Krátos Batteries

Background:

Using existing data to predict volume change and capacity of new electrode materials

- Small Volume Change → Longer Battery Life
- Bigger Capacity → More Energy Stored

Predicting them is useful:

1. Guide experiments seeking to develop novel materials for battery applications
2. Perform a quick screening before starting synthesis procedures

User Cases

Krátos

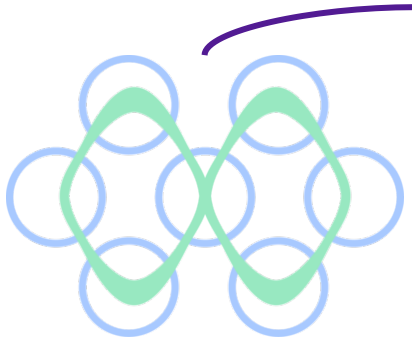
User will input electrolyte parameters of choice for battery design

ML

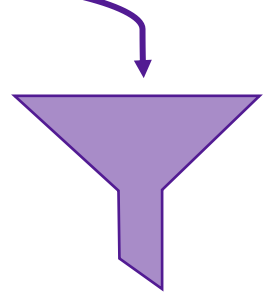
- Volume Expansion
- Volumetric Capacity
- Gravimetric Capacity

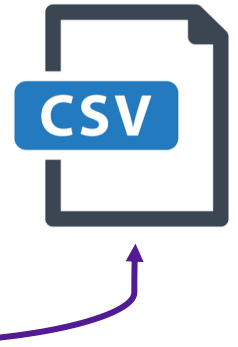
ML = Neural Network and SVR

Template for Extraction of Data from *The Materials Project Database*



Materials Project Database

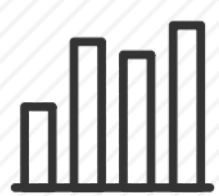




Sort by ID, Elements, Formulas, etc.

Future Developments/Versions

- Classify electrode material for specific battery design.
- Look for important trends in predicted parameters.



Software Engineering

Software Engineering Tools Used:

Unit Testing:
Allows us to segregate each part of the program and test that the individual parts are working correctly

- Verifies the accuracy of each functional unit
- Improves code quality
- Provides documentation

Unit Test Results

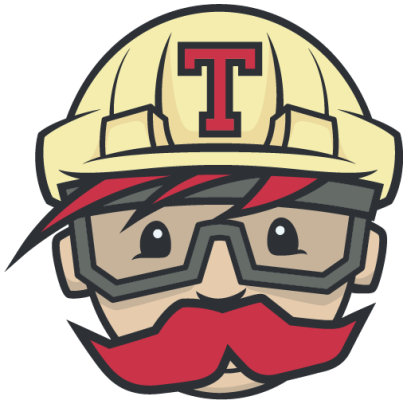
```
----------------------------------------
Ran 7 tests in 190.340s

OK
```

Travis Continuous Integration: build passing
Allows us to continuously integrate code into a single shared and easy to access repository

- Better test reliability
- Allows for easy maintenance and updates

Coveralls.io: coverage 100%
Takes the build data from CI service our project uses, and provides constant updates and statistics on code coverage



Results – Data Extraction

Krátos

Battery Data Extraction

Extracting data from materialsproject.org

We will first import some relevant modules required for extracting data from the materialsproject.org. The submodule for any data extraction methods is known as `data_extract`:

In [4]:
```
import data_extract
import pandas as pd
```

The `get_bat_dat` function is able to mine the entire materials project for all materials classified as battery materials and will return a dataframe of all battery materials and their properties.

NOTE The function takes an API key as an input to access the materialsproject database. We suggest generating your own API key from the dashboard of materials project.

In [5]:
```
data_extract.get_bat_dat('EcOxTpa0ymKFe24R')
```

Out[5]:

	Reduced Cell Formula	Average Voltage (V)	Min Voltage (V)	Max Voltage (V)	Number of Steps	Min Instability	Gravimetric Capacity (mAh/g)	Volumetric Capacity (Ah/L)	Working Ion	Min Fraction
Battery ID										
mp-504791_Li	P(WO4)2	2.325185	2.325185	2.325185	1	0.000000	50.228609	269.262771	Li	0.0
mp-763480_Li	P3W2O13	3.291748	3.291748	3.291748	1	0.000097	39.674483	154.215698	Li	0.0

In [10]:
```
data_extract.update_check('EcOxTpa0ymKFe24R')
```

```
The Current BatteryData.csv file is up to date!
```

```
# making a list of all the battery IDs
all_bat_ids_list = (mpr._make_request('/battery/all_ids'))

# making an empty dataframe to hold all of the battery data
all_battery_dataframe = pd.DataFrame([])

# looping through every id in the list of all battery IDs
for batt_id in all_bat_ids_list:
    # gets all the data for one battery id and stores it in a result df
    result_bat_id = pd.DataFrame(mpr.get_battery_data(batt_id))
```

Customizable!

Instead of the path */battery/all_ids*:

- Formula (e.g. FeO_2)
- Elements Present (e.g. Li-Co)
- Different class
- Etc.

Results – Data Extraction

Krátos

MAGPIE

Magpie works by using a chemically diverse list of attributes, which we demonstrate are suitable for describing a wide variety of properties, and a novel method for partitioning the data set into groups of similar materials to boost the predictive accuracy.

Table S1. Elemental properties used to compute elemental-property-based attributes. Elemental property is taken from that dataset available with the Wolfram programming language,[8] unless otherwise specified

Atomic Number	Mendeleev Number[9]	Atomic Weight	Melting Temperature	Column
Row	Covalent Radius	Electronegativity*	# s Valence Electrons	# p Valence Electrons
# d Valence Electrons	# f Valence Electrons	Total # Valance Electrons	# Unfilled s States†	# Unfilled p States†
# Unfilled d States†	# Unfilled f States†	Total # Unfilled States†	Specific Volume of 0 K Ground State‡	Band Gap Energy of 0 K Ground State‡
Magnetic Moment (per atom) of 0 K ground state‡		Space Group Number of 0 K Ground State‡		

Elemental-Property-Based Attributes (115 in total)

Most of the attributes created using our method are based on statistics of the elemental properties listed in Table S1. For each property, the minimum, maximum, and range of the values of the properties of each element present in the material is computed along with the fraction-weighted mean, average deviation, and mode (i.e. the property of the most prevalent element). The mean and average deviation are calculated using the following formulae:

$$\bar{f} = \sum x_i f_i \quad \text{(S1)}$$

$$\hat{f} = \sum x_i |f_i - \bar{f}| \quad \text{(S2)}$$

Results – Data Process

Raw Data

↓

Transfer String to Binary Value
(LabelBinarizer)

↓

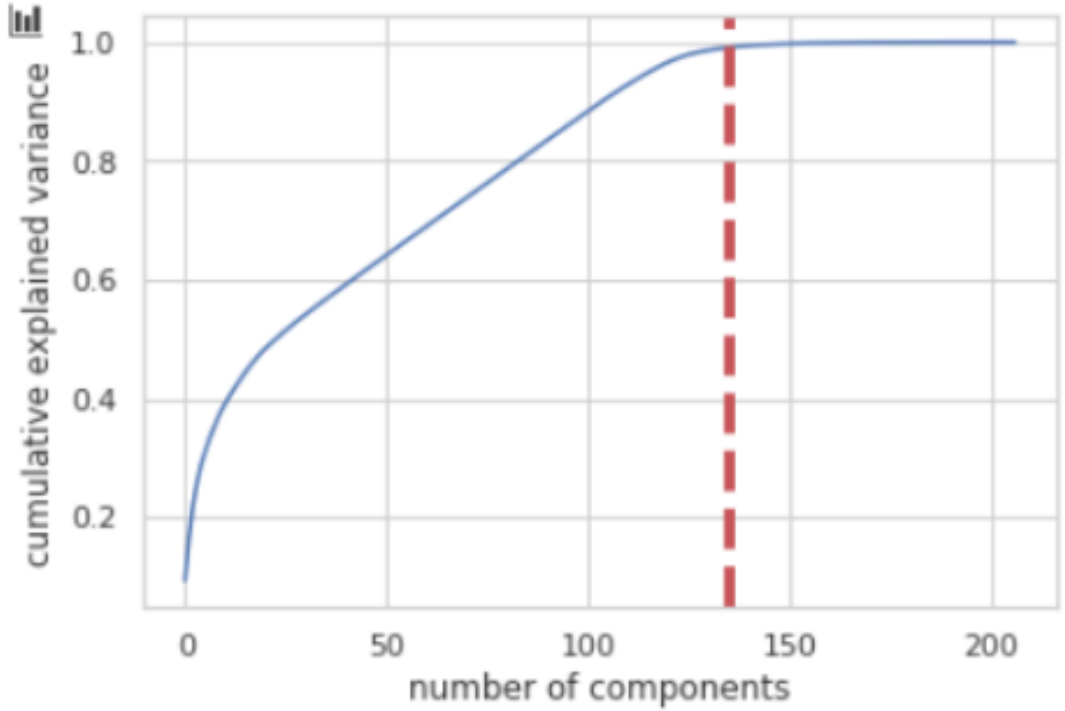
Normalization (MinMaxScaler)

↓

Feature Selection (RFE)

↓

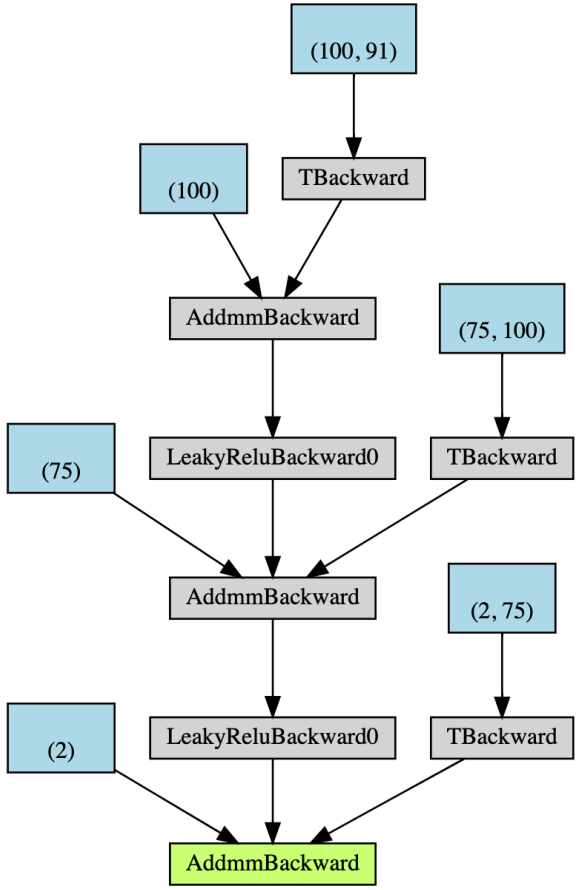
Feature Extraction (PCA)



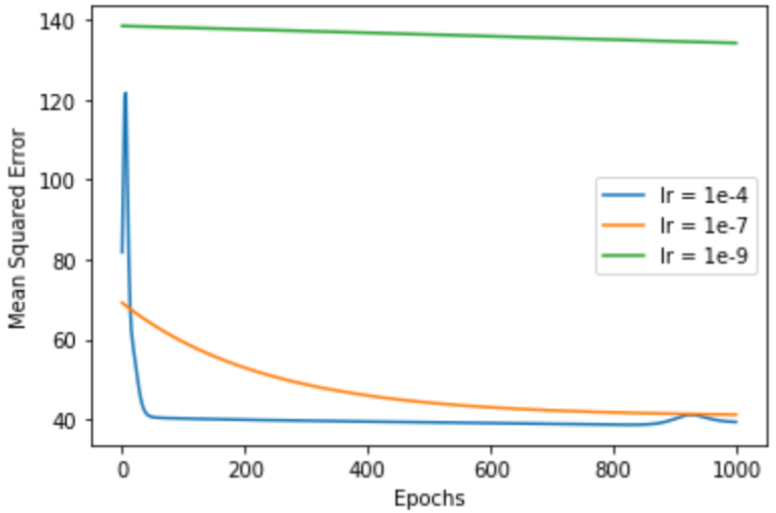
Results – Prediction (NN)

Key Points:

- Neural network to predict gravimetric and volumetric for battery electrodes
- Sequential model
- Used PCA analysis
- 2 hidden layers
- Model put through 1000 epochs to reduce MSE to make final predictor model



Schematic of NN



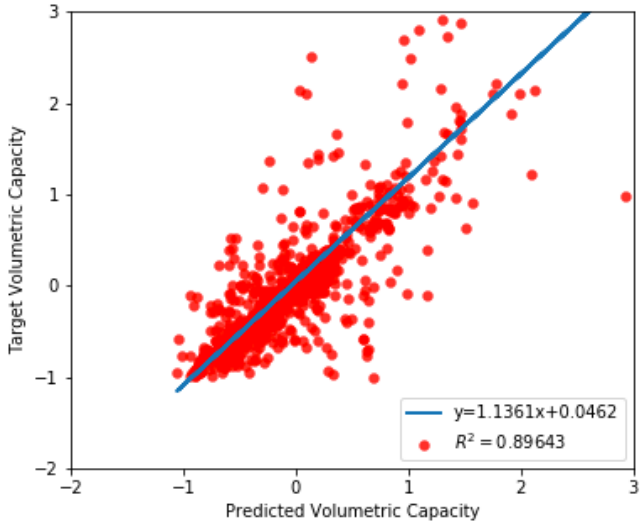
Loss function of NN

Results – Prediction (SVR)

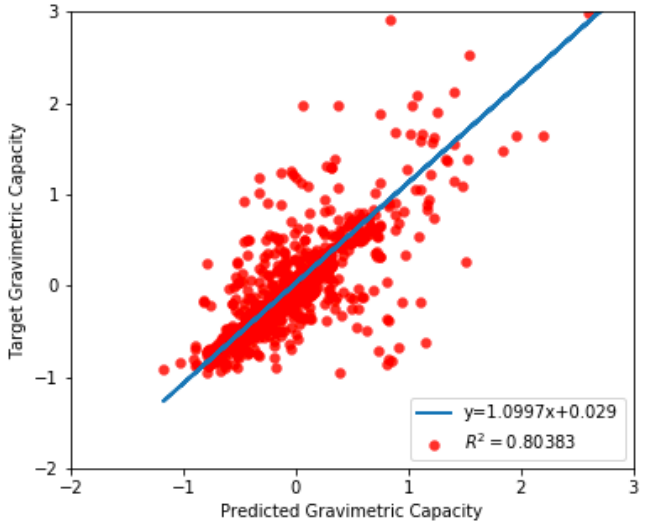
Key Points:

The main advantages of SVR is that its computational complexity does not depend on the dimensionality of the input space, which means it is more effective in high dimensional spaces. Additionally, it has excellent generalization capability, with high prediction accuracy

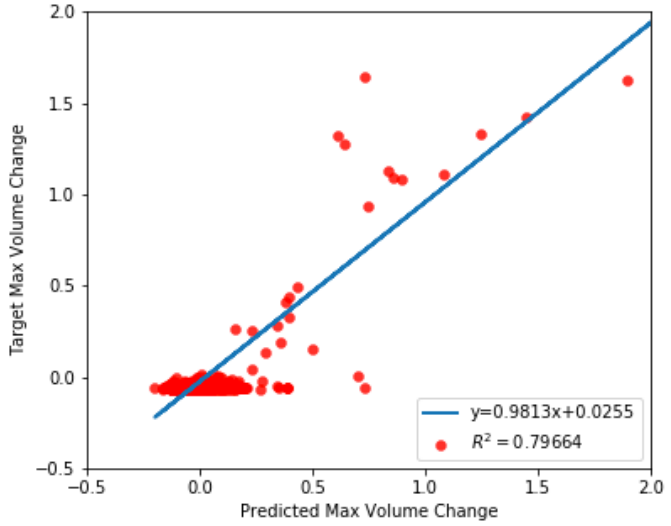
Volumetric Capacity



Gravimetric Capacity



Volume Change



Looking Back/Forward

Lessons Learned:

- “We wish we had followed Dave's advice and used test-driven development from day 1” == **True**
- **FORCE PUSH IS EVIL**
  - Importance of good collaborative techniques (branches, merge, etc.)
- Design entire structure before starting
  - Avoid ”add as you go” approach

Future Direction:

- Make a GUI to make package resemble a tool
- Allow for more data manipulation by User

Acknowledgements

DIRECT
Data Intensive Research Enabling Clean Technologies

We would like to thank Professors Beck and Cao (and all TAs) for their continued support, invaluable advice and most importantly their patience!