Using Machine Learning Techniques to Solve Problems in Materials Science and Engineering

Eryn Dennis[1,2], Simon Billinge, Ph.D.[1]

[1]Billinge Group, Department of Applied Physics & Applied Mathematics, Columbia University

[2]College of Information Sciences & Technology, The Pennsylvania State University

Amazon SURE Program 2023

Abstract

- Machine Learning (ML) is a useful tool for Materials Science & Engineering (MATSE) students; teaching ML would be most impactful if lessons are relevant to students
- We assembled a software infrastructure that professors can interact with to build a course
- We created a database of ML educational examples (edexes) that are labeled and developed functions to select, sequence, and upload them to Canvas for professors
- Professors can draw from and contribute to the database where edexes are stored

Methods

Step One: The edex is created and placed in the database

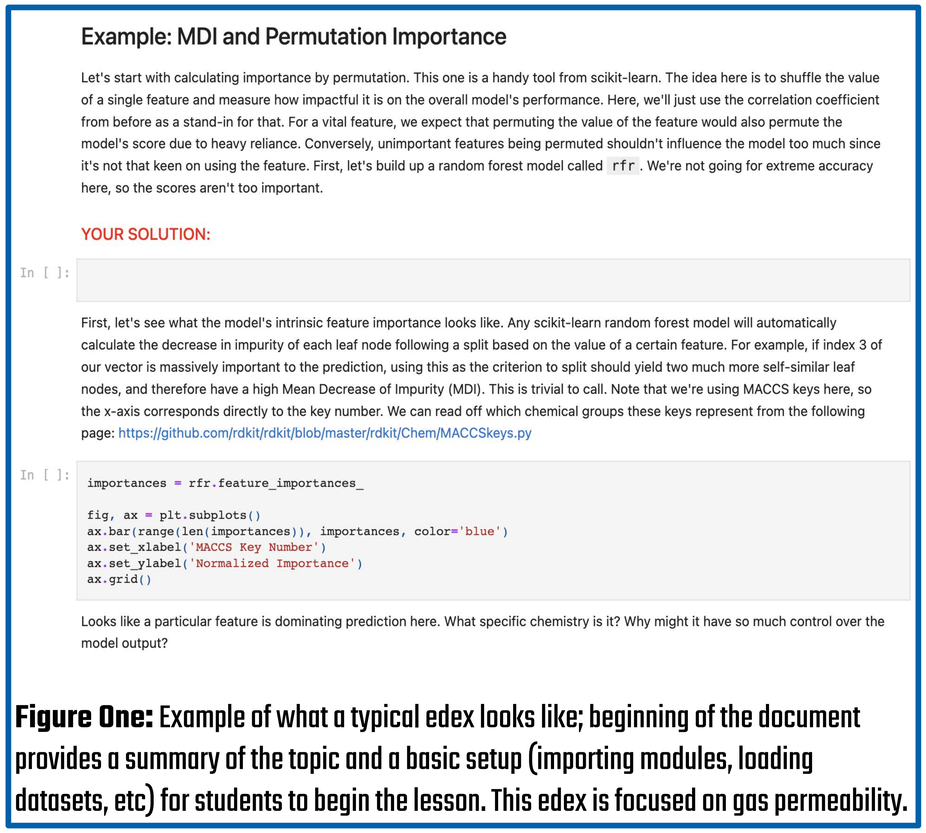
Step Two: The edex is labeled by the creator based on its content

Step Three: Once the professor needs edexes, they're all passed through the filtering function

Step Four: If an edex passes the filter, it is then passed through the sequencing function

Step Five: Steps 3 & 4 are repeated until the professor is satisfied; then the professor chooses which edexes they want to use from the suggested edexes in the list

Step Six: The chosen edexes are confirmed by the professor, and are then uploaded to Canvas in their respective modules



Example: MDI and Permutation Importance

Let's start with calculating importance by permutation. This one is a handy tool from scikit-learn. The idea here is to shuffle the value of a single feature and measure how impactful it is on the overall model's performance. Here, we'll just use the correlation coefficient from before as a stand-in for that. For a vital feature, we expect that permuting the value of the feature would also permute the model's score due to heavy reliance. Conversely, unimportant features being permuted shouldn't influence the model too much since it's not that keen on using the feature. First, let's build up a random forest model called `rfr`. We're not going for extreme accuracy here, so the scores aren't too important.

YOUR SOLUTION:

In [ ]:

First, let's see what the model's intrinsic feature importance looks like. Any scikit-learn random forest model will automatically calculate the decrease in impurity of each leaf node following a split based on the value of a certain feature. For example, if index 3 of our vector is massively important to the prediction, using this as the criterion to split should yield two much more self-similar leaf nodes, and therefore have a high Mean Decrease of Impurity (MDI). This is trivial to call. Note that we're using MACCS keys here, so the x-axis corresponds directly to the key number. We can read off which chemical groups these keys represent from the following page: https://github.com/rdkit/rdkit/blob/master/rdkit/Chem/MACCSkeys.py

In [ ]:

```
importances = rfr.feature_importances_

fig, ax = plt.subplots()
ax.bar(range(len(importances)), importances, color='blue')
ax.set_xlabel('MACCS Key Number')
ax.set_ylabel('Normalized Importance')
ax.grid()
```

Looks like a particular feature is dominating prediction here. What specific chemistry is it? Why might it have so much control over the model output?

Figure One: Example of what a typical edex looks like; beginning of the document provides a summary of the topic and a basic setup (importing modules, loading datasets, etc) for students to begin the lesson. This edex is focused on gas permeability.

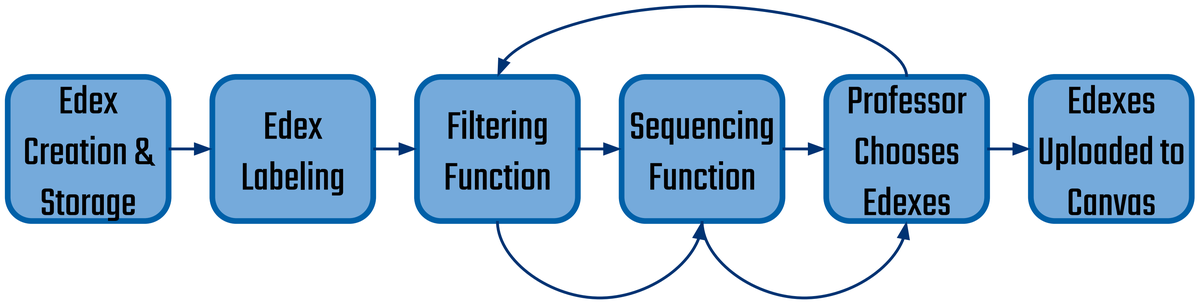


Figure Two: Our process for selecting, sequencing, and uploading edexes. This process ensures that professors can easily decide which edexes to use for their courses. Refer to the Methods section for more information.

Results

- Designed a process for edex selection, sequencing, and uploading
- Produced a JSON file to store information about each edex
- Successfully created a filtering function
- Started producing an edex

Conclusion

- Edexes are a great way for MATSE students to practice machine learning techniques
- Software infrastructure and ecb tools make course creation faster and more efficient for professors
- Easy to collaborate with other professors and universities
- Software can be expanded to an openly shared community website

Acknowledgements: I would like to thank the SURE program for giving me the opportunity to do this research, Professor Billinge for allowing me to work on this project and in his lab, and all the members of the Billinge Group for mentoring me and helping me navigate the lab.