krishnaik06 / Develop-Multi-Output-Regression-Models Public

Code Issues Pull requests Actions Projects Security Insights

master

Develop-Multi-Output-Regression-Models / Untitled1.ipynb

krishnaik06 Add files via upload

1 contributor

```
431 lines (431 sloc) | 13.9 KB
```

MultiOutput Regression Models In Python

In [51]:

```
##https://machinelearningmastery.com/multi-output-regression-models-with-p
from sklearn.datasets import make_regression
```

In [24]:

```
## Create the dataset
X,y =make_regression(n_samples=1500,n_informative=5,n_features=10,n_targe
```

In [25]:

```
X
```

Out[25]:

```
array([[ 0.69907569,  0.32187966,  0.61588344, ..., -0.61671409,
        -1.32796987,  0.44671079],
       [-0.20861179, -0.51124433, -0.57848941, ...,  0.15557089,
        -1.59989122,  1.09469799],
       [-0.15920883,  0.25933781, -1.2751625 , ...,  1.19685677,
         0.20902432, -1.14243785],
       ...,
       [ 0.06488138,  0.61067528, -1.39884742, ..., -1.46799878,
         0.21239277, -0.01979268],
       [ 0.64787586, -0.93727554,  0.75609481, ...,  1.30296143,
        -0.02829069, -0.96125378],
       [-1.23811447, -0.3771934 ,  0.39785007, ...,  0.18800362,
         0.01558403, -0.14071533]])
```

In [26]:

```
y
```

Out[26]:

```
array([[ -10.84499368,   35.41474598],
       [ -72.03787795,  -83.27541199],
       [ 134.13556611,  -60.5052339 ],
       ...,
       [  34.69065243,  -36.78356982],
       [-191.16828249,  -40.78592399],
       [  10.95807384,    8.03794533]])
```

Apply Linear Regression

In [44]:

```
from sklearn.linear_model import LinearRegression
```

In [45]:

```
lrregression=LinearRegression()
lrregression.fit(X,y)
```

Out[45]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=F
alse)
```

In [46]:

```
test_data=[[-0.35383149,  0.39382202, -2.03033197,  0.08873402, -0.385765
        0.0032707 , -0.56476034, -0.67236167,  0.31317233,  1.5208706 ]]
```

In [47]:

```
lrregression.predict(test_data)
```

Out[47]:

```
array([[ -33.3323708 , -111.26522141]])
```

Decision Tree Regression

In [33]:
```
from sklearn.tree import DecisionTreeRegressor
dtregressor=DecisionTreeRegressor()
dtregressor.fit(X,y)
```

Out[33]:
```
DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None,
                      max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort='deprecated',
                      random_state=None, splitter='best')
```

In [34]:
```
dtregressor.predict(test_data)
```

Out[34]:
```
array([[ -43.98022752, -160.17506145]])
```

Random Forest Regressor

In [35]:
```
from sklearn.ensemble import RandomForestRegressor
rdregressor=RandomForestRegressor()
rdregressor.fit(X,y)
```

Out[35]:
```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=
None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

In [36]:
```
rdregressor.predict(test_data)
```

Out[36]:
```
array([[ -18.44775161, -103.10376047]])
```

CrossValidation

In [37]:
```
from sklearn.model_selection import cross_val_score
```

In [38]:
```
scores=cross_val_score(rdregressor,X,y,scoring='neg_mean_squared_error',cv
```

In [39]:
```
print(scores)
```

```
[-708.82256827 -852.86677926 -716.82390539 -678.08208514 -532.81215354]
```

All Algorithms does not support Multioutput Regressor

In [48]:
```
from sklearn.svm import LinearSVR
svregressor=LinearSVR()
```

```
svregressor.fit(X,y)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-48-66a85235e0ef> in <module>
      1 from sklearn.svm import LinearSVR
      2 svregressor=LinearSVR()
----> 3 svregressor.fit(X,y)

c:\users\krish naik\anaconda3\envs\nlp\lib\site-packages\sklearn\svm\_clas
ses.py in fit(self, X, y, sample_weight)
    424         X, y = check_X_y(X, y, accept_sparse='csr',
    425                          dtype=np.float64, order="C",
--> 426                          accept_large_sparse=False)
    427         penalty = 'l2'  # SVR only accepts l2 penalty
    428         self.coef_, self.intercept_, self.n_iter_ = _fit_liblinear
(

c:\users\krish naik\anaconda3\envs\nlp\lib\site-packages\sklearn\utils\val
idation.py in check_X_y(X, y, accept_sparse, accept_large_sparse, dtype, o
rder, copy, force_all_finite, ensure_2d, allow_nd, multi_output, ensure_mi
n_samples, ensure_min_features, y_numeric, warn_on_dtype, estimator)
    742                         dtype=None)
    743     else:
--> 744         y = column_or_1d(y, warn=True)
    745         _assert_all_finite(y)
    746     if y_numeric and y.dtype.kind == 'O':

c:\users\krish naik\anaconda3\envs\nlp\lib\site-packages\sklearn\utils\val
idation.py in column_or_1d(y, warn)
    779         return np.ravel(y)
    780
--> 781     raise ValueError("bad input shape {0}".format(shape))
    782
    783

ValueError: bad input shape (1500, 2)
```

MultiOutput Regressor sklearn

In [40]:

```
from sklearn.multioutput import MultiOutputRegressor
```

In [49]:

```
mulregressor=MultiOutputRegressor(svregressor)
mulregressor.fit(X,y)
```

Out[49]:

```
MultiOutputRegressor(estimator=LinearSVR(C=1.0, dual=True, epsilon=0.0,
                                         fit_intercept=True,
                                         intercept_scaling=1.0,
                                         loss='epsilon_insensitive',
                                         max_iter=1000, random_state=None,
                                         tol=0.0001, verbose=0),
                     n_jobs=None)
```

In [50]: