model_selection

January 22, 2023

Subject of notebook : Comment each step of best_model set_matplotlib_close
Name of the auther : Qadir Shahbaz
Where to contact : qadir_shahbaz@yahoo.co.uk
date : 22/01/2023

0.1 Import libraries

[ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

0.1.1 Load dataset

0.1.2 Assign X and y

0.1.3 used get_dummies to convert data type. This is called label encoding

0.1.4 deal with missing values

[ ]:
```
df = sns.load_dataset('titanic')
X = df[["pclass", "sex", "age", "sibsp", "parch", "fare"]]
y = df["survived"]
X = pd.get_dummies(X, columns=["sex"])
X.age.fillna(value= X["age"].mean(), inplace = True)
```

0.2 from sklearn import supervised machine learning algorithm

0.3 from sklearn import evaluation methods for classification

[ ]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, f1_score, precision_score,␣
 ↪recall_score
```

0.4 We spit the data to check on the basis of metrics

```
[ ]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
  ↪random_state=42)
```

0.5 We have assign variable models to all the classification algorithm.

0.6 We have assign variable models_names to all the model names

```
[ ]: models = [LogisticRegression(), SVC(), DecisionTreeClassifier(),␣
  ↪RandomForestClassifier(), KNeighborsClassifier()]
model_names = ['Logistic Regression', 'SVM', 'Decision Tree', 'Random Forest',␣
  ↪'KNN']
```

0.6.1 first we have made an empty list and assign variable models_scores to it

0.6.2 Secondly we used for loop to iterate the variables models and model names. This code will find y_pred and accuracy and will append the values in varaible models_scores for each model.

```
[ ]: models_scores = []
for model, model_name in zip(models, model_names):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    models_scores.append([model_name,accuracy])
```

0.7 The below code will sort the result in descending order

```
[ ]: sorted_models = sorted(models_scores, key=lambda x: x[1], reverse=True)
for model in sorted_models:
    print("Accuracy Score: ",f'{model[0]} : {model[1]:.2f}')
```

```
Accuracy Score:  Logistic Regression : 0.81
Accuracy Score:  Random Forest : 0.81
Accuracy Score:  Decision Tree : 0.77
Accuracy Score:  KNN : 0.69
Accuracy Score:  SVM : 0.66
```

0.8 This code will find y_pred and precision and will append the values in varaible models_scores for each model.

```
[ ]: models = [LogisticRegression(), SVC(), DecisionTreeClassifier(),␣
  ↪RandomForestClassifier(), KNeighborsClassifier()]
model_names = ['Logistic Regression', 'SVM', 'Decision Tree', 'Random Forest',␣
  ↪'KNN']
models_scores = []
```

```
for model, model_name in zip(models, model_names):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    Precision = precision_score(y_test, y_pred)
    models_scores.append([model_name,Precision])

sorted_models = sorted(models_scores, key=lambda x: x[1], reverse=True)
for model in sorted_models:
    print("Precision Score: ", f'{model[0]} : {model[1]:.2f}')
```

```
Precision Score:  Logistic Regression : 0.80
Precision Score:  Random Forest : 0.79
Precision Score:  SVM : 0.76
Precision Score:  Decision Tree : 0.73
Precision Score:  KNN : 0.66
```

0.9 This code will find y_pred and recall and will append the values in varaible models_scores for each model.

[ ]:
```
models = [LogisticRegression(), SVC(), DecisionTreeClassifier(),␣
 ↪RandomForestClassifier(), KNeighborsClassifier()]
model_names = ['Logistic Regression', 'SVM', 'Decision Tree', 'Random Forest',␣
 ↪'KNN']
models_scores = []
for model, model_name in zip(models, model_names):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    Recall = recall_score(y_test, y_pred)
    models_scores.append([model_name,Recall])

sorted_models = sorted(models_scores, key=lambda x: x[1], reverse=True)
for model in sorted_models:
    print("Recall Score: ",f'{model[0]} : {model[1]:.2f}')
```

```
Recall Score:  Random Forest : 0.73
Recall Score:  Logistic Regression : 0.72
Recall Score:  Decision Tree : 0.69
Recall Score:  KNN : 0.54
Recall Score:  SVM : 0.26
```

0.10 This code will find y_pred and F1 and will append the values in varaible models_scores for each model.

[ ]:
```
models = [LogisticRegression(), SVC(), DecisionTreeClassifier(),␣
 ↪RandomForestClassifier(), KNeighborsClassifier()]
model_names = ['Logistic Regression', 'SVM', 'Decision Tree', 'Random Forest',␣
 ↪'KNN']
models_scores = []
```

```
for model, model_name in zip(models, model_names):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    F1 = f1_score(y_test, y_pred)
    models_scores.append([model_name,F1])

sorted_models = sorted(models_scores, key=lambda x: x[1], reverse=True)
for model in sorted_models:
    print("F1 Score: ",f'{model[0]} : {model[1]:.2f}')
```

```
F1 Score:  Logistic Regression : 0.76
F1 Score:  Random Forest : 0.75
F1 Score:  Decision Tree : 0.72
F1 Score:  KNN : 0.59
F1 Score:  SVM : 0.38
```