In [1]:
```
import numpy
import pandas

from sklearn.feature_selection import RFE
from sklearn.ensemble import ExtraTreesRegressor


import matplotlib.pyplot as plt
from pandas.tools.plotting import scatter_matrix


from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.metrics import explained_variance_score
from sklearn.metrics import mean_absolute_error

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
import h5py
warnings.resetwarnings()

from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.utils import np_utils
from keras.constraints import maxnorm
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler



# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
```

```
Using TensorFlow backend.
C:\Users\Sonu Agrawal\New folder\lib\importlib\_bootstrap_external.py:426: Im
portWarning: Not importing directory C:\Users\Sonu Agrawal\New folder\lib\sit
e-packages\google: missing __init__
  _warnings.warn(msg.format(portions[0]), ImportWarning)
C:\Users\Sonu Agrawal\New folder\lib\site-packages\tensorflow\python\framewor
k\ops.py:875: DeprecationWarning: builtin type EagerTensor has no __module__
attribute
  EagerTensor = c_api.TFE_Py_InitEagerTensor(_EagerTensorBase)
C:\Users\Sonu Agrawal\New folder\lib\site-packages\tensorflow\python\util\tf_
inspect.py:45: DeprecationWarning: inspect.getargspec() is deprecated, use in
spect.signature() or inspect.getfullargspec()
  if d.decorator_argspec is not None), _inspect.getargspec(target))
C:\Users\Sonu Agrawal\New folder\lib\site-packages\tensorflow\python\keras\_i
mpl\keras\backend.py:4622: ResourceWarning: unclosed file <_io.TextIOWrapper
name='C:\\Users\\Sonu Agrawal\\.keras\\keras.json' mode='r' encoding='cp125
2'>
  _config = json.load(open(_config_path))
C:\Users\Sonu Agrawal\New folder\lib\site-packages\tensorflow\python\util\tf_
inspect.py:45: DeprecationWarning: inspect.getargspec() is deprecated, use in
spect.signature() or inspect.getfullargspec()
  if d.decorator_argspec is not None), _inspect.getargspec(target))
```

In [2]:
```
# load dataset
dataframe = pandas.read_csv("F://c3.csv")



# Encode Data
#dataframe.month.replace(('jan','feb','mar','apr','may','jun','jul','aug','se
p','oct','nov','dec'),(1,2,3,4,5,6,7,8,9,10,11,12), inplace=True)
#dataframe.day.replace(('mon','tue','wed','thu','fri','sat','sun'),(1,2,3,4,5,
6,7), inplace=True)
```

In [3]:
```
print("Head:", dataframe.head())
```

```
Head:    S.No.  FFMC   DMC     DC   ISI  temp  RH  wind  rain  area clusterno
  \
0      2  90.6  35.4  669.1   6.7  18.0  33   0.9     0   0.0        c1
1      3  90.6  43.7  686.9   6.7  14.6  33   1.3     0   0.0        c1
2      6  92.3  85.3  488.0  14.7  22.2  29   5.4     0   0.0        c1
3      7  92.3  88.9  495.6   8.5  24.1  27   3.1     0   0.0        c1
4     10  92.5  88.0  698.6   7.1  22.8  40   4.0     0   0.0        c1

  $KM-K-Means  $KMD-K-Means
0   cluster-3         0.437
1   cluster-3         0.440
2   cluster-3         0.343
3   cluster-3         0.261
4   cluster-3         0.156
```

In [4]:
```
print("Correlation:", dataframe.corr(method='pearson'))
```

```
Correlation:                 S.No.      FFMC       DMC        DC       ISI
    temp  \
S.No.          1.000000  0.155539  0.409057  0.019997  0.137396  0.416801
FFMC           0.155539  1.000000  0.365022 -0.030214  0.632008  0.240935
DMC            0.409057  0.365022  1.000000  0.155381  0.266165  0.408025
DC             0.019997 -0.030214  0.155381  1.000000 -0.155508 -0.208631
ISI            0.137396  0.632008  0.266165 -0.155508  1.000000  0.267841
temp           0.416801  0.240935  0.408025 -0.208631  0.267841  1.000000
RH            -0.024650 -0.165688 -0.112914 -0.048308 -0.123695 -0.538731
wind          -0.037655  0.093488  0.005184 -0.167575  0.293632  0.098322
rain                NaN       NaN       NaN       NaN       NaN       NaN
area           0.084233  0.023150  0.108358  0.004618 -0.014489  0.074415
$KMD-K-Means   0.235579 -0.005095  0.116807 -0.280684  0.130772  0.215162

                     RH      wind  rain      area  $KMD-K-Means
S.No.         -0.024650 -0.037655   NaN  0.084233      0.235579
FFMC          -0.165688  0.093488   NaN  0.023150     -0.005095
DMC           -0.112914  0.005184   NaN  0.108358      0.116807
DC            -0.048308 -0.167575   NaN  0.004618     -0.280684
ISI           -0.123695  0.293632   NaN -0.014489      0.130772
temp          -0.538731  0.098322   NaN  0.074415      0.215162
RH             1.000000 -0.176228   NaN -0.063520     -0.079644
wind          -0.176228  1.000000   NaN  0.063225      0.094649
rain                NaN       NaN   NaN       NaN           NaN
area          -0.063520  0.063225   NaN  1.000000      0.462157
$KMD-K-Means  -0.079644  0.094649   NaN  0.462157      1.000000
```

In [5]:
```
dataset = dataframe.values


X = dataset[:,0:9]
Y = dataset[:,9]
print(dataset)
print(X)
print(Y)
```

```
[[2 90.6 35.4 ... 'c1' 'cluster-3' 0.43700000000000006]
 [3 90.6 43.7 ... 'c1' 'cluster-3' 0.44]
 [6 92.3 85.3 ... 'c1' 'cluster-3' 0.34299999999999997]
 ...
 [512 81.6 56.7 ... 'c1' 'cluster-3' 0.36]
 [513 81.6 56.7 ... 'c1' 'cluster-3' 0.364]
 [516 94.4 146.0 ... 'c1' 'cluster-3' 0.177]]
[[2 90.6 35.4 ... 33 0.9 0]
 [3 90.6 43.7 ... 33 1.3 0]
 [6 92.3 85.3 ... 29 5.4 0]
 ...
 [512 81.6 56.7 ... 35 2.7 0]
 [513 81.6 56.7 ... 32 2.7 0]
 [516 94.4 146.0 ... 42 4.0 0]]
[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.43 0.47 0.55 0.77 0.9 0.96 1.12 1.19 1.36 1.43
 1.46 1.46 1.56 1.61 1.63 1.64 1.69 1.9 1.95 2.29 2.51 2.55 2.57 2.69 2.74
 3.07 3.5 4.53 4.69 4.88 5.33 5.44 6.83 6.96 7.04 7.3 8.68 8.71 9.41 10.01
 10.02 11.06 11.32 11.53 13.7 14.57 15.45 17.2 19.23 23.41 24.23 26.13
 29.48 31.72 35.88 48.55 49.37 58.3 64.1 71.3 88.49 95.18 103.39 105.66
 154.88 196.48 200.94 212.88 1090.84 0.76 0.09 0.75 0.0 2.47 0.68 0.24
 0.21 1.52 10.34 0.0 8.02 0.68 0.0 1.38 8.85 4.25 1.56 6.54 0.0 0.0 4.4
 0.52 3.09 0.0 0.0 0.0 0.0 0.0 8.0 2.64 86.45 0.0 0.0 0.0 5.18 0.0 0.0 0.0
 14.29 0.0 1.58 0.0 0.0 3.78 0.0 4.41 34.36 7.21 1.01 2.18 4.42 0.0 0.0
 0.0 0.0 3.33 6.58 15.64 11.22 2.13 0.0 56.04 7.48 1.47 3.93 0.0 0.0 5.83
 28.19 2.03 1.72 5.97 13.06 1.26 0.0 8.12 1.09 3.94 2.93 5.65 20.03 1.75
 0.0 12.64 0.0 174.63 7.73 5.86 16.0 28.74 0.0 30.18 3.63 0.0 0.0 8.16
 4.95 0.0 0.0 6.04 0.0 0.0 1.63 0.0 746.28 7.02 2.44 3.05 185.76 0.0 6.3
 0.72 4.96 2.35 0.0 3.2 6.36 0.0 0.54 0.33 1.23 0.0 10.08 1.76 7.36 278.53
 2.75 0.0 1.29 0.0 26.43 2.07 2.0 16.4 0.0 0.0 0.0 0.0 43.32 8.59 0.0 2.77
 14.68 40.54 1.95 49.59 5.8 0.0 0.0 0.0 6.44 0.0]
```

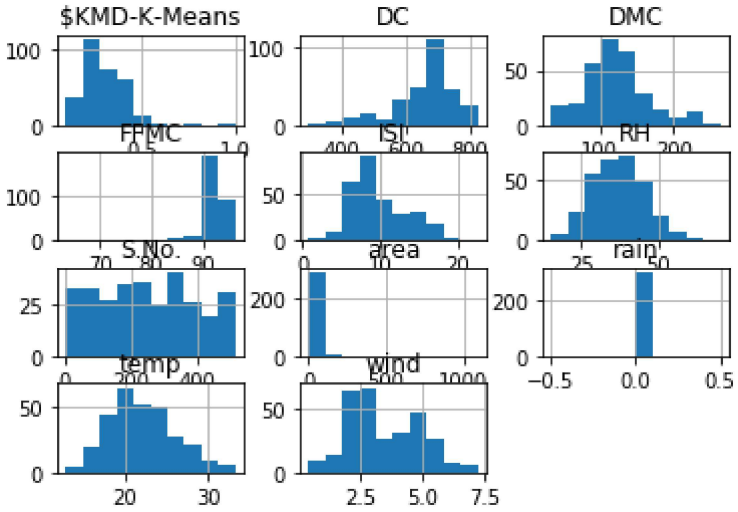
In [6]:
```
#Feature Selection
model = ExtraTreesRegressor()
rfe = RFE(model, 3)
fit = rfe.fit(X, Y)

print("Number of Features: ", fit.n_features_)
print("Selected Features: ", fit.support_)
print("Feature Ranking: ", fit.ranking_)
```

```
Number of Features:  3
Selected Features:  [ True False  True False False False False  True False]
Feature Ranking:  [1 6 1 4 5 2 3 1 7]
```

In [7]:
```
dataframe.hist()
```

Out[7]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F1011D0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F4366A0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F4705C0
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F495438>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F4DB5C0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F4DB5F8
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F54B860>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F586860>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F5B9080
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F5F3080>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F6294E0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000025F7F662A20
>]],
      dtype=object)
```



In [8]:
```
num_instances = len(X)

models = []
models.append(('LiR', LinearRegression()))
models.append(('Ridge', Ridge()))
models.append(('Lasso', Lasso()))
models.append(('ElasticNet', ElasticNet()))
models.append(('Bag_Re', BaggingRegressor()))
models.append(('RandomForest', RandomForestRegressor()))
models.append(('ExtraTreesRegressor', ExtraTreesRegressor()))
models.append(('KNN', KNeighborsRegressor()))
models.append(('CART', DecisionTreeRegressor()))
models.append(('SVM', SVR()))

# Evaluations
results = []
names = []
scoring = []

for name, model in models:
    # Fit the model
    model.fit(X, Y)

    predictions = model.predict(X)

    # Evaluate the model
    score = explained_variance_score(Y, predictions)
    mae = mean_absolute_error(predictions, Y)
    # print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
    results.append(mae)
    names.append(name)

    msg = "%s: %f (%f)" % (name, score, mae)
    print(msg)
```

```
LiR: 0.025218 (26.402451)
Ridge: 0.025218 (26.399858)
Lasso: 0.024996 (26.201808)
ElasticNet: 0.024689 (26.094608)
Bag_Re: 0.926109 (6.869710)
RandomForest: 0.791795 (8.508113)
ExtraTreesRegressor: 1.000000 (0.000000)
KNN: 0.375183 (17.707787)
CART: 1.000000 (0.000000)
SVM: 0.005014 (16.398515)
```

In [9]:
```
from keras.layers import Input, Dense
from keras.models import Model
from keras.optimizers import Adam

def build_model():

    fire_in = Input((9,))

    dense = Dense(9, activation="sigmoid")(fire_in)
    dense = Dense(1,  activation="linear")(dense)

    model = Model(inputs=fire_in, outputs=dense)

    adam = Adam(lr=0.01)

    model.compile(optimizer=adam, loss="mse")

    return model
```

In [ ]:
```

```

In [10]:
```
Y = numpy.array(Y).reshape((len(Y), 1))
#Y.reshape(-1, 1)
print(Y)
# normalize the dataset
scaler = MinMaxScaler(feature_range=(0, 1))
Y = scaler.fit_transform(Y)
print(Y)
```

```
[[0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
 [0.0]
```

```
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[0.43]
[0.47]
[0.55]
[0.77]
[0.9]
[0.96]
[1.12]
[1.19]
[1.36]
[1.43]
[1.46]
[1.46]
[1.56]
[1.61]
[1.63]
[1.64]
[1.69]
[1.9]
[1.95]
[2.29]
[2.51]
[2.55]
[2.57]
[2.69]
[2.74]
[3.07]
[3.5]
[4.53]
[4.69]
[4.88]
[5.33]
[5.44]
[6.83]
[6.96]
[7.04]
[7.3]
```

```
[8.68]
[8.71]
[9.41]
[10.01]
[10.02]
[11.06]
[11.32]
[11.53]
[13.7]
[14.57]
[15.45]
[17.2]
[19.23]
[23.41]
[24.23]
[26.13]
[29.48]
[31.72]
[35.88]
[48.55]
[49.37]
[58.3]
[64.1]
[71.3]
[88.49]
[95.18]
[103.39]
[105.66]
[154.88]
[196.48]
[200.94]
[212.88]
[1090.84]
[0.76]
[0.09]
[0.75]
[0.0]
[2.47]
[0.68]
[0.24]
[0.21]
[1.52]
[10.34]
[0.0]
[8.02]
[0.68]
[0.0]
[1.38]
[8.85]
[4.25]
[1.56]
[6.54]
[0.0]
[0.0]
[4.4]
[0.52]
[3.09]
```

```
[0.0]
[0.0]
[0.0]
[0.0]
[0.0]
[8.0]
[2.64]
[86.45]
[0.0]
[0.0]
[0.0]
[5.18]
[0.0]
[0.0]
[0.0]
[14.29]
[0.0]
[1.58]
[0.0]
[0.0]
[3.78]
[0.0]
[4.41]
[34.36]
[7.21]
[1.01]
[2.18]
[4.42]
[0.0]
[0.0]
[0.0]
[0.0]
[3.33]
[6.58]
[15.64]
[11.22]
[2.13]
[0.0]
[56.04]
[7.48]
[1.47]
[3.93]
[0.0]
[0.0]
[5.83]
[28.19]
[2.03]
[1.72]
[5.97]
[13.06]
[1.26]
[0.0]
[8.12]
[1.09]
[3.94]
[2.93]
[5.65]
```

```
[20.03]
[1.75]
[0.0]
[12.64]
[0.0]
[174.63]
[7.73]
[5.86]
[16.0]
[28.74]
[0.0]
[30.18]
[3.63]
[0.0]
[0.0]
[8.16]
[4.95]
[0.0]
[0.0]
[6.04]
[0.0]
[0.0]
[1.63]
[0.0]
[746.28]
[7.02]
[2.44]
[3.05]
[185.76]
[0.0]
[6.3]
[0.72]
[4.96]
[2.35]
[0.0]
[3.2]
[6.36]
[0.0]
[0.54]
[0.33]
[1.23]
[0.0]
[10.08]
[1.76]
[7.36]
[278.53]
[2.75]
[0.0]
[1.29]
[0.0]
[26.43]
[2.07]
[2.0]
[16.4]
[0.0]
[0.0]
[0.0]
```

```
 [0.0]
 [43.32]
 [8.59]
 [0.0]
 [2.77]
 [14.68]
 [40.54]
 [1.95]
 [49.59]
 [5.8]
 [0.0]
 [0.0]
 [0.0]
 [6.44]
 [0.0]]
[[0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
```

```
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[3.94191632e-04]
[4.30860621e-04]
[5.04198599e-04]
[7.05878039e-04]
[8.25052253e-04]
[8.80055737e-04]
[1.02673169e-03]
[1.09090242e-03]
[1.24674563e-03]
[1.31091636e-03]
[1.33841810e-03]
[1.33841810e-03]
[1.43009057e-03]
[1.47592681e-03]
[1.49426130e-03]
[1.50342855e-03]
[1.54926479e-03]
[1.74177698e-03]
[1.78761322e-03]
[2.09929962e-03]
[2.30097906e-03]
```

```
[2.33764805e-03]
[2.35598255e-03]
[2.46598951e-03]
[2.51182575e-03]
[2.81434491e-03]
[3.20853654e-03]
[4.15276301e-03]
[4.29943896e-03]
[4.47361666e-03]
[4.88614279e-03]
[4.98698251e-03]
[6.26122988e-03]
[6.38040409e-03]
[6.45374207e-03]
[6.69209050e-03]
[7.95717062e-03]
[7.98467236e-03]
[8.62637967e-03]
[9.17641451e-03]
[9.18558175e-03]
[1.01389755e-02]
[1.03773239e-02]
[1.05698361e-02]
[1.25591287e-02]
[1.33566793e-02]
[1.41633970e-02]
[1.57676653e-02]
[1.76286165e-02]
[2.14605258e-02]
[2.22122401e-02]
[2.39540171e-02]
[2.70250449e-02]
[2.90785083e-02]
[3.28920832e-02]
[4.45069854e-02]
[4.52586997e-02]
[5.34450515e-02]
[5.87620549e-02]
[6.53624730e-02]
[8.11209710e-02]
[8.72538594e-02]
[9.47801694e-02]
[9.68611345e-02]
[1.41982326e-01]
[1.80118074e-01]
[1.84206666e-01]
[1.95152360e-01]
[1.00000000e+00]
[6.96710792e-04]
[8.25052253e-05]
[6.87543544e-04]
[0.00000000e+00]
[2.26431007e-03]
[6.23372814e-04]
[2.20013934e-04]
[1.92512192e-04]
[1.39342158e-03]
```

```
[9.47893367e-03]
[0.00000000e+00]
[7.35213230e-03]
[6.23372814e-04]
[0.00000000e+00]
[1.26508012e-03]
[8.11301382e-03]
[3.89608009e-03]
[1.43009057e-03]
[5.99537971e-03]
[0.00000000e+00]
[0.00000000e+00]
[4.03358879e-03]
[4.76696857e-04]
[2.83267940e-03]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[7.33379781e-03]
[2.42015328e-03]
[7.92508526e-02]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[4.74863408e-03]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[1.30999963e-02]
[0.00000000e+00]
[1.44842507e-03]
[0.00000000e+00]
[0.00000000e+00]
[3.46521946e-03]
[0.00000000e+00]
[4.04275604e-03]
[3.14986616e-02]
[6.60958527e-03]
[9.25891973e-04]
[1.99845990e-03]
[4.05192329e-03]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[0.00000000e+00]
[3.05269334e-03]
[6.03204870e-03]
[1.43375747e-02]
[1.02856514e-02]
[1.95262367e-03]
[0.00000000e+00]
[5.13732536e-02]
[6.85710095e-03]
[1.34758535e-03]
[3.60272817e-03]
```

```
[0.00000000e+00]
[0.00000000e+00]
[5.34450515e-03]
[2.58424700e-02]
[1.86095119e-03]
[1.57676653e-03]
[5.47284661e-03]
[1.19724249e-02]
[1.15507315e-03]
[0.00000000e+00]
[7.44380477e-03]
[9.99229951e-04]
[3.61189542e-03]
[2.68600345e-03]
[5.17949470e-03]
[1.83619963e-02]
[1.60426827e-03]
[0.00000000e+00]
[1.15874005e-02]
[0.00000000e+00]
[1.60087639e-01]
[7.08628213e-03]
[5.37200689e-03]
[1.46675956e-02]
[2.63466686e-02]
[0.00000000e+00]
[2.76667522e-02]
[3.32771076e-03]
[0.00000000e+00]
[0.00000000e+00]
[7.48047376e-03]
[4.53778739e-03]
[0.00000000e+00]
[0.00000000e+00]
[5.53701734e-03]
[0.00000000e+00]
[0.00000000e+00]
[1.49426130e-03]
[0.00000000e+00]
[6.84133328e-01]
[6.43540758e-03]
[2.23680833e-03]
[2.79601041e-03]
[1.70290785e-01]
[0.00000000e+00]
[5.77536577e-03]
[6.60041803e-04]
[4.54695464e-03]
[2.15430311e-03]
[0.00000000e+00]
[2.93351912e-03]
[5.83036926e-03]
[0.00000000e+00]
[4.95031352e-04]
[3.02519160e-04]
[1.12757141e-03]
[0.00000000e+00]
```

```
 [9.24058524e-03]
 [1.61343552e-03]
 [6.74709398e-03]
 [2.55335338e-01]
 [2.52099300e-03]
 [0.00000000e+00]
 [1.18257490e-03]
 [0.00000000e+00]
 [2.42290345e-02]
 [1.89762018e-03]
 [1.83344945e-03]
 [1.50342855e-02]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [3.97125151e-02]
 [7.87466540e-03]
 [0.00000000e+00]
 [2.53932749e-03]
 [1.34575190e-02]
 [3.71640204e-02]
 [1.78761322e-03]
 [4.54603792e-02]
 [5.31700341e-03]
 [0.00000000e+00]
 [0.00000000e+00]
 [0.00000000e+00]
 [5.90370723e-03]
 [0.00000000e+00]]
```

```
C:\Users\Sonu Agrawal\New folder\lib\site-packages\sklearn\utils\validation.p
y:475: DataConversionWarning: Data with input dtype object was converted to f
loat64 by MinMaxScaler.
  warnings.warn(msg, DataConversionWarning)
```

In [11]:
```
# define base model
def baseline_model():
    # create model
    model = Sequential()
    model.add(Dense(9, input_dim=9, kernel_initializer='normal', activation='r
elu'))
    model.add(Dense(1, kernel_initializer='normal'))

    # compile model
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model


# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)

# evaluate model with standardized dataset
estimator = KerasRegressor(build_fn=baseline_model, nb_epoch=600, batch_size=5
, verbose=0)

kfold = KFold(n_splits=30, random_state=seed)
results = cross_val_score(estimator, X, Y, cv=kfold)
print("Results: %.2f (%.2f) MSE",results)
print("Results: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

```
C:\Users\Sonu Agrawal\New folder\lib\site-packages\tensorflow\python\util\tf_
inspect.py:45: DeprecationWarning: inspect.getargspec() is deprecated, use in
spect.signature() or inspect.getfullargspec()
  if d.decorator_argspec is not None), _inspect.getargspec(target))

Results: %.2f (%.2f) MSE [-3.61708831e-02 -5.04039311e+00 -7.20032491e-03 -1.
22529264e-02
 -1.89613353e-03 -3.09316725e-01 -3.89410667e-02 -1.39303893e+00
 -1.81725709e-02 -1.14551922e-02 -5.00464272e-02 -8.52823257e-02
 -2.39250466e-03 -8.08399485e-03 -1.19730124e-01 -4.58708273e-05
 -6.49725925e-03 -1.10892588e-02 -1.30242221e-02 -2.37135785e-02
 -9.69509431e-03 -1.09332796e-01 -6.40651211e-03 -7.04329764e-03
 -3.46980151e-02 -1.13186747e-01 -3.91147975e-02 -2.57082088e-02
 -7.66131766e-02 -6.57925173e-03]
Results: -0.25 (0.92) MSE
```

In [12]:
```
from sklearn.model_selection import train_test_split
```

In [13]:
```
train_x,test_x,train_y,test_y = train_test_split(X,Y)
```

In [14]:
```
print(train_x.shape,train_y.shape,test_x.shape,test_y.shape)
```

```
(225, 9) (225, 1) (75, 9) (75, 1)
```

In [15]:
```
model.fit(train_x,train_y)
```

```
C:\Users\Sonu Agrawal\New folder\lib\site-packages\sklearn\utils\validation.p
y:578: DataConversionWarning: A column-vector y was passed when a 1d array wa
s expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

Out[15]:
```
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='auto',
  kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

In [16]:
```
res =model.predict(test_x)
from sklearn.metrics import accuracy_score
```

In [17]:
```
mean_squared_error(test_y,res)
```

Out[17]:
```
0.010745646664570686
```