Project Check-in: May 1st, 2024

So far up to now, I was able to import the data and organize it by dropping irrelevant attributes so that a ML algorithm can be ran on it. This involves extensive cleaning of the data as the dataset has many irrelevant and broken columns. The first challenge is that the datasets are stored as JSON files. Since I have never worked with JSON files before, the project started out by identifying how information is stored in JSON files and how they can be imported so that they can be put into a ML algorithm. Looking at the JSON files, from the “root”, there are a maximum of 5 layers. In the first layer, there is a “filter” and a “results”. Now, I am interested in the results and so when I read in the JSON file, I read in just the results portion using the following code.

```
data=json.load(open('dataset.json'))
```

However, because it is currently in the json format, it was then reformatted into a pandas data-frame. When we do this, I only wanted the “results” portion and so I used the following code to read it in as a pandas dataframe.

```
df=pd.json_normalize(data["results"])
```

Upon assessment of the dataset, there were a total of 51 columns. However, of the 51 columns, many contained obviously irrelevant information such as the doi, author name, and date. Therefore, these columns were manually dropped.

Specifically, these columns were the following.

```
['adsorption_measurement.doi', 'adsorption_measurement.external_note',

'adsorption_measurement.internal_note',

'adsorption_measurement.bulk_surface_property_set.secondary_bulk_class',

'adsorption_measurement.bulk_surface_property_set.second_layer_composition',

'adsorption_measurement.bulk_surface_property_set.first_layer_composition',

'adsorption_measurement.emn_user.first_name', 'adsorption_measurement.emn_user.last_name',

'adsorption_measurement.emn_user.affiliation', 'adsorption_measurement.emn_user.email',

'adsorption_measurement.approver', 'adsorption_measurement.adsorption_site',

'adsorption_measurement.bulk_surface_property_set.lattice_constant',

'adsorption_measurement.bulk_surface_property_set.cell_symmetry',

'adsorption_measurement.bulk_surface_property_set.secondary_bulk_class.name',
'adsorption_measurement.bulk_surface_property_set.first_layer_composition.name',

'adsorption_measurement.bulk_surface_property_set.second_layer_composition.name',

'adsorption_measurement.bulk_surface_property_set.facet']
```

Additionally, attributes that did not contain actual data were dropped. This includes attributes where every single instance had TRUE for that attribute.

Specifically, these attributes were dropped.

```
['adsorption_measurement.is_most_stable_site',

'adsorption_measurement.adsorbate_species.name',

'adsorption_measurement.adsorbate_species.elemental_formula',

'adsorption_measurement.bulk_surface_property_set.is_stretched',

'adsorption_measurement.bulk_surface_property_set.is_compressed',

'adsorption_measurement.bulk_surface_property_set.nano_number_of_atoms',

'adsorption_measurement.bulk_surface_property_set.bulk_surface_material.elemental_formula',

'adsorption_measurement.adsorbate_fraction.fraction',

'adsorption_measurement.adsorption_reference_species_set',

'adsorption_measurement.bulk_surface_property_set.primary_bulk_class.name']
```

With these attributes now excluded, the number of columns decreased from 51 initially to 23 attributes which, now all of them containing chemically and physically relevant data.

One thing that has gone particularly well was using *Jupyter notebooks*, thanks to all the practice we have been doing until now. I was able to read in JSON files easily thanks to all the practice I have had throughout the semester. Even though I haven't worked with JSON files before, I was able to find solutions by myself.

The challenges I currently face is converting some of the chemically relevant attributes that are currently remaining, into formats that can be read by the ML algorithm. For instance, some attributes are trying to express chemical structures through symbols. For instance,

[O][C][C]([C][O])O, O[CH][C]([CH]O)O, [O][C][C]([C][O])O

Now, at its current form, this doesn't provide chemically relevant data. I need to think of a way to convert this. Additionally, there are some attributes with missing datapoints. If there are too many missing datapoints for a given attribute, I probably have to drop them. However, if there are only a few missing points, I should think of a way to resolve that issue.

As a revised timeline, I would like to use the next week to prepare a clean dataset including reorganizing some of the structural information so that it can be read into a ML algorithm. Then, in the following week, I will start running the actual algorithm on the dataset. Then, in the final week, I will complete the write-up.

Currently, I need some help with how to deal with missing data points and pointers on when to drop them completely and when to fix them. Additionally, I need some ideas on how to extract important information from molecular structures.