In [1]:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 22 11:42:18 2021

@author: stephaniewatkins
"""

import pandas as pd

#1 explore data
a9 = pd.read_table('~/Desktop/DANN862/parkinsons_updrs.data', sep=',')
a9= a9.drop(['motor_UPDRS'], axis=1)
a91 = a9.drop(['total_UPDRS'], axis=1)
a92=a9['total_UPDRS']
print(a9.index)
print(a9.columns)
print(a9.shape)
print(a9.size)
print(a9.axes)
#parameter types
a9.dtypes
#check for null values
a9.isnull().any()
#measure for asymmetry
a9.skew()
#statistc summary (mean, std, IQR)
a9.describe()
#correlation
a9.corr()
#covarience
a9.cov()
#2
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn import linear_model
from sklearn.model_selection import cross_val_score
from sklearn.metrics import mean_absolute_error
lreg = linear_model.LinearRegression()
acc1 = cross_val_score(lreg,a91, a92, cv = 10, scoring = "neg_mean_absolute_erro
print("Neg_MAE: %0.2f (+/- 0,2%f)"%(acc1.mean(),acc1.std()*2))



#3
from sklearn import tree
treem=tree.DecisionTreeRegressor()
acc2 = cross_val_score(treem, a91,a92, cv=10, scoring ="neg_mean_absolute_error"
print("Neg_MAE: %0.2f (+/- 0,2%f)"%(acc2.mean(),acc2.std()*2))


#4
from sklearn import neural_network
nn = neural_network.MLPRegressor(max_iter=10000)
acc3 = cross_val_score(nn,a91,a92,cv=10, scoring = "neg_mean_absolute_error")
print("Neg_MAE: %0.2f (+/- 0,2%f)"%(acc3.mean(),acc3.std()*2))

#5
```

```
print("the linear regression model performed the best, but it could be improved
#example of increasing cv
lreg=linear_model.LinearRegression(copy_X=True)
acc = cross_val_score(lreg,a91, a92, cv = 220, scoring = "neg_mean_absolute_erro
print("Neg_MAE: %0.2f (+/- 0,2%f)"%(acc.mean(),acc.std()*2))


#6 optimize tree model
from sklearn import neural_network
nn2 = neural_network.MLPRegressor(max_iter= 10000, activation = 'logistic')
acc4 = cross_val_score(nn2, a91, a92, cv=10, scoring = "neg_mean_absolute_error"

print("New Neural Network Neg_MAE: %0.2f (+/- 0,2%f)"%(acc4.mean(),acc4.std()*2)
```

```
RangeIndex(start=0, stop=5875, step=1)
Index(['subject#', 'age', 'sex', 'test_time', 'total_UPDRS', 'Jitter(%)',
       'Jitter(Abs)', 'Jitter:RAP', 'Jitter:PPQ5', 'Jitter:DDP', 'Shimmer',
       'Shimmer(dB)', 'Shimmer:APQ3', 'Shimmer:APQ5', 'Shimmer:APQ11',
       'Shimmer:DDA', 'NHR', 'HNR', 'RPDE', 'DFA', 'PPE'],
      dtype='object')
(5875, 21)
123375
[RangeIndex(start=0, stop=5875, step=1), Index(['subject#', 'age', 'sex', 'test_
time', 'total_UPDRS', 'Jitter(%)',
       'Jitter(Abs)', 'Jitter:RAP', 'Jitter:PPQ5', 'Jitter:DDP', 'Shimmer',
       'Shimmer(dB)', 'Shimmer:APQ3', 'Shimmer:APQ5', 'Shimmer:APQ11',
       'Shimmer:DDA', 'NHR', 'HNR', 'RPDE', 'DFA', 'PPE'],
      dtype='object')]
Neg_MAE: -8.85 (+/- 0,25.126862)
Neg_MAE: -11.09 (+/- 0,24.272721)
Neg_MAE: -9.44 (+/- 0,24.974829)
the linear regression model performed the best, but it could be improved by incr
easing the cross validation (cv=10). Additionally, another way to improve the mo
de l would be to reduce the number of attributes.
Neg_MAE: -7.80 (+/- 0,29.082110)
New Neural Network Neg_MAE: -8.41 (+/- 0,25.984840)
```