

Finding best model and hyper parameters for sklearn digits dataset classification

```
In [1]: from sklearn import datasets
digits = datasets.load_digits()
```

```
In [2]: from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
```

```
In [3]: model_params = {
    'svm': {
        'model': svm.SVC(gamma='auto'),
        'params': {
            'C': [1,10,20],
            'kernel': ['rbf','linear']
        }
    },
    'random_forest': {
        'model': RandomForestClassifier(),
        'params': {
            'n_estimators': [1,5,10]
        }
    },
    'logistic_regression': {
        'model': LogisticRegression(solver='liblinear',multi_class='auto'),
        'params': {
            'C': [1,5,10]
        }
    },
    'naive_bayes_gaussian': {
        'model': GaussianNB(),
        'params': {}
    },
    'naive_bayes_multinomial': {
        'model': MultinomialNB(),
        'params': {}
    },
    'decision_tree': {
        'model': DecisionTreeClassifier(),
        'params': {
            'criterion': ['gini','entropy'],
        }
    }
}
```

```
In [4]: from sklearn.model_selection import GridSearchCV
import pandas as pd
scores = []

for model_name, mp in model_params.items():
    clf = GridSearchCV(mp['model'], mp['params'], cv=5, return_train_score=False)
    clf.fit(digits.data, digits.target)
    scores.append({
        'model': model_name,
        'best_score': clf.best_score_,
        'best_params': clf.best_params_
    })

df = pd.DataFrame(scores, columns=['model', 'best_score', 'best_params'])
df
```

Out[4]:

| | model | best_score | best_params |
|---|-------------------------|------------|------------------------------|
| 0 | svm | 0.947697 | {'C': 1, 'kernel': 'linear'} |
| 1 | random_forest | 0.900410 | {'n_estimators': 10} |
| 2 | logistic_regression | 0.922114 | {'C': 1} |
| 3 | naive_bayes_gaussian | 0.806928 | {} |
| 4 | naive_bayes_multinomial | 0.870350 | {} |
| 5 | decision_tree | 0.808035 | {'criterion': 'entropy'} |

For me the winner is svm (C=1, kernel=linear) with 94.93% score. It could be different for you as I have limited my parameters to be certain values only

| Date | Author |
|------------|---------------------------|
| 11-14-2021 | Ehsan Zia |