicted values to our dataset.

ML Models: Grid search to optimize parameters

Models:

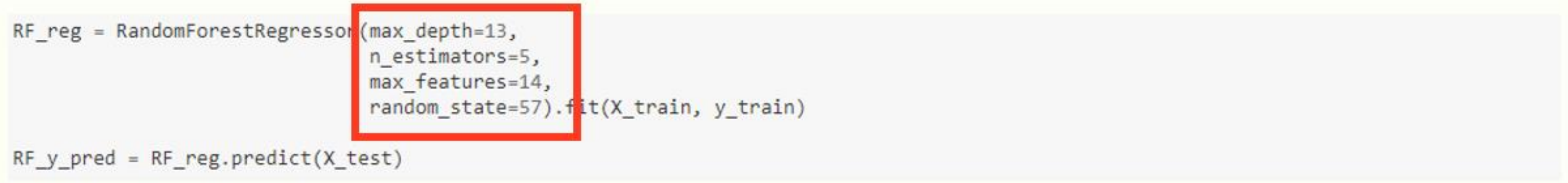
Multilinear Regression
Gradient Boosting
SVR (linear)
SVR (rbf)
Extra Trees
Lasso
Ada Boost
Ridge CV
Ridge
Decision Tree
KNN
Random Forest



Random Forest

```
max_r2 = 0

max_i, max_j, max_k, max_m  = 0, 0, 0, 0

for i in tqdm(range(5, 15)):
    for j in range(5, 20):
        for k in range(10, 20):
            for m in range (40, 60):
                RF_reg = RandomForestRegressor(max_depth=i,
                                                n_estimators=j,
                                                max_features=k,
                                                random_state=m).fit(X_train, y_train)

                RF_y_pred = RF_reg.predict(X_test)

                RF_r2 = r2_score(y_test, pd.DataFrame(RF_y_pred))

                if (max_r2 < RF_r2):
                    max_r2 = RF_r2
                    max_i = i
                    max_j = j
                    max_k = k
                    max_m = m

print(max_r2, max_i, max_j, max_k, max_m)
```

```
 10%|█          | 1/10 [01:27<13:06, 87.41s/it]
```



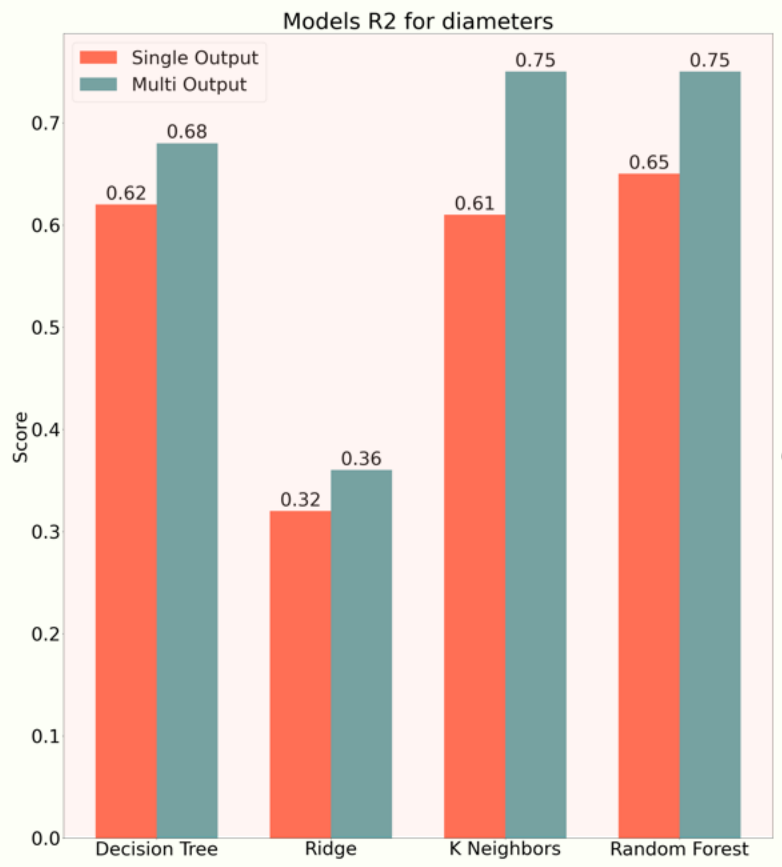
```
RF_reg = RandomForestRegressor(max_depth=13,
                               n_estimators=5,
                               max_features=14,
                               random_state=57).fit(X_train, y_train)

RF_y_pred = RF_reg.predict(X_test)
```

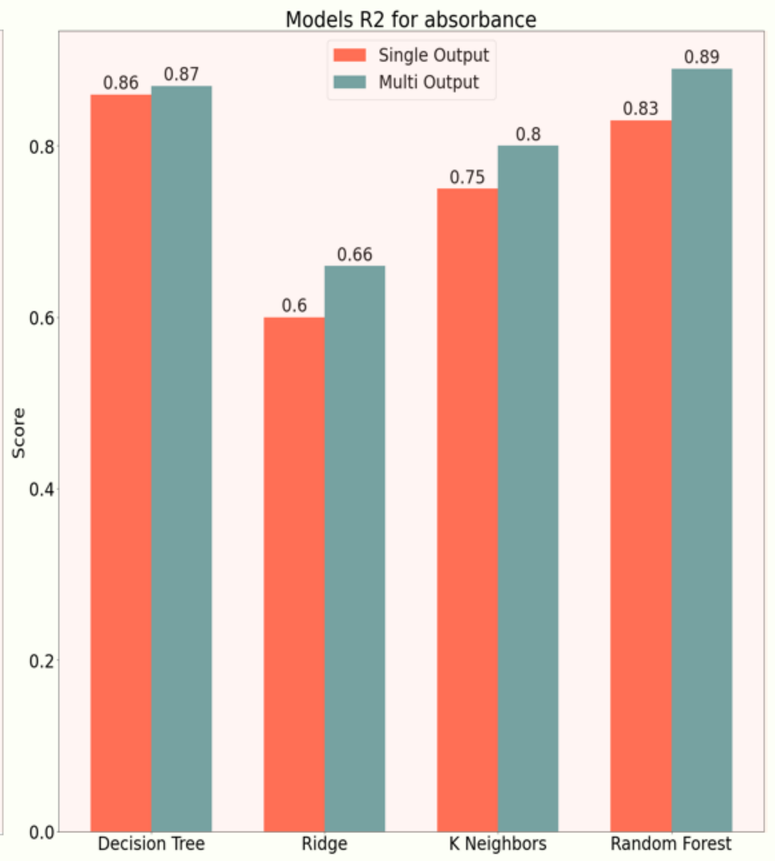
Narrative:

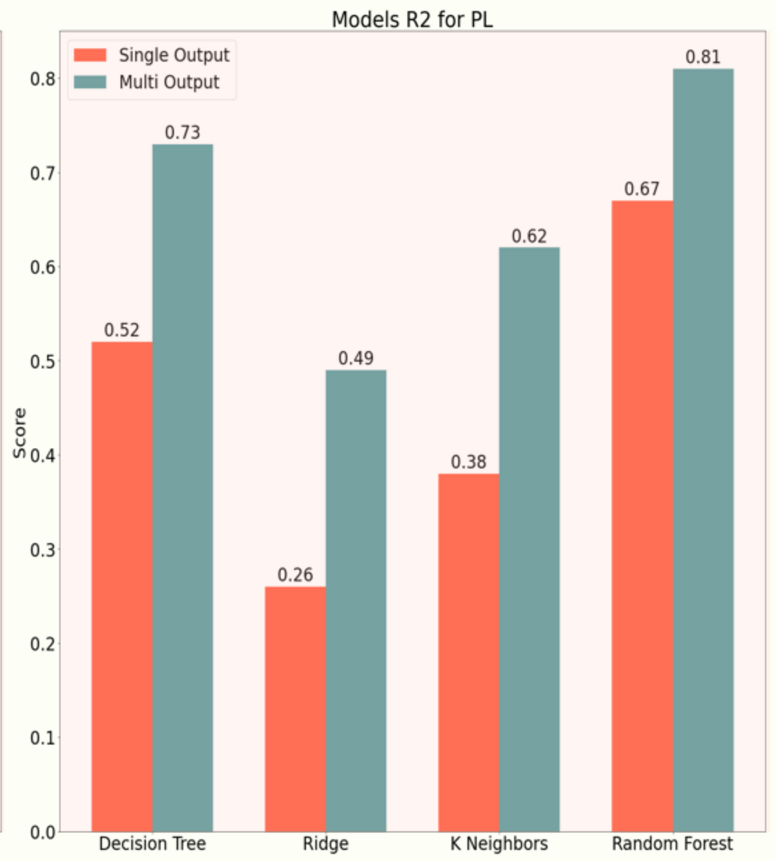
We tested our data on 12 different models with both single out put and multi output.

For each model, we optimized the parameters by grid searching to see which parameters combination gives the best performance. The results were then applied to the final model.

Absorbance predicts best, multi- outperforms single output







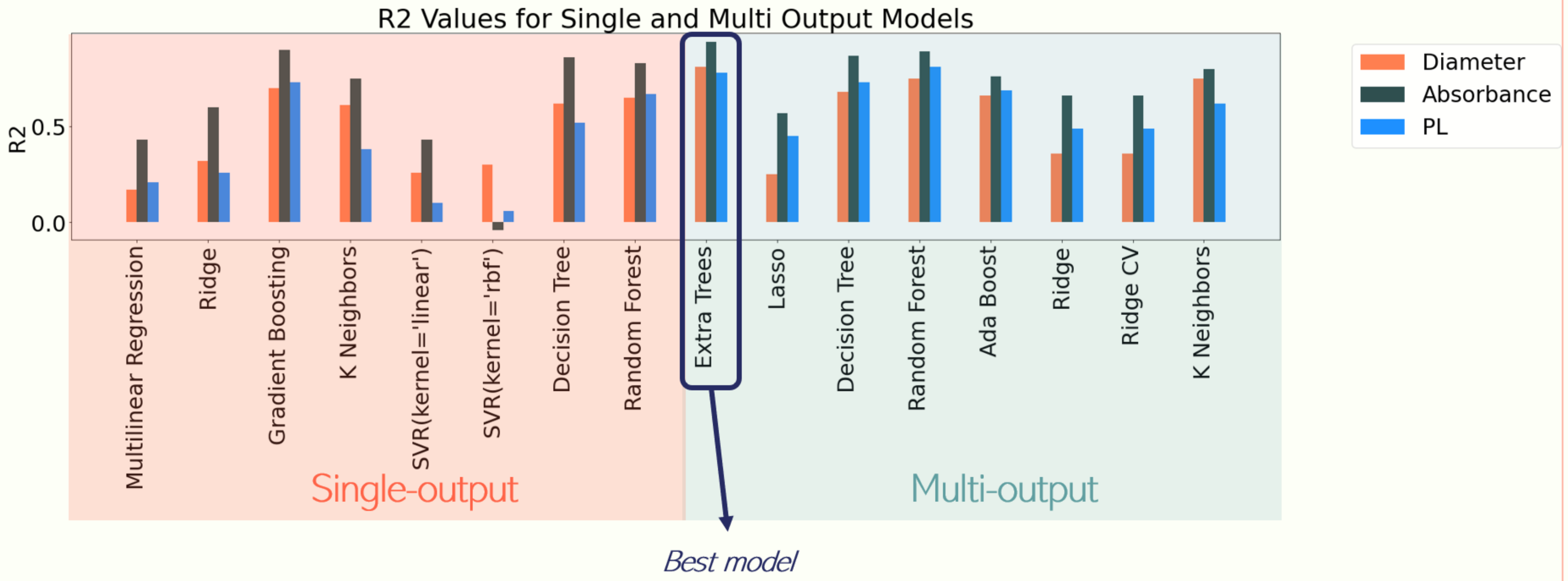
Narrative:

Based on the output, we saw that absorbance prediction was best compared to the other 2

This was seen by the R2 metric which was closed to 1 for absorbance this due to the fact that even the paper back calculated the diameter based on absorbance: diameter and absorbance are closely related together

Multi output was proven to improve the model because output are related to each other

Overall best model was chosen

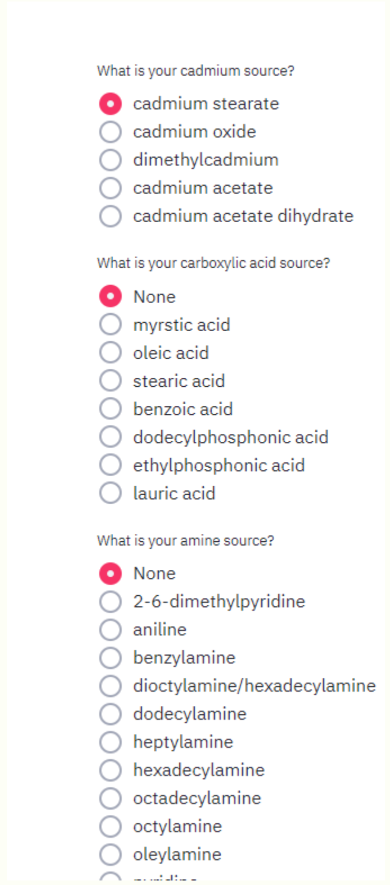


Extra Trees R^2: 0.94 (absorbance), 0.81 (diameter), 0.78 (PL)

Narrative:

Extra tree was seen to be the best model to predict with an R2 for absorbance of 0.94

Streamlit – User interface



What is your cadmium source?

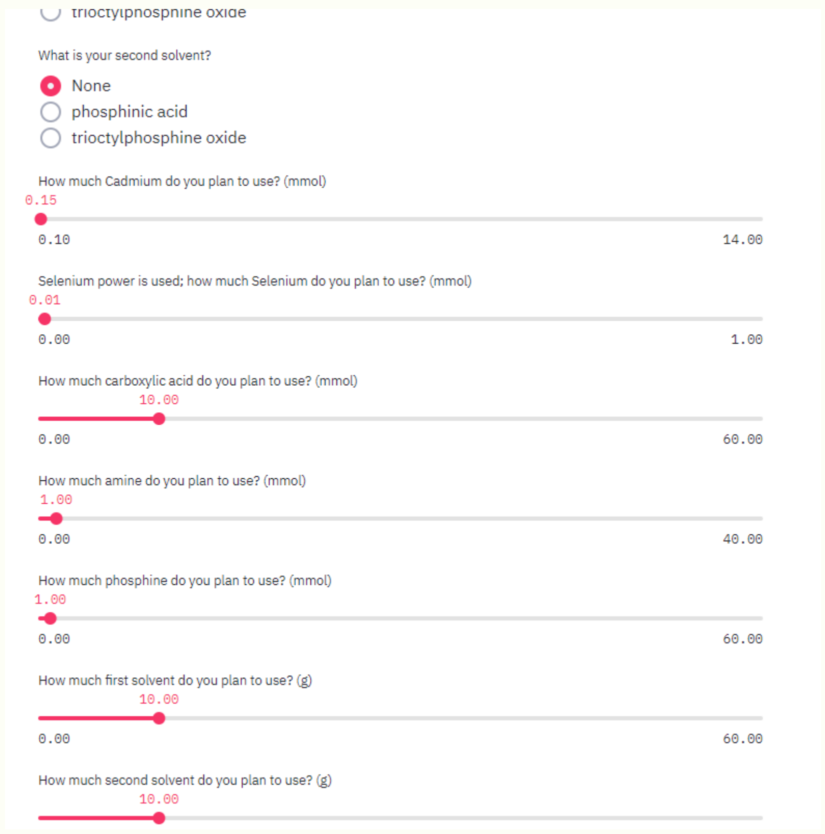
- [✓ cadmium stearate]
- cadmium oxide
- dimethylcadmium
- cadmium acetate
- cadmium acetate dihydrate

What is your carboxylic acid source?

- [✓ None]
- myrstic acid
- oleic acid
- stearic acid
- benzoic acid
- dodecylphosphonic acid
- ethylphosphonic acid
- lauric acid

What is your amine source?

- [✓ None]
- 2-6-dimethylpyridine
- aniline
- benzylamine
- dioctylamine/hexadecylamine
- dodecylamine
- heptylamine
- hexadecylamine
- octadecylamine
- octylamine
- oleylamine



- trioctylphosphine oxide

What is your second solvent?

- [✓ None]
- phosphinic acid
- trioctylphosphine oxide

How much Cadmium do you plan to use? (mmol)
0.15
0.10 — 14.00

Selenium power is used; how much Selenium do you plan to use? (mmol)
0.01
0.00 — 1.00

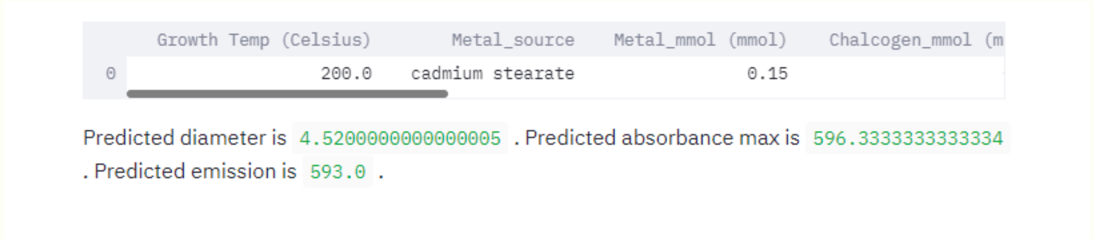
How much carboxylic acid do you plan to use? (mmol)
10.00
0.00 — 60.00

How much amine do you plan to use? (mmol)
1.00
0.00 — 40.00

How much phosphine do you plan to use? (mmol)
1.00
0.00 — 60.00

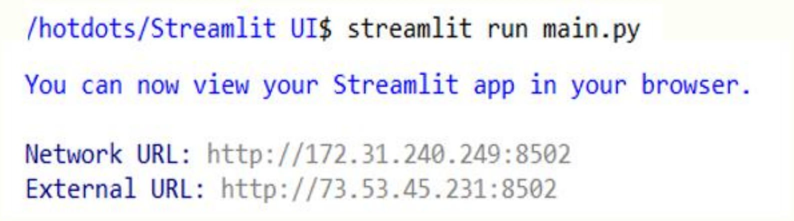
How much first solvent do you plan to use? (g)
10.00
0.00 — 60.00

How much second solvent do you plan to use? (g)
10.00

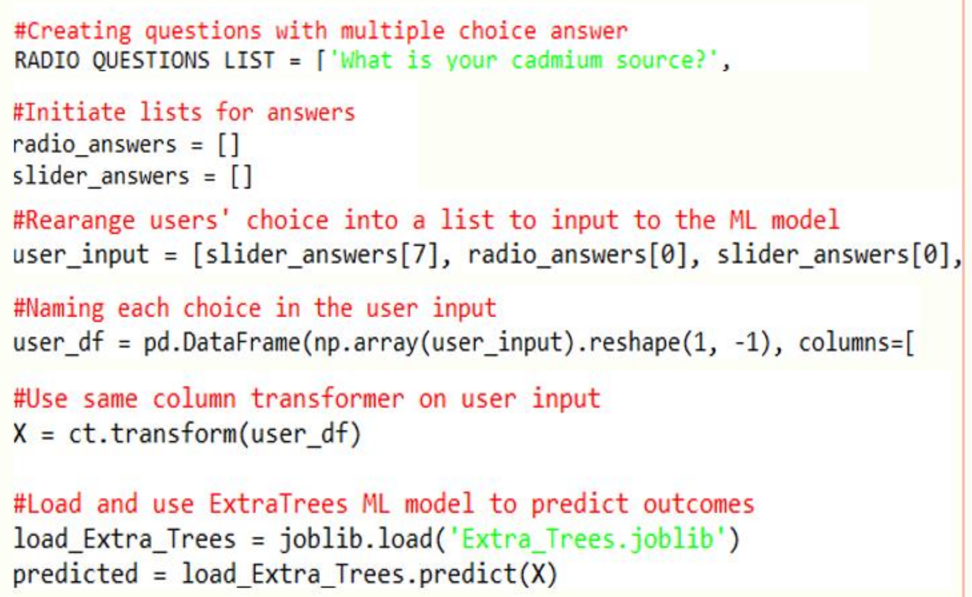


	Growth Temp (Celsius)	Metal_source	Metal_mmol (mmol)	Chalcogen_mmol (m
0	200.0	cadmium stearate	0.15	

Predicted diameter is 4.5200000000000005 . Predicted absorbance max is 596.3333333333334 . Predicted emission is 593.0 .



```
/hotdots/Streamlit UI$ streamlit run main.py

You can now view your Streamlit app in your browser.

Network URL: http://172.31.240.249:8502
External URL: http://73.53.45.231:8502
```



```
#Creating questions with multiple choice answer
RADIO QUESTIONS LIST = ['What is your cadmium source?',

#Initiate lists for answers
radio_answers = []
slider_answers = []

#Rearange users' choice into a list to input to the ML model
user_input = [slider_answers[7], radio_answers[0], slider_answers[0],

#Naming each choice in the user input
user_df = pd.DataFrame(np.array(user_input).reshape(1, -1), columns=[

#Use same column transformer on user input
X = ct.transform(user_df)

#Load and use ExtraTrees ML model to predict outcomes
load_Extra_Trees = joblib.load('Extra_Trees.joblib')
predicted = load_Extra_Trees.predict(X)
```

Narrative:

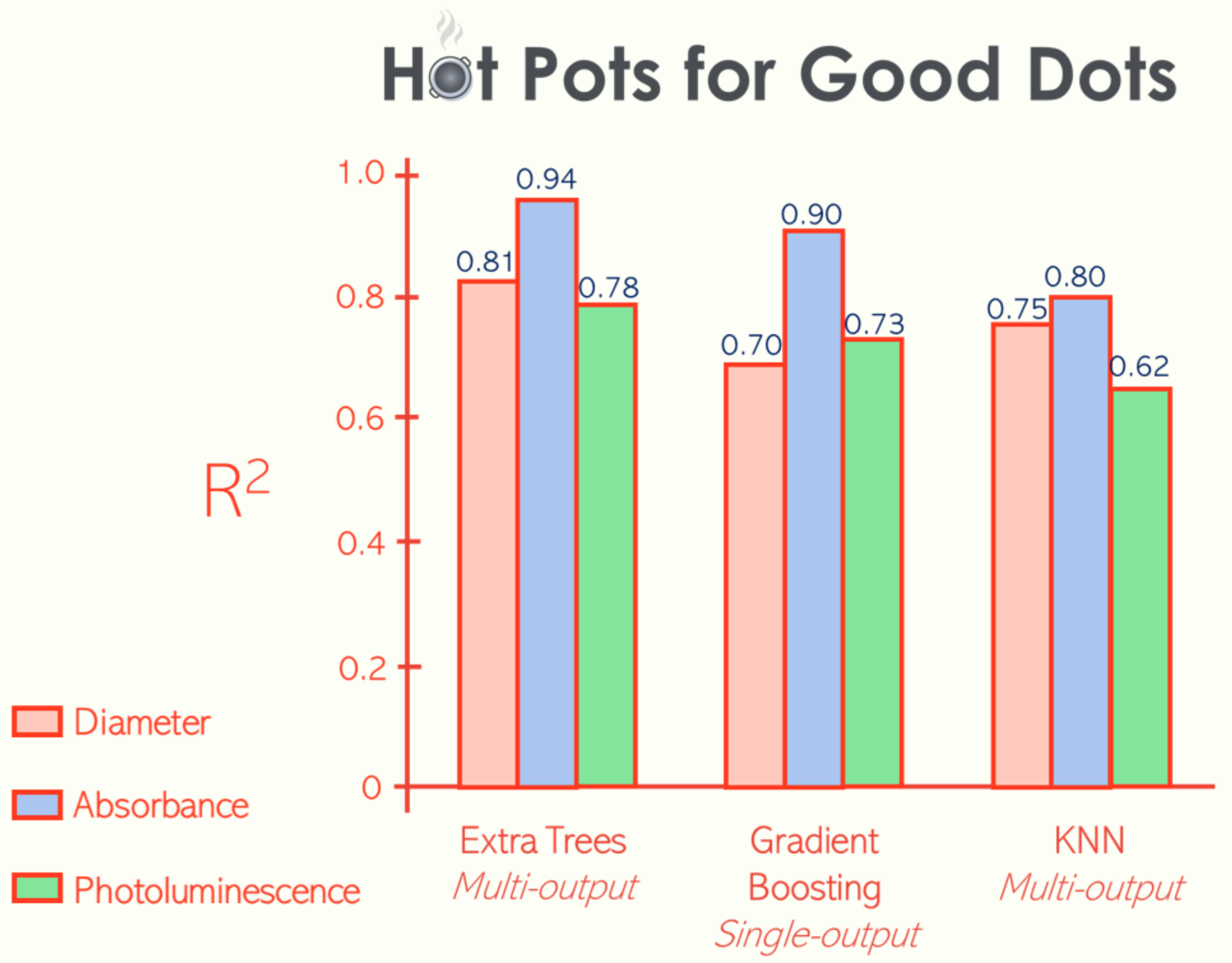
We used Streamlit as the user interface.
This is a fairly new package, where users can open the Streamlit app in their browser using the command line.
The interface is also very controlled, where the user can choose their input from the multiple-choice questions or adjust the amount of chemical used from the sliders.

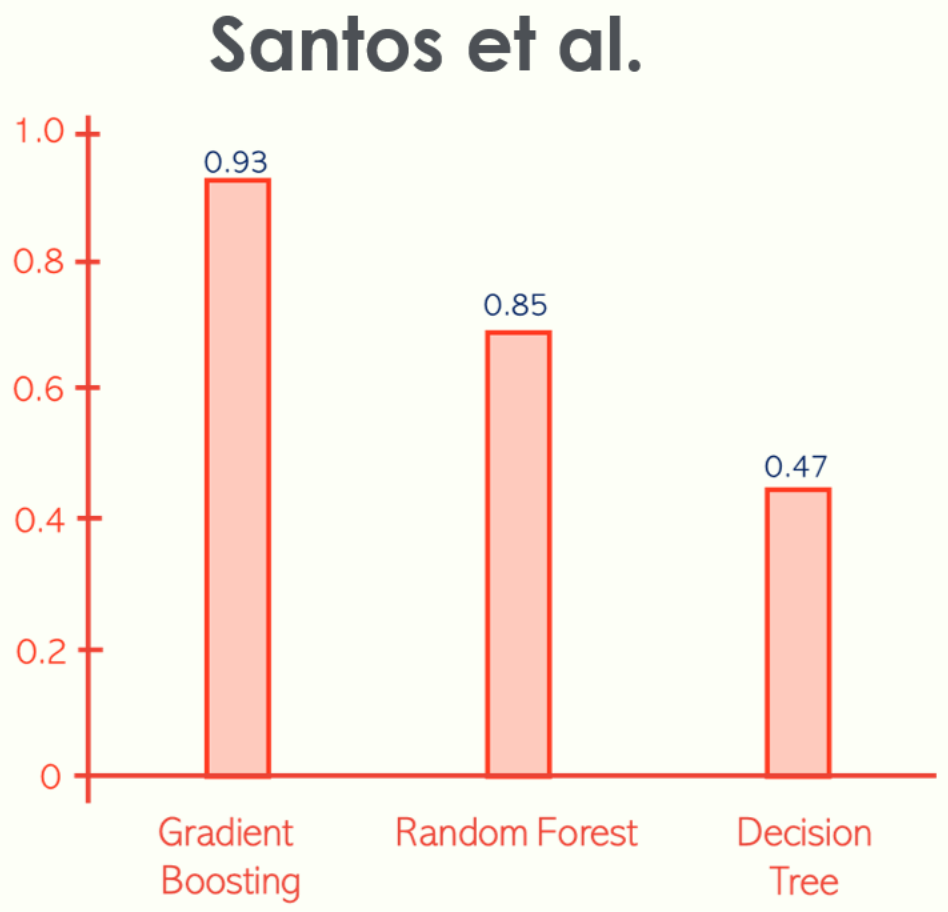
In terms of coding, Streamlit package is also very easy to use. You can create questions in a different styles and add the answers as a list.

When the user input their choices, the choices will be put into a list, that list will then be transformed just as we scaled, encoded our dataset.

The final user input is stored as a list and will be ran through out Extra Trees predictor.
This will give the user the predicted diameter, absorbance max, and photoluminescence in real time.

Model Performance vs. *Santos et al*



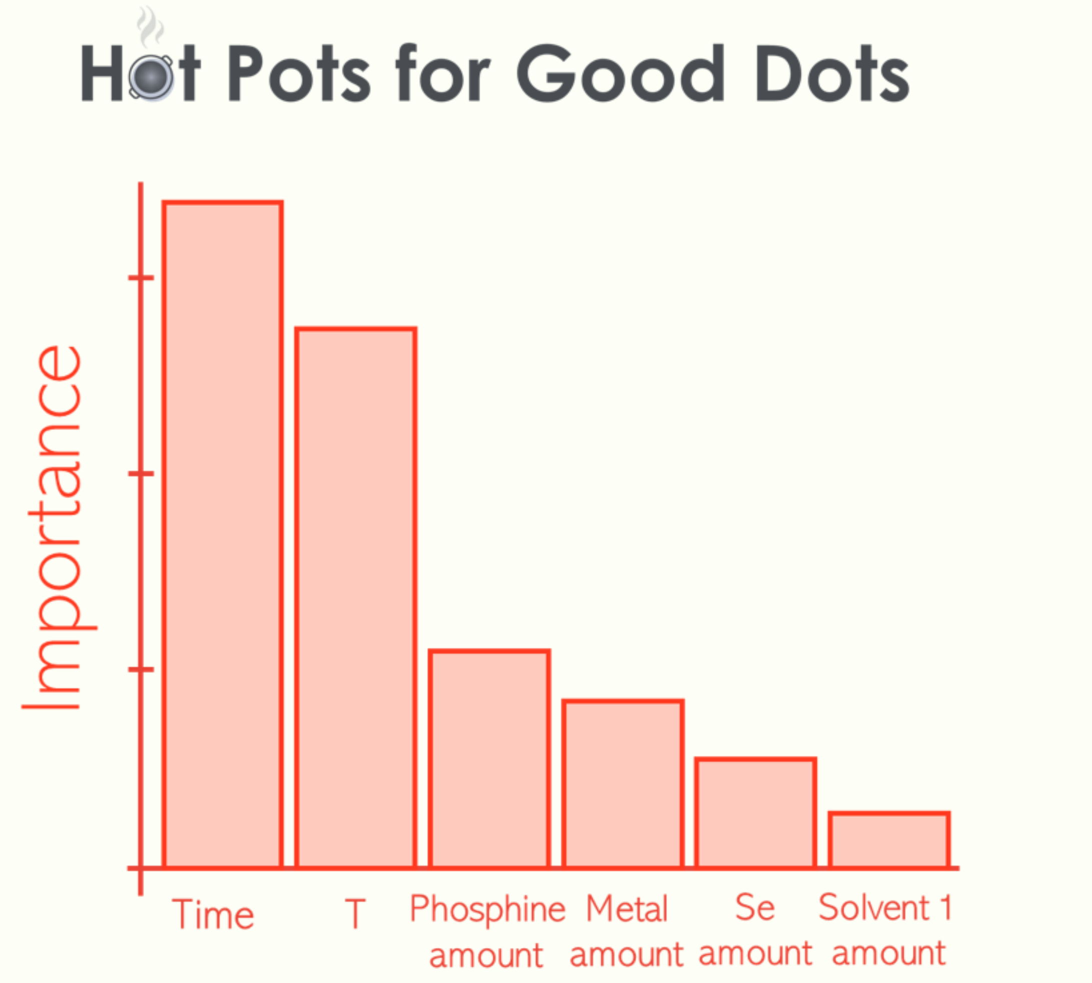


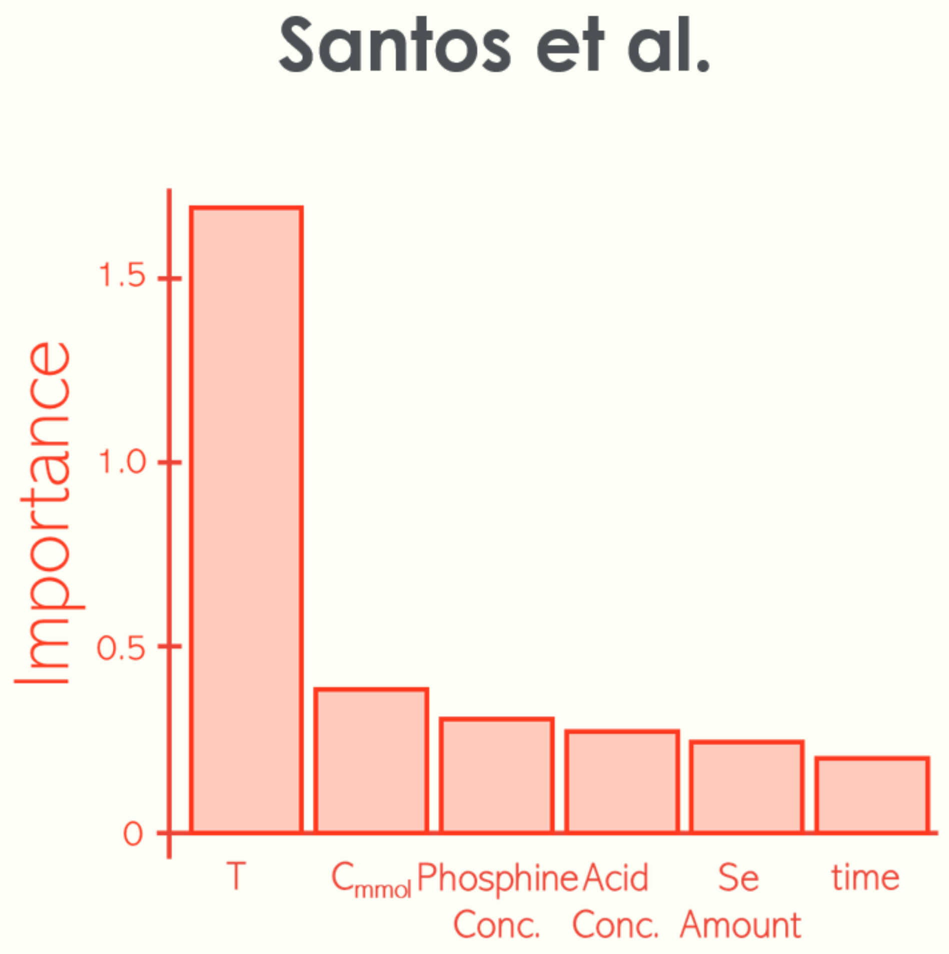
ML models with best performance

Narrative:

The right graphs are from the paper: From the random forest and gradient boosting machine algorithms, the most influential parameters on the final diameter of the quantum dots were the time of reaction, temperature, and metal precursors. However for our case, the important feature of random forest model and GBM weren't the same. For GBM Growth temp, time, and phosphine_mmol and for random forest : time, growth temp, and phosphine_mmol

Feature Importance vs. *Santos et al*





From Random Forest model

Narrative:

The right graphs are from the JPCC paper: From the random forest algorithms, the most influential parameters on the final diameter of the quantum dots were the temperature of reaction.

However, for our case, the important feature of random forest model is time growth.

We will need to consider some chemical aspects of quantum dot synthesis in order to interpret this comparison more correctly.

Conclusions & Future Directions

We created a good predictor with only 36 papers and 233 datarows

We made a handy user interface using *Streamlit*

Our analysis gave different results from the original literature

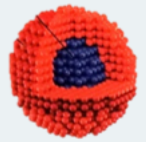
Expand the dataset

_Extract data

_Create data

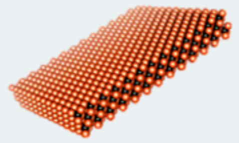
Apply on other materials

_Other QDs



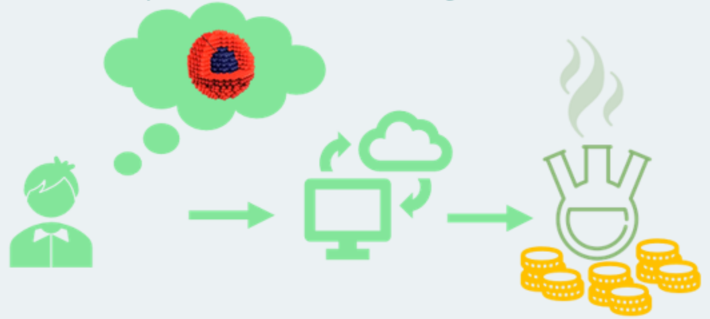
InP, core/shell QDs, perovskites

_CdSe Nanoplatelets



Price/Financial consideration

_Create a model that calculates the most inexpensive procedure to synthesize QDs



Narrative:

In conclusion, we created a good predictor with only 36 papers and 233 datarows, which is comparable with the JPCC paper, which is the first and only ML paper on CdSe quantum dots.

We also made a handy user interface so that chemists can easily use Finally, Our analysis interestingly gave different results from the JPCC paper, to which we will need to consider some chemical aspects to explain.

For our future work,
We can expand the dataset by extracting data from the literature or create data by our own lab.

We can also apply this approach to other materials such as other QDs or nano-platelets, which was actually our initial project idea.

Last but not least, we also consider making a cost predictor that can give the cheapest procedure to synthesize quantum dots.

Thank you

Dr. David Beck

Dr. Stephanie Valleau

Our TAs: Sabiha and Nisarg

Our CHEM E 545 & 546 peers

Narrative:

And that's the end of our presentation today.

want to say thank you to Dr. Beck, Dr Valleau, our Tas, and our classmates.