Marcos Geraldo Braga Emiliano

19.1.4012

Estimando o desempenho de classificadores

Importaçoes:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from google.colab import drive
drive.mount('/content/drive')
from sklearn.svm import SVC
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
train_df= pd.read_csv('/content/drive/My Drive/kaggle-titanic/train.csv')
```

Pequenos Tratamentos:

```
train_df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
```

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ×

```
Cabin          687
Embarked         2
dtype: int64
```

```
train_df["Age"].fillna(train_df["Age"].mean(), inplace = True)
```

```
train_df['Sex']=train_df['Sex'].replace('male', 0)
train_df['Sex']=train_df['Sex'].replace('female', 1)
```

```
train_df.drop(['Name', 'PassengerId', 'Fare', 'Ticket','Embarked', 'Cabin'], axis = 1, inp
```

```
missing=train_df.isnull().sum().sort_values(ascending=False)
missing=missing.drop(missing[missing==0].index)
missing
```

```
Series([], dtype: int64)
```

```
X= train_df.drop('Survived', axis=1)
y= train_df['Survived']
```

Criando as 10 Folds

```
from sklearn.model_selection import KFold, StratifiedKFold, cross_val_score
kf =KFold(n_splits=10, shuffle=True, random_state=42)

cnt = 1
# split()  method generate indices to split data into training and test set.
for train_index, test_index in kf.split(X, y):
    print(f'Fold:{cnt}, Train set: {len(train_index)}, Test set:{len(test_index)}')
    cnt += 1
```

```
Fold:1, Train set: 801, Test set:90
Fold:2, Train set: 802, Test set:89
Fold:3, Train set: 802, Test set:89
Fold:4, Train set: 802, Test set:89
Fold:5, Train set: 802, Test set:89
Fold:6, Train set: 802, Test set:89
Fold:7, Train set: 802, Test set:89
Fold:8, Train set: 802, Test set:89
Fold:9, Train set: 802, Test set:89
Fold:10, Train set: 802, Test set:89
```

Definindo função RMSE

```
def rmse(score):
    rmse = np.sqrt(-score)
```

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ×

Aplicando a SVM com kernel rbf para as folds

```
score = cross_val_score(SVC(kernel='rbf'), X, y, cv= kf, scoring="neg_mean_squared_error")
print(f'Scores for each fold: {score}')
rmse(score.mean())
```

```
Scores for each fold: [-0.38888889 -0.39325843 -0.41573034 -0.28089888 -0.35955056 -(
 -0.39325843 -0.33707865 -0.46067416 -0.26966292]
rmse= 0.60
```

Media:

```
score.mean()
```

```
-0.36023720349563054
```

Desvio Padrão:

```
score.std()
```

```
0.058683593235936644
```

Separando os dados a serem utilizados no Grid Search

```
from sklearn.model_selection import train_test_split

X= train_df.drop('Survived', axis=1)
y= train_df['Survived']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

Criando um modelo grid que utiliza uma SVM

```
from sklearn.model_selection import GridSearchCV

svm = SVC()
grid_parameters = {'C':[0.01,0.1,1, 10, 100, 1000],'gamma':[1, 0.1, 0.01, 0.001, 0.0001]}
grid_search = GridSearchCV(svm,grid_parameters, cv=5)
```

```
grid_search.fit(X_train, y_train)
```

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ×

```
             param_grid={'C': [0.01, 0.1, 1, 10, 100, 1000],
                         'gamma': [1, 0.1, 0.01, 0.001, 0.0001]})
```

Buscando o melhor resultado para o "estimador" e os parametros

```
grid_search.best_estimator_
```

```
SVC(C=1000, gamma=0.001)
```

```
grid_search.best_params_

     {'C': 1000, 'gamma': 0.001}
```

Inferindo os dados de Validação

```
y_pred_grid= grid_search.predict(X_test)
```

Matriz Confusão dos resultados obtidos

```
from sklearn.metrics import classification_report, confusion_matrix
```

```
confusion_matrix(y_test, y_pred_grid)

     array([[137,  17],
            [ 35,  79]])
```

```
print(classification_report(y_test, y_pred_grid))

                   precision    recall  f1-score   support

                0       0.80      0.89      0.84       154
                1       0.82      0.69      0.75       114

         accuracy                           0.81       268
        macro avg       0.81      0.79      0.80       268
     weighted avg       0.81      0.81      0.80       268
```

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ✕

Produtos pagos do Colab - Cancelar contratos

✓ 0s conclusão: 09:43 ● ×

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ×