stack

1. https://www.kaggle.com/filemide/distance-criskiev-hyparam-cont-1-662 (https://www.kaggle.com/filemide/distance-criskiev-hyparam-cont-1-662)
2. https://www.kaggle.com/criskiev/distance-is-all-you-need-lb-1-481 (https://www.kaggle.com/criskiev/distance-is-all-you-need-lb-1-481)
3. https://www.kaggle.com/marcelotamashiro/lgb-public-kernels-plus-more-features (https://www.kaggle.com/marcelotamashiro/lgb-public-kernels-plus-more-features)
4. https://www.kaggle.com/scaomath/no-memory-reduction-workflow-for-each-type-lb-1-28 (https://www.kaggle.com/scaomath/no-memory-reduction-workflow-for-each-type-lb-1-28)
5. https://www.kaggle.com/fnands/1-mpnn/output?scriptVersionId=18233432 (https://www.kaggle.com/fnands/1-mpnn/output?scriptVersionId=18233432)
6. https://www.kaggle.com/harshit92/fork-from-kernel-1-481 (https://www.kaggle.com/harshit92/fork-from-kernel-1-481)
7. https://www.kaggle.com/xwxw2929/keras-neural-net-and-distance-features (https://www.kaggle.com/xwxw2929/keras-neural-net-and-distance-features)
8. https://www.kaggle.com/marcogorelli/criskiev-s-distances-more-estimators-groupkfold?scriptVersionId=18843561 (https://www.kaggle.com/marcogorelli/criskiev-s-distances-more-estimators-groupkfold?scriptVersionId=18843561) 9.https://www.kaggle.com/toshik/schnet-starter-kit (https://www.kaggle.com/toshik/schnet-starter-kit)

In [1]:

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats.mstats import gmean
import seaborn as sns
%matplotlib inline
from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))
```

```
champs-scalar-coupling
models
```

In [2]:

```
os.listdir('/kaggle/input')
```

Out[2]:

```
['champs-scalar-coupling', 'models']
```

In [3]:

```
sub_path = "../input/models"
all_files = os.listdir(sub_path)
all_files
```

Out[3]:

```
['submission-1.286.csv',
 'submission-1.701.csv',
 'submission-1.327.csv',
 'submission-1.708.csv',
 'submission-1.302.csv',
 'submission-1.648.csv',
 'submission-1.782.csv',
 'submission-1.710.csv',
 'submission-1.415.csv',
 'submission-1.696.csv',
 'submission-1.662.csv',
 'submission-1.481.csv',
 'submission-1.619.csv',
 'submission-1.643.csv',
 'submission-1.672.csv',
 'submission-1.780.csv',
 'submission-1.714.csv',
 'submission-1.618.csv']
```

In [4]:

```
all_files.remove('submission-1.701.csv')
all_files.remove('submission-1.643.csv')
all_files.remove('submission-1.481.csv')
all_files.remove('submission-1.302.csv')
all_files.remove('submission-1.619.csv')
all_files.remove('submission-1.662.csv')
all_files.remove('submission-1.696.csv')
all_files.remove('submission-1.780.csv')
all_files.remove('submission-1.708.csv')
all_files.remove('submission-1.714.csv')
all_files.remove('submission-1.415.csv')
all_files.remove('submission-1.286.csv',)
```

In [5]:

```
all_files
```

Out[5]:

```
['submission-1.327.csv',
 'submission-1.648.csv',
 'submission-1.782.csv',
 'submission-1.710.csv',
 'submission-1.672.csv',
 'submission-1.618.csv']
```

In [6]:

```
import warnings
warnings.filterwarnings("ignore")
outs = [pd.read_csv(os.path.join(sub_path, f), index_col=0) for f in all_files]
concat_sub = pd.concat(outs, axis=1)
cols = list(map(lambda x: "mol" + str(x), range(len(concat_sub.columns))))
concat_sub.columns = cols
concat_sub.reset_index(inplace=True)
concat_sub.head()
ncol = concat_sub.shape[1]
```

In [7]:

```
# check correlation
concat_sub.iloc[:,1:].corr()
```

Out[7]:

	mol0	mol1	mol2	mol3	mol4	mol5
mol0	1.000000	0.999826	0.999876	0.999875	0.999812	0.999845
mol1	0.999826	1.000000	0.999898	0.999900	0.999940	0.999914
mol2	0.999876	0.999898	1.000000	0.999959	0.999880	0.999922
mol3	0.999875	0.999900	0.999959	1.000000	0.999881	0.999932
mol4	0.999812	0.999940	0.999880	0.999881	1.000000	0.999896
mol5	0.999845	0.999914	0.999922	0.999932	0.999896	1.000000

In [8]:

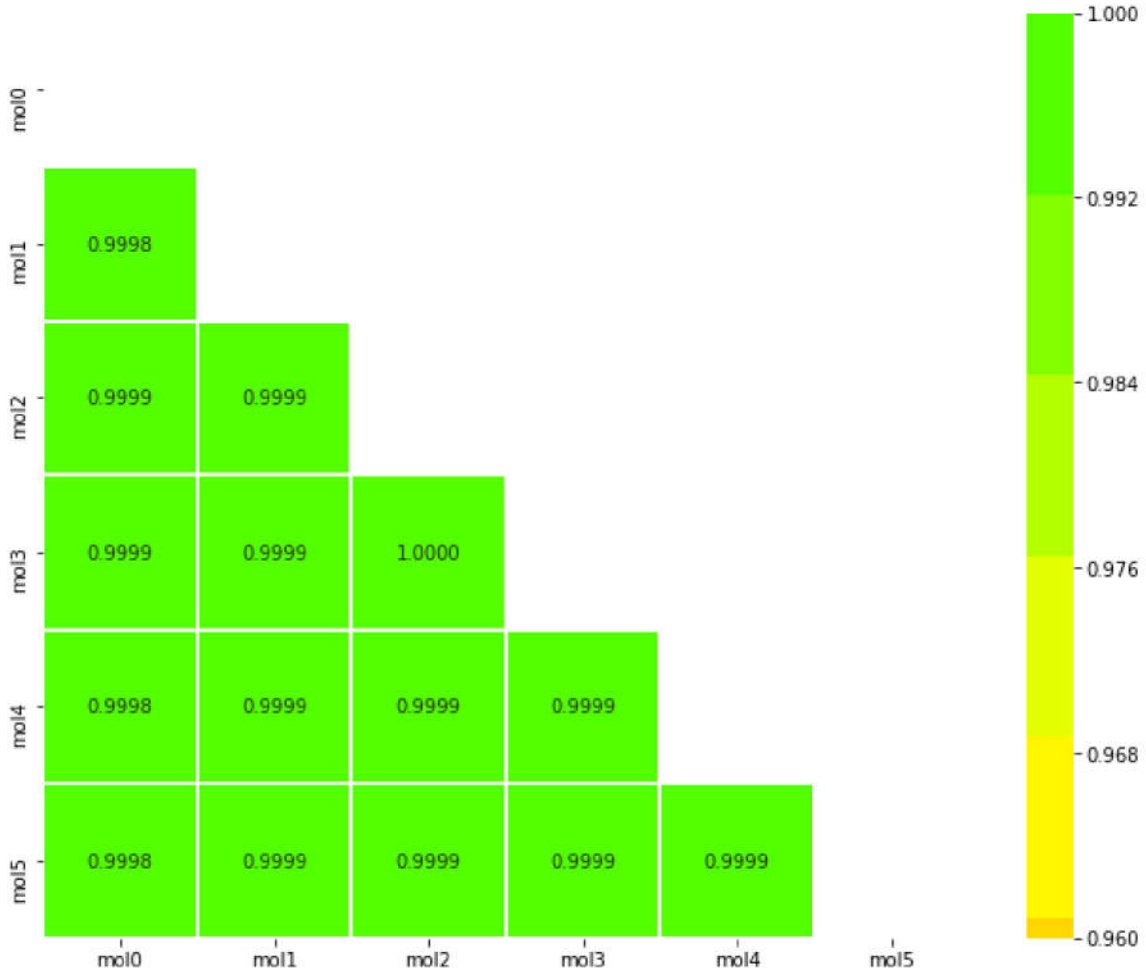
```
corr = concat_sub.iloc[:,1:].corr()
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap='prism', vmin=0.96, center=0, square=True, linewidths=1, annot
=True, fmt='.4f')
```

Out[8]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f4c39b3b9e8>



In [9]:

```
# get the data fields ready for stacking
concat_sub['m_max'] = concat_sub.iloc[:, 1:].max(axis=1)
concat_sub['m_min'] = concat_sub.iloc[:, 1:].min(axis=1)
concat_sub['m_median'] = concat_sub.iloc[:, 1:].median(axis=1)
concat_sub['m_mean']=concat_sub.iloc[:,1:].mean(axis=1)
```

In [10]:

```
concat_sub.describe()
```

Out[10]:

	id	mol0	mol1	mol2	mol3	m
count	2.505542e+06	2.505542e+06	2.505542e+06	2.505542e+06	2.505542e+06	2.505542e+
mean	5.910918e+06	1.592437e+01	1.588320e+01	1.588755e+01	1.588637e+01	1.588054e+
std	7.232878e+05	3.493039e+01	3.485671e+01	3.487233e+01	3.486492e+01	3.485870e+
min	4.658147e+06	-2.474941e+01	-3.299384e+01	-3.418129e+01	-3.307354e+01	-3.309956e+
25%	5.284532e+06	-2.278156e-01	-2.502056e-01	-2.409186e-01	-2.415684e-01	-2.461876e
50%	5.910918e+06	2.290658e+00	2.282187e+00	2.283019e+00	2.283028e+00	2.278162e+
75%	6.537303e+06	7.355250e+00	7.377225e+00	7.354648e+00	7.350433e+00	7.355285e+
max	7.163688e+06	2.057290e+02	2.085859e+02	2.031379e+02	2.030828e+02	2.844774e+

In [11]:

```
concat_sub.head(10)
```

Out[11]:

	id	mol0	mol1	mol2	mol3	mol4	mol5	m_
0	4658147	55.943756	5.418903	15.929076	17.896688	16.752722	20.500710	55.943
1	4658148	185.007780	102.854691	201.214020	152.958954	129.551773	151.843719	201.214
2	4658149	1.717238	5.910957	13.849666	10.104980	5.626666	9.891783	13.849
3	4658150	185.007780	102.854691	201.214020	150.669815	129.551773	151.843719	201.214
4	4658151	55.943756	5.418903	17.064312	18.489357	16.752722	20.500710	55.943
5	4658152	92.056030	83.566948	90.308708	91.235405	82.277916	92.437202	92.437
6	4658153	2.406619	2.133362	2.688423	3.183611	1.578010	2.753030	3.183
7	4658154	-7.956534	-9.036317	-7.543731	-7.594494	-9.356209	-7.472149	-7.472
8	4658155	-9.950280	-10.134884	-9.638301	-9.611823	-9.920478	-9.576653	-9.576
9	4658156	92.062430	83.571823	90.198280	91.430664	82.279091	92.451614	92.451

In [12]:

```
cutoff_lo = 0.8
cutoff_hi = 0.2
```

In [13]:

```
concat_sub['scalar_coupling_constant'] = concat_sub['m_mean']
concat_sub[['id', 'scalar_coupling_constant']].to_csv('stack_mean.csv',
                                       index=False, float_format='%.6f')
```

In [14]:

```
concat_sub['scalar_coupling_constant'] = concat_sub['m_median']
concat_sub[['id', 'scalar_coupling_constant']].to_csv('stack_median.csv',
                                       index=False, float_format='%.6f')
```