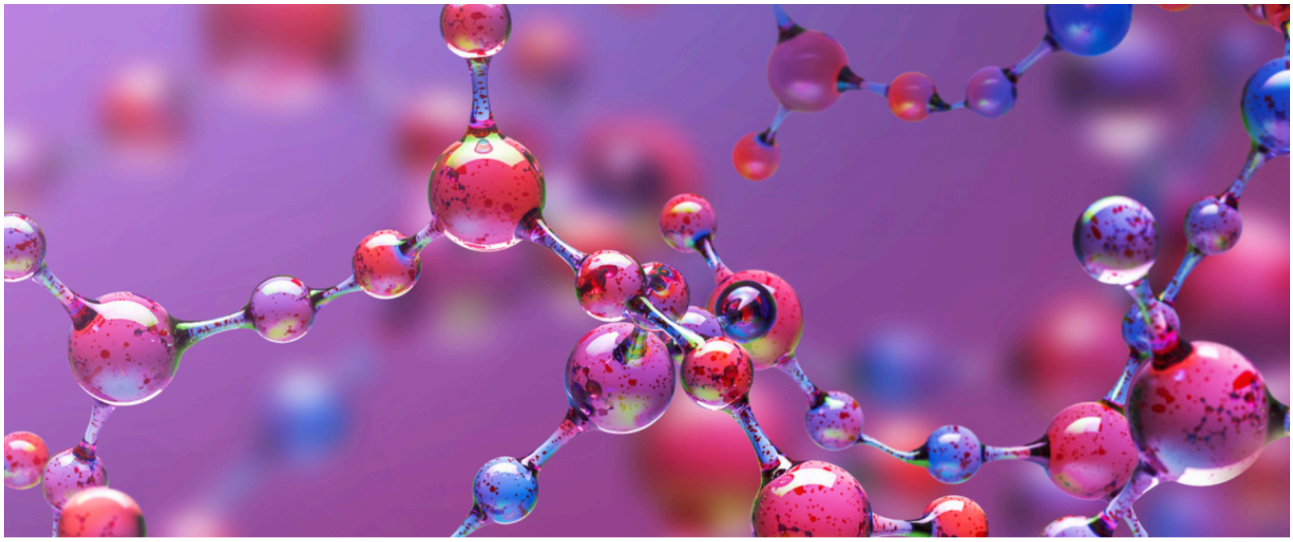
Name: Vidushi Katare
Registration number: 12016318
Section: K20UP

EDA PROJECT



Predicting Molecular Properties

This project aims to predict interactions between atoms. Nuclear Magnetic Resonance (NMR) is technology which uses the same principles as of an imaging technologies like MRI to understand the structure and dynamics of proteins and molecules. Around the world researchers conduct NMR experiments to get a further understanding of the structure and dynamics of molecules, across areas like environmental science, pharmaceutical science, and materials science. Gaining insights into a molecule's structure and dynamics, using NMR, depends on the ability to accurately predict "scalar couplings". These are effectively the magnetic interactions between a pair of atoms. The strength of which depends on intervening electrons and chemical bonds making up a molecule's three-dimensional structure. Using methods from quantum mechanics like state-of-the-art, it is possible to calculate scalar coupling constants accurately, given only a 3D molecular structure as input. However, these quantum mechanics calculations take days or weeks per molecule and thus are extremely expensive , and therefore have limited applicability in day-to-day workflows. A fast and reliable method to predict these interactions is required for the medicinal chemists to gain structural insights faster and cheaper, this will enable scientists to understand how the 3D chemical structure of a molecule affects its properties and behaviour. As a result, such tools will enable researchers to make faster progress in a wide range of important domains and problems faced by the people, like designing molecules to carry out specific cellular tasks, or designing better drug molecules to fight disease.

Dataset Overview

- test.csv - the test set; same info as train, without the target variable
- train.csv - the training set, where the first column (molecule_name) is the name of the molecule where the coupling constant originates, the second (atom_index_0) and third

column (atom_index_1) is the atom indices of the atom-pair creating the coupling and the fourth column (scalar_coupling_constant) is the scalar coupling constant that we want to be able to predict

[The training and test splits are by molecule, so that no molecule in the training data is found in the test data.]

- sample_submission.csv - a sample submission file in the correct format
- structures.zip - folder containing molecular structure (xyz) files, where the first line is the number of atoms in the molecule, followed by a blank line, and then a line for every atom, where the first column contains the atomic element (H for hydrogen, C for carbon etc.) and the remaining columns contain the X, Y and Z cartesian coordinates (a standard format for chemists and molecular visualization programs)
- structures.csv - this file contains the same information as the individual xyz structure files, but in a single file

[There is some additional data as well, provided for the molecules in 'Train'. Like dipole_moments.csv, magnetic_shielding_tensors.csv, mulliken_charges.csv, potential_energy.csv, scalar_coupling_contributions.csv, to get extra insight and features of the molecules.]

After importing necessary libraries and loading our dataset, we will first be trying to get an idea of what our dataset looks like.

In [6]:
```
train.head()
```

Out[6]:

	id	molecule_name	atom_index_0	atom_index_1	type	scalar_coupling_constant
0	0	dsgdb9nsd_000001	1	0	1JHC	84.807599999999994
1	1	dsgdb9nsd_000001	1	2	2JHH	-11.257000000000000
2	2	dsgdb9nsd_000001	1	3	2JHH	-11.254799999999999
3	3	dsgdb9nsd_000001	1	4	2JHH	-11.254300000000001
4	4	dsgdb9nsd_000001	2	0	1JHC	84.807400000000001

In [7]:
```
structures.head()
```

Out[7]:

	molecule_name	atom_index	atom	x	y	z
0	dsgdb9nsd_000001	0	C	-0.0126981359	1.0858041580	0.0080009958
1	dsgdb9nsd_000001	1	H	0.0021504160	-0.0060313176	0.0019761204
2	dsgdb9nsd_000001	2	H	1.0117308430	1.4637511620	0.0002765748
3	dsgdb9nsd_000001	3	H	-0.5408150690	1.4475266140	-0.8766437152
4	dsgdb9nsd_000001	4	H	-0.5238136345	1.4379326440	0.9063972942

There are different csv files in our dataset, like mentioned in the ‘Dataset Overview’, so we are trying to see what our top 5 rows look like in each one of them.

In [9]:
```
train.describe()
```

Out[9]:

	id	atom_index_0	atom_index_1	scalar_coupling_constant
count	4.658147000000000e+06	4.658147000000000e+06	4.658147000000000e+06	4.658147000000000e+06
mean	2.329073000000000e+06	1.335688568866547e+01	5.883966306773917e+00	1.592164991825936e+01
std	1.344691356527209e+06	3.267712409449629e+00	4.993943098105586e+00	3.494197741570011e+01
min	0.000000000000000e+00	0.000000000000000e+00	0.000000000000000e+00	-3.621860000000000e+01
25%	1.164536500000000e+06	1.100000000000000e+01	2.000000000000000e+00	-2.549780000000000e-01
50%	2.329073000000000e+06	1.300000000000000e+01	5.000000000000000e+00	2.281130000000000e+00
75%	3.493609500000000e+06	1.600000000000000e+01	8.000000000000000e+00	7.390655000000000e+00
max	4.658146000000000e+06	2.800000000000000e+01	2.800000000000000e+01	2.048800000000000e+02

Out[5]:

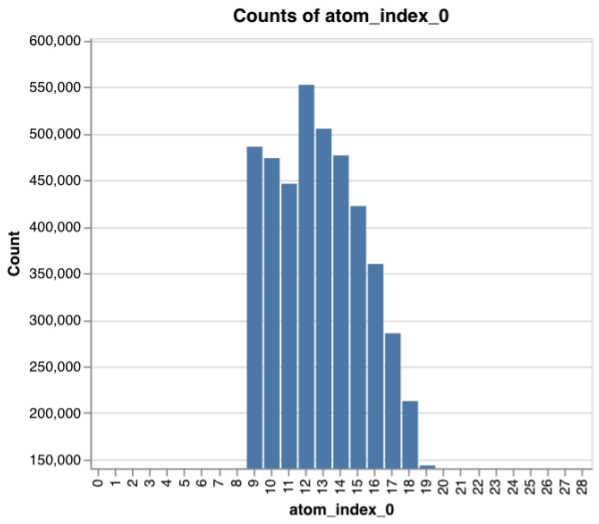
```
3JHC    1511207
2JHC    1140867
1JHC     709133
3JHH     590529
2JHH     377988
3JHN     166613
2JHN     119059
1JHN      43680
Name: type, dtype: int64
```

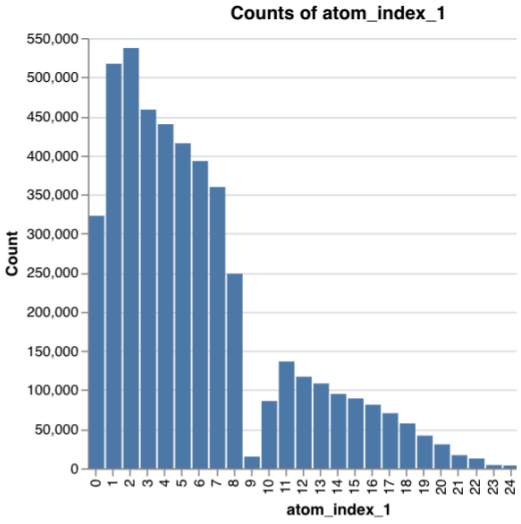
Out[6]:

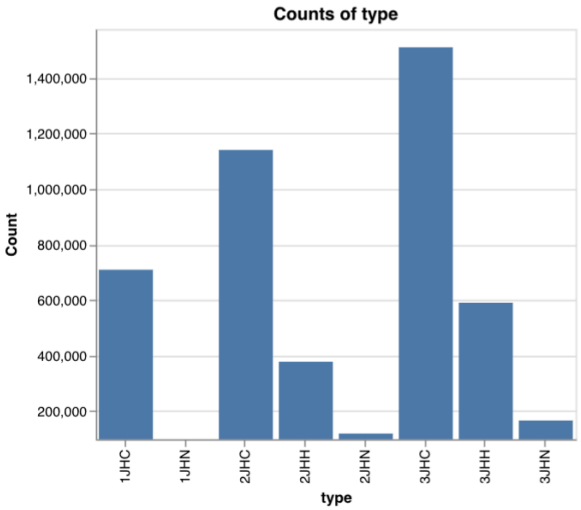
	molecule_name	atom_index	atom	x	y	z
30332	dsgdb9nsd_002222	1	C	0.062675	0.066261	0.041167

We get to see the columns they have and what kind of statistical data they hold. What the distribution of the data is. What values in general would the data in a column would oscillate around. What the minimum and maximum value a column might take. All of this can be seen from a simple method ‘.describe()’. It gives us an overview about the data and its distribution.
We can see the value counts of each type using .value_counts() method, which we will be later visualising. We see what a row looks like in our file using .sample().

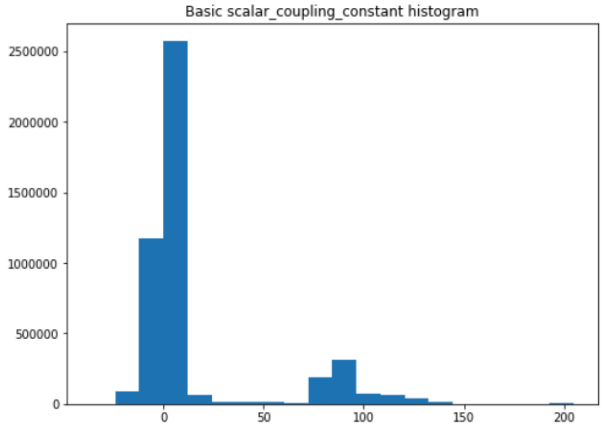
Out[10]:

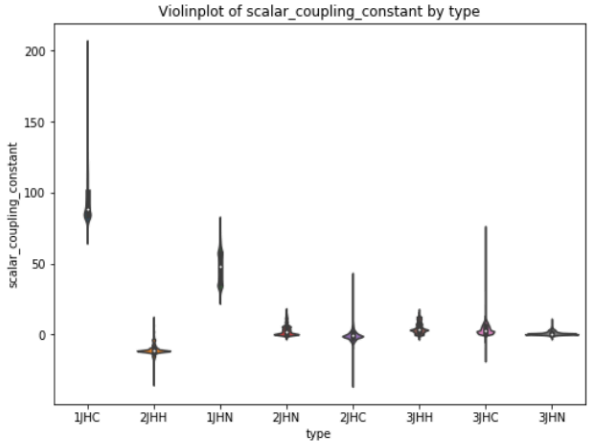






By doing these univariate analysis, we get to see a visual representation of the count of both atom indexed 0 and 1 according to their spins. We also graphed the count on the basis of types. It gives us a clear idea about the difference in the number of count of a type compared to others. This later might help us in making accurate predictions according to the analysis.

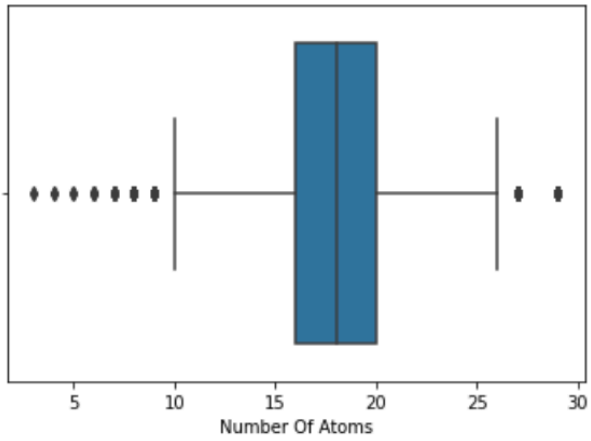


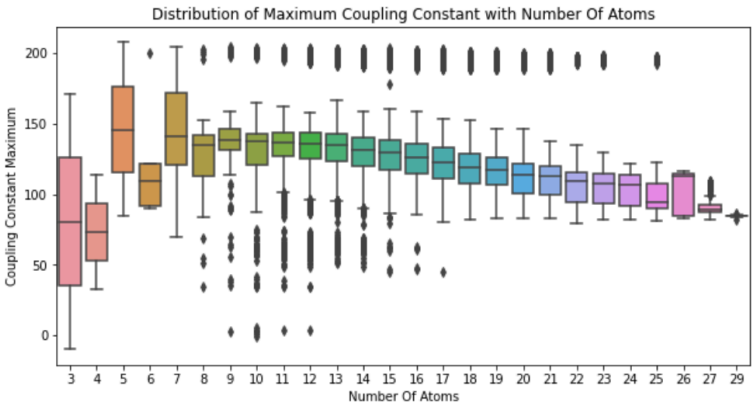


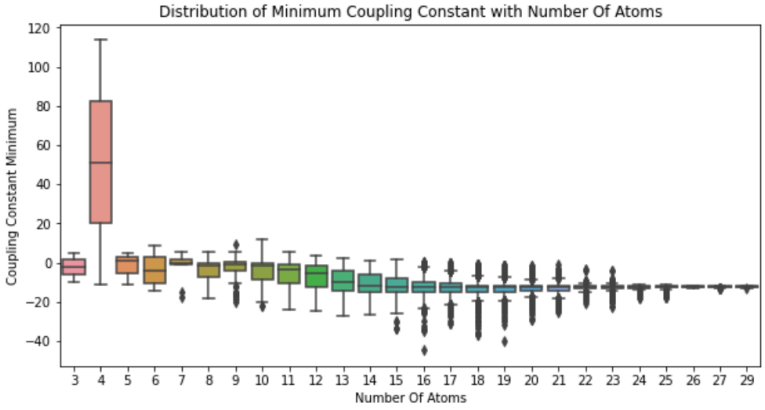
There are many interesting things here:

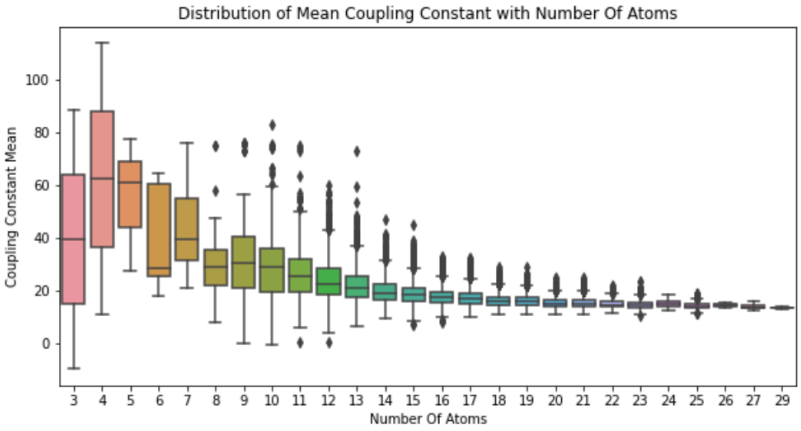
- among first atoms there is a little number of atoms with index lower than 7 or higher than 24;
- among second atoms there is a little number of atoms with index higher than 24. Also index with atom with index 9 in quite rare;
- coupling types are unevenly distributed. There are 3 very popular, 3 quite rare and 2 with medium frequency;
- target variable has a bimodal distribution;
- different coupling types have really different values of target variable. Maybe it would make sense to build separate models for each of them;

To see if **number of atoms** in the molecules has any impact on the coupling constant we plot this box plot.





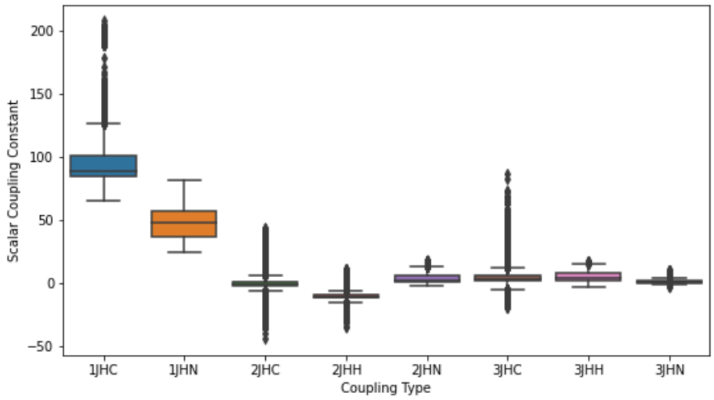


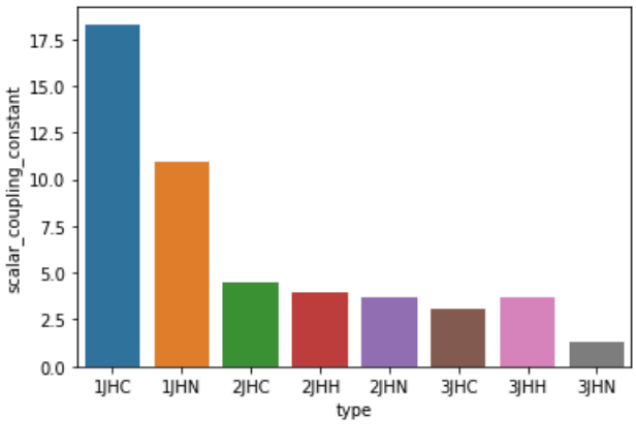


1. Number Of Atoms in the molecule had an impact on the Mean of the coupling constants of the given atoms in molecule.

2. Mean Coupling is reduced as the number of Atoms increase in the molecule. It could be because of some of the bonds could be unstable

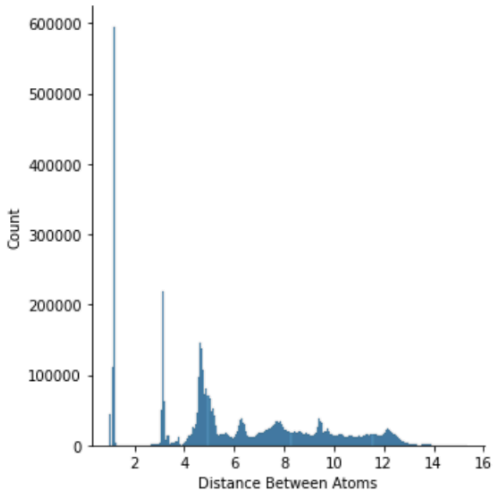
Impact of the **Type of Coupling**:

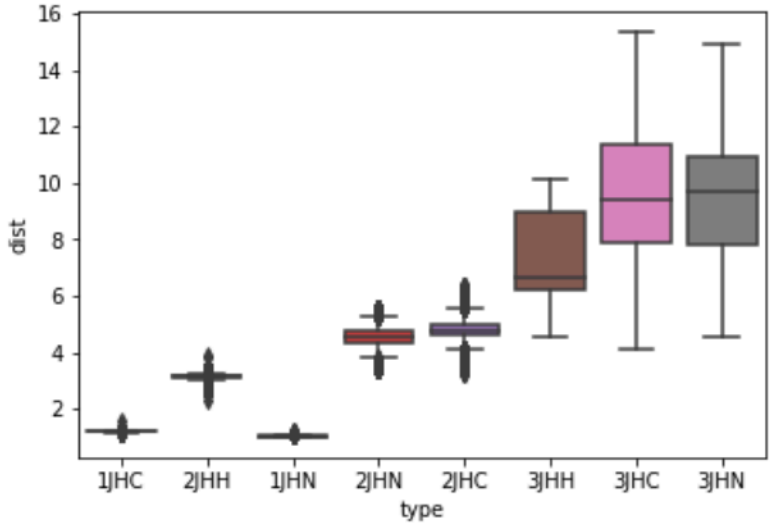




- Scalar Coupling Constant reduces from 1 -> 3
- 1JHC, 1JHN have high coupling constants
- 3JHN --> very less amount of variation in the coupling constants and around zeros.

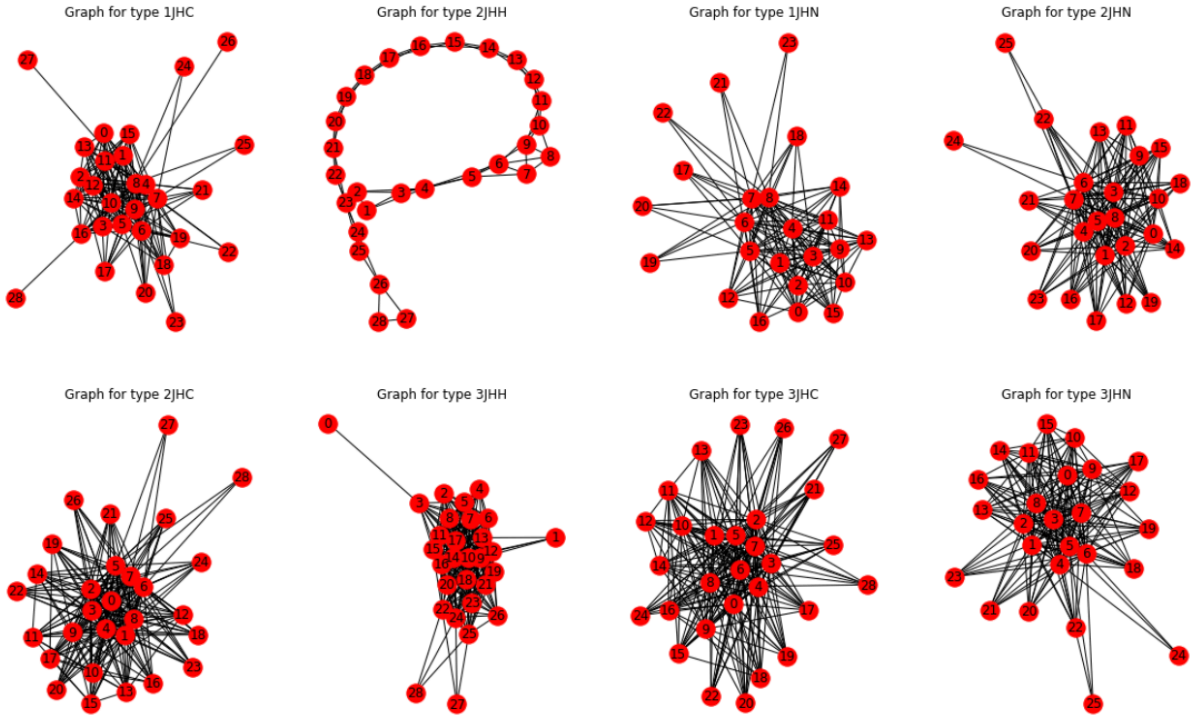
Impact of the **distance between the atoms**:



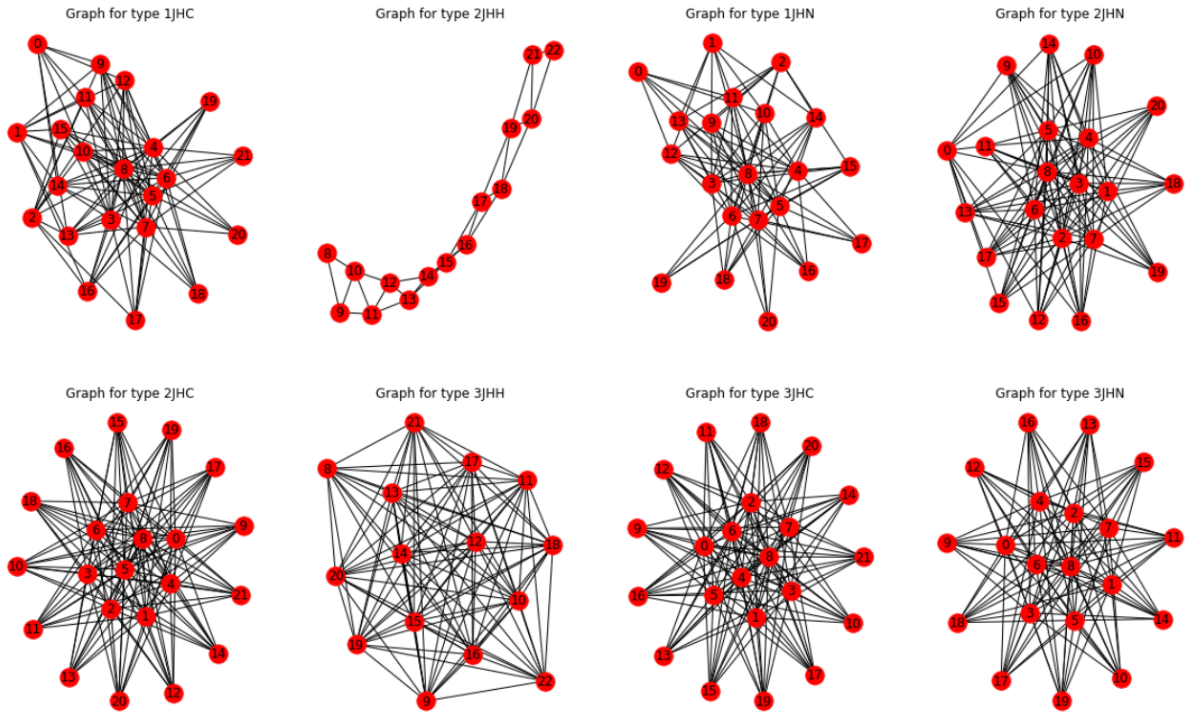


1. Distance along with the coupling Type had the impact on the scalar coupling constants
2. 1JHC, 1JHN -> had less distance and have higher coulding constants.
3. 2JHH, 2JHN, 2JHC -> have distances in range (2-8) and have negative coulings
4. 3JHC, 3JHN, 3JHH -> have very less variation in coupling constant and have higher variation in the distances between atoms described.

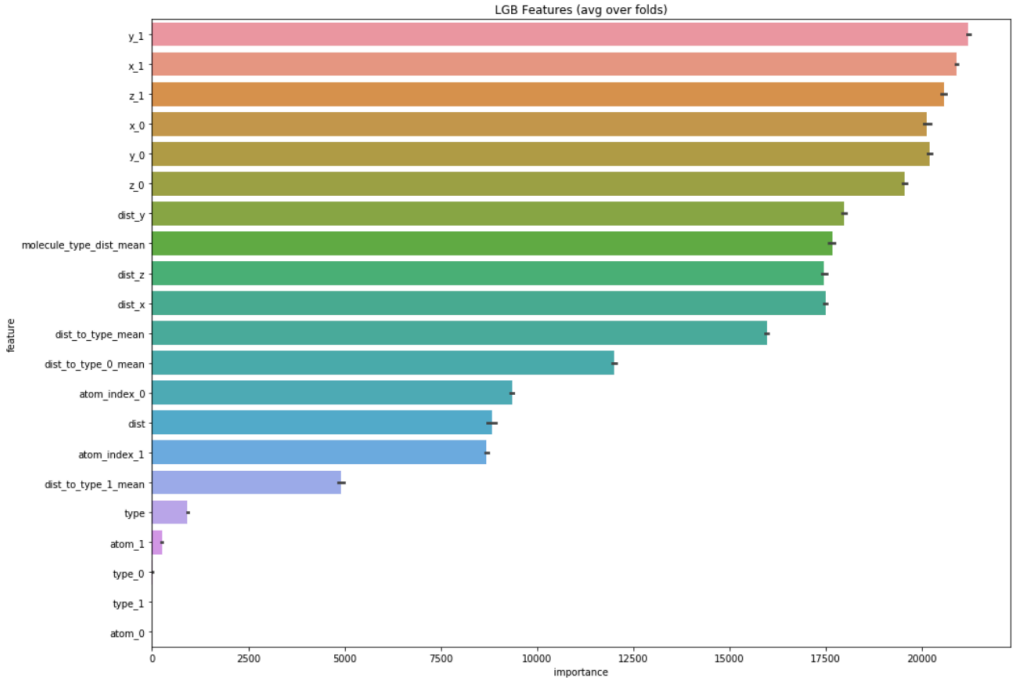
To see and understand the structure of different types and how, if they do, they affect the coupling constant, we plotted the network graph by type. We can see that atom connections have different shapes for different types. Type 2JHH has an especially unique scheme. Also we can see that some atoms are connected only to several other atoms.



But there is a little problem: as we saw earlier, there are atoms which are very rare, as a result graphs will be skewed due to them.

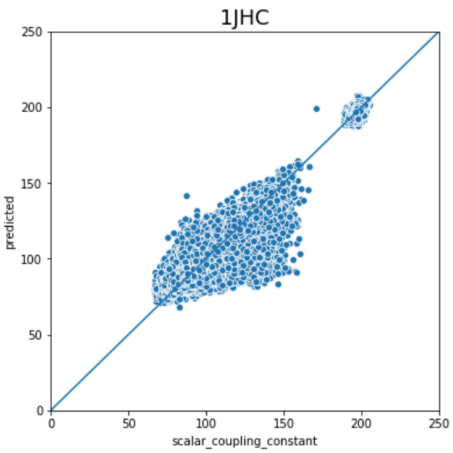


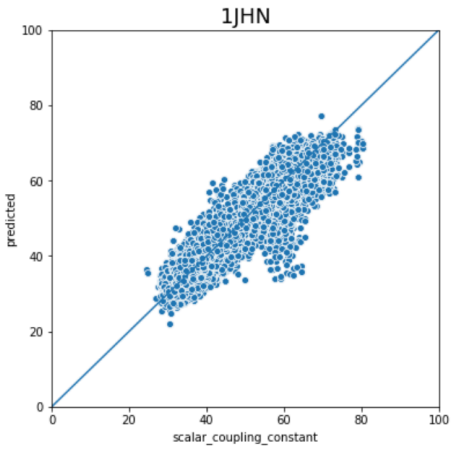
So after dropping atoms for each type which are present in less then 1% of connections, we get the above graphs. Now the graphs are much more clear.

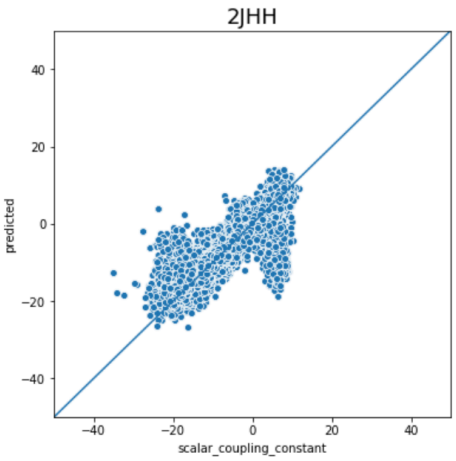


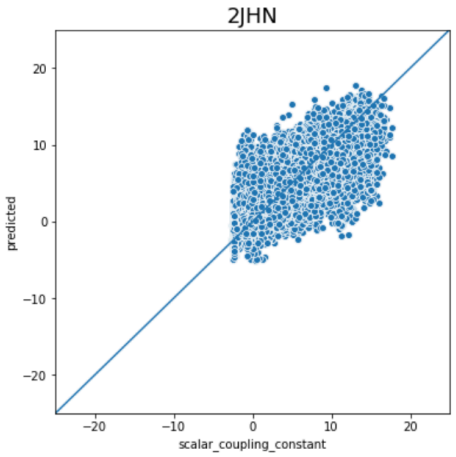
We can what features would affect our predictions the most. This would help us to to build an accurate model for predicting the scalar coupling constant.

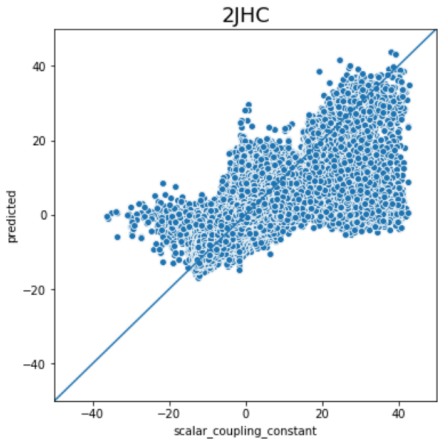
Now after building the model. We plot the predictions vs target for different types. We get the following visualisation:

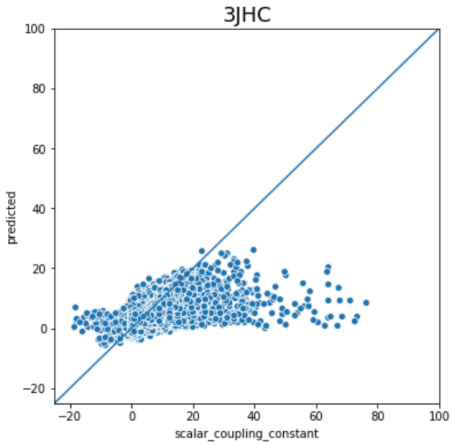


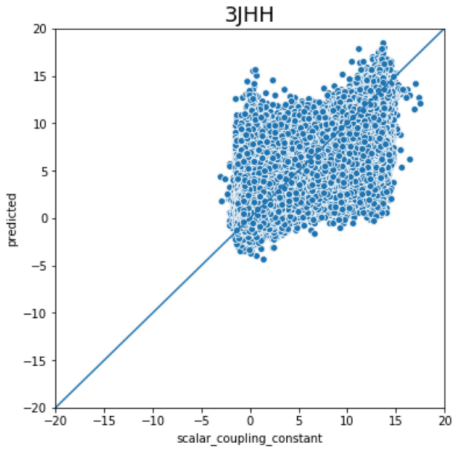


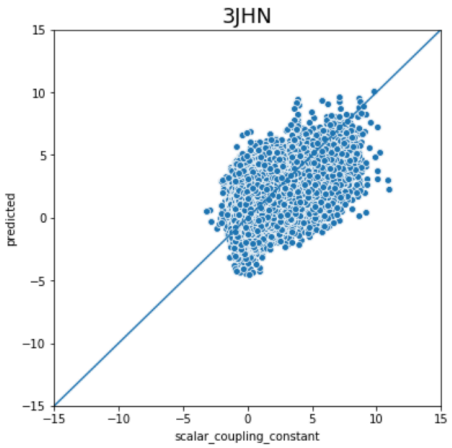












We can see that some types has better predictions than others. Like for type 1JHN it is pretty good. Whereas for 3JHC it is not.

So in the future scope of the project we can work on improving the models for these specific type to get better predictions.