

```
In [1]: # LASSO Regression
import pandas as pd
import numpy as np
#Load Libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso
from sklearn.feature_selection import SelectFromModel
from sklearn.preprocessing import StandardScaler
```

```
In [2]: # Load dataset
data = pd.read_csv('face_behave.csv')
data = data.dropna()
data.shape
```

Out[2]: (2955, 43)

```
In [3]: x = pd.DataFrame(data.drop(labels=['PP','Condition'], axis=1))
y = pd.DataFrame(data['Condition'])
```

```
In [4]: #Scaling and Splitting the data set
from sklearn.preprocessing import MinMaxScaler
Min_Max = MinMaxScaler()
X = Min_Max.fit_transform(x)
Y = Min_Max.fit_transform(y)
# Split the data into 30% test and 70% training
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_s
X_train.shape, X_test.shape
```

Out[4]: ((2068, 41), (887, 41))

```
In [5]: #Selecting features using Lasso regularization using SelectFromModel
from sklearn.linear_model import Lasso, LogisticRegression
from sklearn.feature_selection import SelectFromModel
sel_ = SelectFromModel(LogisticRegression(C=0.5, penalty='l1', solver='liblinear')
sel_.fit(X_train, np.ravel(Y_train,order='C'))
sel_.get_support()
X_train = pd.DataFrame(X_train)
```

```
In [6]: ##To See Selected set of features
selected_feat = X_train.columns[(sel_.get_support())]
print('total features: {}'.format((X_train.shape[1])))
print('selected features: {}'.format(len(selected_feat)))
print('features with coefficients rank to zero: {}'.format(
    np.sum(sel_.estimator_.coef_ == 0)))
```

```
total features: 41
selected features: 22
features with coefficients rank to zero: 19
```

```
In [7]: np.sum(sel_.estimator_.coef_ == 0)

removed_feats = X_train.columns[(sel_.estimator_.coef_ == 0).ravel().tolist()]
removed_feats
```

```
Out[7]: Int64Index([0, 3, 4, 5, 6, 7, 8, 14, 16, 20, 21, 22, 24, 26, 28, 29, 30, 34,
                  37],
                  dtype='int64')
```

```
In [8]: removed_feats = X_train.columns[(sel_.estimator_.coef_ == 0).ravel().tolist()]
removed_feats
X_train_selected = sel_.transform(X_train)
X_test_selected = sel_.transform(X_test)
X_train_selected.shape, X_test_selected.shape
```

```
Out[8]: ((2068, 22), (887, 22))
```

```
In [14]: selected_feat.sort_values()
```

```
Out[14]: Int64Index([ 1,  2,  9, 10, 11, 12, 13, 15, 17, 18, 19, 23, 25, 27, 31, 32, 33,
                    35, 36, 38, 39, 40],
                    dtype='int64')
```

```
In [10]: from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
# Create a random forest classifier
clf = RandomForestClassifier(n_estimators=10000, random_state=0, n_jobs=-1)
# Train the classifier
clf.fit(X_train_selected, np.ravel(Y_train, order='C'))
# Apply The Full Featured Classifier To The Test Data
y_pred = clf.predict(X_test_selected)
# View The Accuracy Of Our Selected Feature Model
accuracy_score(Y_test, y_pred)
```

```
Out[10]: 0.8060879368658399
```