krishnaik06 / **GRIDSearchCV** Public

Code Issues Pull requests Actions Projects Security Insights

master ···

GRIDSearchCV / **Gridsearchcv.ipynb**

 **krishnaik06** Add files via upload History

1 contributor

405 lines (405 sloc) | 9.68 KB ···

Use GridSearchCV and select the best hyperparamter for Support Vector machine

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Advertising_data.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

In [4]:

```
dataset.head()
```

Out[4]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0

In [39]:

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 5)
```

In [40]:
```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [46]:
```
# Fitting Kernel SVM to the Training set
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)
```

Out[46]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=-1, probability=False, random_state=0,
  shrinking=True, tol=0.001, verbose=False)
```

In [47]:
```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

In [48]:
```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [49]:
```
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_test,y_pred)
```

In [50]:
```
accuracy
```

Out[50]: 0.85

In [29]:
```
# Applying Grid Search to find the best model and the best parameters
from sklearn.model_selection import GridSearchCV
parameters = [{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
              {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0
grid_search = GridSearchCV(estimator = classifier,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           cv = 10,
                           n_jobs = -1)
grid_search = grid_search.fit(X_train, y_train)
```

```
C:\Users\krish.naik\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:84
1: DeprecationWarning: The default of the `iid` parameter will change from True to False in version 0.22 and w
ill be removed in 0.24. This will change numeric results when test-set sizes are unequal.
  DeprecationWarning)
```

In [30]:
```
accuracy = grid_search.best_score_
```

In [31]:
```
accuracy
```

Out[31]:
```
0.9033333333333333
```

In [20]:
```
grid_search.best_params_
```

Out[20]:
```
{'C': 1, 'gamma': 0.7, 'kernel': 'rbf'}
```

In [51]:
```
classifier = SVC(kernel = 'rbf', gamma=0.7)
classifier.fit(X_train, y_train)
```

Out[51]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=0.7, kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

In [52]:
```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

In [53]:
```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [54]:
```
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_test,y_pred)
```

In [55]:
```
accuracy
```

Out[55]: 0.93

In [ ]: