

neural_network

January 30, 2019

```
In [1]: import numpy
        from sklearn.utils import shuffle
        from sklearn.model_selection import train_test_split
        X = numpy.loadtxt("./data/Train/X_train.txt")
        y = numpy.loadtxt("./data/Train/y_train.txt")

In [5]: X, y = shuffle(X, y)
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

In [5]: from models import neural_network
        from sklearn.metrics import accuracy_score
        ls = [5, 10, 50, 100, 561]
        report = {}

In [6]: from sklearn.feature_selection import SelectKBest, mutual_info_classif
        report['mutual_info_classif'] = []
        for l in ls:
            transformer = SelectKBest(mutual_info_classif, k=1)
            X_train_new = transformer.fit_transform(X_train, y_train)
            neural_network_model = neural_network.without_penalty(X_train_new, y_train)
            y_pred = neural_network_model.predict(transformer.transform(X_test))
            score = accuracy_score(y_test, y_pred)
            data={
                'l': l,
                'score': score
            }
            report['mutual_info_classif'].append(data)

In [8]: report['mutual_info_classif']

Out[8]: [{'l': 5, 'score': 0.5688545688545689},
          {'l': 10, 'score': 0.5611325611325612},
          {'l': 50, 'score': 0.8648648648648649},
          {'l': 100, 'score': 0.8803088803088803},
          {'l': 561, 'score': 0.9691119691119691}]

In [9]: from sklearn.feature_selection import SelectFromModel
        from models import svm
```

```

clf = svm.penalty_l1(X_train, y_train, 0.01)
report['sfm_svml1_0.01'] = []
for l in ls:
    transformer = SelectFromModel(clf, prefit=True, max_features=l)
    X_train_new = transformer.transform(X_train)
    neural_network_model = neural_network.without_penalty(X_train_new, y_train)
    y_pred = neural_network_model.predict(transformer.transform(X_test))
    score = accuracy_score(y_test, y_pred)
    data={
        'l': l,
        'score': score
    }
    report['sfm_svml1_0.01'].append(data)

```

/home/mahdi/.local/share/virtualenvs/machine-learning-final-project-uk2p9d2v/lib/python3.6/site-
 "the number of iterations.", ConvergenceWarning)

In [10]: report['sfm_svml1_0.01']

```

Out[10]: [{ 'l': 5, 'score': 0.7541827541827542},
           { 'l': 10, 'score': 0.9034749034749034},
           { 'l': 50, 'score': 0.9510939510939511},
           { 'l': 100, 'score': 0.9633204633204633},
           { 'l': 561, 'score': 0.9633204633204633}]

```