Easy - TPOT Machine Learning

linkedin.com/pulse/easy-tpot-machine-learning-moshe-beeri

Today as I am looking for new position I have been asked about my python skills and remembered that 8 month ago I have been playing with TPOT for machine learning automation, I've been following TPOT for three years already, and I finally got the time to play with it.

So I did, I came across interesting dataset I could use, and I started with a classifier and then a regression, while I was considering the result, the regression technique got to 96.5% while the classification got 98.9% !!! I was wondering is I can put it all in a one pager!

No more machine learning in a nut shell, but real effective machine learning ONE PAGER.

I am reposting in an article hopping more to come.

jupyter TPOT Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [3]:
```
from tpot import TPOTClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split

digits = load_digits()
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target,
                                                    train_size=0.75, test_size=0.25)

tpot = TPOTClassifier(generations=5, population_size=20, verbosity=2)
tpot.fit(X_train, y_train)
print(tpot.score(X_test, y_test))
tpot.export('tpot_mnist_pipeline.py')
```

```
Generation 1 - Current best internal CV score: 0.9613589039341084
Generation 2 - Current best internal CV score: 0.9613589039341084
Generation 3 - Current best internal CV score: 0.9643802172254123
Generation 4 - Current best internal CV score: 0.9643802172254123
Generation 5 - Current best internal CV score: 0.966558515711851

Best pipeline: LinearSVC(KNeighborsClassifier(ExtraTreesClassifier(input_matrix, bootstrap=False, criterion=entrop
y, max_features=0.5, min_samples_leaf=16, min_samples_split=5, n_estimators=100), n_neighbors=3, p=2, weights=unif
orm), C=0.001, dual=True, loss=squared_hinge, penalty=l2, tol=0.1)
0.9888888888888889
```

In [ ]:
```
from tpot import TPOTRegressor
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split

housing = load_boston()
X_train, X_test, y_train, y_test = train_test_split(housing.data, housing.target,
                                                    train_size=0.75, test_size=0.25)

tpot = TPOTRegressor(generations=5, population_size=20, verbosity=2)
tpot.fit(X_train, y_train)
print(tpot.score(X_test, y_test))
tpot.export('tpot_boston_pipeline.py')
```

Now I'd like to have TPOT run on CSV dataset The goal to achive that is to have the following parameters ready X_train, X_test, y_train, y_test = train_test_split(someDataset.data, someDataset.target, train_size=0.75, test_size=0.25)

In [30]:
```
import pandas as pd

cancerDataFrame = pd.read_csv('data.csv')
target = cancerDataFrame['diagnosis']
#convert target values: Malignant=1
target = target.replace('B', 0)
target = target.replace('M', 1)
target.head()
data = cancerDataFrame.drop(columns=['id', 'diagnosis'])
data.head()

X_train, X_test, y_train, y_test = train_test_split(data, target,
                                                    train_size=0.75, test_size=0.25)

tpot = TPOTClassifier(generations=5, population_size=20, verbosity=2)
tpot.fit(X_train, y_train)
print(tpot.score(X_test, y_test))
tpot.export('tpot_cancer_pipeline.py')
```

```
Imputing missing values in feature set

Generation 1 - Current best internal CV score: 0.9671650055370986
Generation 2 - Current best internal CV score: 0.9671650055370986
Generation 3 - Current best internal CV score: 0.9671650055370986
Generation 4 - Current best internal CV score: 0.9671650055370986
Generation 5 - Current best internal CV score: 0.9671650055370986

Best pipeline: LinearSVC(PolynomialFeatures(input_matrix, degree=2, include_bias=False, interaction_only=False), C
=0.5, dual=False, loss=squared_hinge, penalty=l1, tol=0.001)
Imputing missing values in feature set
0.965034965034965
```

In [ ]: