```
# -*- coding: utf-8 -*-
"""Assignment-2 Wine dataset.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1iB57oxuX22PdzN4jEsAsmRAr2sIPVcGL

#Import required modules
"""

import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import seaborn as sns
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier

"""#Load Dataset"""

wine = datasets.load_wine() # it's source is same as :
https://archive.ics.uci.edu/ml/datasets/wine

dir(wine)

wine.data

print(wine.feature_names)
print(wine.target_names)
print(wine.target)

df = pd.DataFrame(data=wine.data, columns=wine.feature_names)
df.head()

df["target"] = wine.target
df.head()

wine.target_names

"""#DataFrame ready to perform"""

len(df)

X = df.drop(["target"], axis="columns")
y = df.target
print(X.head())
print(y.head())

"""#SVC Classfier

##Linear SVC Classifier
"""

linear_SVC_classifier = SVC(kernel='linear')
linear_SVC_classifier

"""###train size : test size = 70% : 30%

"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)     #
70% training data, 30% testing data

print(len(X_train))
print(len(y_test))

linear_SVC_classifier.fit(X_train, y_train)

y_pred = linear_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 60% : 40%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=0)    #
60% training data, 40% testing data

print(len(X_train))
print(len(y_test))

linear_SVC_classifier.fit(X_train, y_train)

y_pred = linear_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 50% : 50%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)    #
50% training data, 50% testing data

print(len(X_train))
print(len(y_test))

linear_SVC_classifier.fit(X_train, y_train)

y_pred = linear_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 40% : 60%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.6, random_state=0)

print(len(X_train))
print(len(y_test))

linear_SVC_classifier.fit(X_train, y_train)

y_pred = linear_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 30% : 70%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=0)

print(len(X_train))
print(len(y_test))

linear_SVC_classifier.fit(X_train, y_train)

y_pred = linear_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""##Polynomial SVC Classifier"""

poly_SVC_classifier = SVC(kernel='poly')
poly_SVC_classifier

"""###train size : test size = 70% : 30%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

print(len(X_train))
print(len(y_test))

poly_SVC_classifier.fit(X_train, y_train)

y_pred = poly_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""### train size : test size = 60% : 40%


"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=0)

print(len(X_train))
print(len(y_test))

poly_SVC_classifier.fit(X_train, y_train)

y_pred = poly_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))
```

```
sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 50% : 50%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)

print(len(X_train))
print(len(y_test))

poly_SVC_classifier.fit(X_train, y_train)

y_pred = poly_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 40% : 60%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.6, random_state=0)

print(len(X_train))
print(len(y_test))

poly_SVC_classifier.fit(X_train, y_train)

y_pred = poly_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 30% : 70%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=0)

print(len(X_train))
print(len(y_test))

poly_SVC_classifier.fit(X_train, y_train)

y_pred = poly_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""##Gaussain SVC Classifier"""

gaussain_SVC_classifier = SVC(kernel='rbf')
gaussain_SVC_classifier

"""###train size : test size = 70% : 30%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

print(len(X_train))
print(len(y_test))
```

```
gaussain_SVC_classifier.fit(X_train, y_train)

y_pred = gaussain_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""### train size : test size = 60% : 40%


"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=0)

print(len(X_train))
print(len(y_test))

gaussain_SVC_classifier.fit(X_train, y_train)

y_pred = gaussain_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 50% : 50%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)

print(len(X_train))
print(len(y_test))

gaussain_SVC_classifier.fit(X_train, y_train)

y_pred = gaussain_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 40% : 60%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.6, random_state=0)

print(len(X_train))
print(len(y_test))

gaussain_SVC_classifier.fit(X_train, y_train)

y_pred = gaussain_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))
```

```
sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 30% : 70%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=0)

print(len(X_train))
print(len(y_test))

gaussain_SVC_classifier.fit(X_train, y_train)

y_pred = gaussain_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""##Sigmoid SVC Classifier"""

sigmoid_SVC_classifier = SVC(kernel='sigmoid')
sigmoid_SVC_classifier

"""###train size : test size = 70% : 30%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

print(len(X_train))
print(len(y_test))

sigmoid_SVC_classifier.fit(X_train, y_train)

y_pred = sigmoid_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""### train size : test size = 60% : 40%


"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=0)

print(len(X_train))
print(len(y_test))

sigmoid_SVC_classifier.fit(X_train, y_train)

y_pred = sigmoid_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 50% : 50%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
```

```
print(len(X_train))
print(len(y_test))

sigmoid_SVC_classifier.fit(X_train, y_train)

y_pred = sigmoid_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 40% : 60%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.6, random_state=0)

print(len(X_train))
print(len(y_test))

sigmoid_SVC_classifier.fit(X_train, y_train)

y_pred = sigmoid_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 30% : 70%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=0)

print(len(X_train))
print(len(y_test))

sigmoid_SVC_classifier.fit(X_train, y_train)

y_pred = sigmoid_SVC_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n", cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""#MLP Classifier"""

mlp_classifier = MLPClassifier(learning_rate='constant', max_iter=600)
mlp_classifier

"""###train size : test size = 70% : 30%

"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)    #
70% training data, 30% testing data

print(len(X_train))
print(len(y_test))

mlp_classifier.fit(X_train, y_train)
```

```
y_pred = mlp_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 60% : 40%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=0)    #
60% training data, 40% testing data

print(len(X_train))
print(len(y_test))

mlp_classifier.fit(X_train, y_train)

y_pred = mlp_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 50% : 50%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)    #
50% training data, 50% testing data

print(len(X_train))
print(len(y_test))

mlp_classifier.fit(X_train, y_train)

y_pred = mlp_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 40% : 60%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.6, random_state=0)

print(len(X_train))
print(len(y_test))

mlp_classifier.fit(X_train, y_train)

y_pred = mlp_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))
```

```
sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 30% : 70%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=0)

print(len(X_train))
print(len(y_test))

mlp_classifier.fit(X_train, y_train)

y_pred = mlp_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""#Random Forest Classifier"""

rfc_classifier = RandomForestClassifier(n_estimators=20)
rfc_classifier

"""###train size : test size = 70% : 30%

"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)    #
70% training data, 30% testing data

print(len(X_train))
print(len(y_test))

rfc_classifier.fit(X_train, y_train)

y_pred = rfc_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 60% : 40%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=0)    #
60% training data, 40% testing data

print(len(X_train))
print(len(y_test))

rfc_classifier.fit(X_train, y_train)

y_pred = rfc_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))
```

```python
sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 50% : 50%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)    # 50% training data, 50% testing data

print(len(X_train))
print(len(y_test))

rfc_classifier.fit(X_train, y_train)

y_pred = rfc_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 40% : 60%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.6, random_state=0)

print(len(X_train))
print(len(y_test))

rfc_classifier.fit(X_train, y_train)

y_pred = rfc_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)

"""###train size : test size = 30% : 70%"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=0)

print(len(X_train))
print(len(y_test))

rfc_classifier.fit(X_train, y_train)

y_pred = rfc_classifier.predict(X_test)
print(f"Accuracy: {100 * accuracy_score(y_test,y_pred)}%\n")
cf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n")
print(cf_matrix)
print("\nClassification Report:\n")
print(classification_report(y_test,y_pred))

sns.heatmap(cf_matrix, annot=True)
```