

BAGGING DEMONSTRATION

```
In [ ]: #import libraries
import pandas as pd
```

```
In [ ]: #Load the dataset
dataset = pd.read_csv("C:\\Users\\Smarty Rohit\\Desktop\\parkinsons.csv")
dataset.sample(10)
```

```
In [ ]: #check missing values
dataset.isnull().sum()
```

```
In [ ]: # quick overview ofa dataset
dataset.describe()
```

```
In [ ]: # counting the values in column--status
dataset.status.value_counts()
```

```
In [ ]: #initialize x and y variables
X=dataset.drop('status',axis='columns')
y=dataset.status
```

```
In [ ]: #dataset scalling
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled=scaler.fit_transform(X)
X_scaled[:3]
```

```
In [ ]: #splitting data into training and testing set
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split (X,y,stratify=y,random_state=10)
```

```
In [ ]: #shape of training set
X_train.shape
```

```
In [ ]: #shape of testing set
X_test.shape
```

```
In [ ]: # count values for training set
y_train.value_counts()
```

```
In [ ]: # define a decision tree classifier

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score

results = cross_val_score(DecisionTreeClassifier(),X,y,cv=5)
results
```

```
In [ ]: #mean accuracy score
results.mean()
```

```
In [ ]: # Define a bagging classifier
```

```
from sklearn.ensemble import BaggingClassifier

model = BaggingClassifier(base_estimator=DecisionTreeClassifier(),n_estimators = 100,
                          max_samples=0.8,oob_score=True,random_state=0)

model.fit(X_train, y_train)
#accuracy score--->training set
model.oob_score_
```

```
In [ ]: #accuracy score --> testing set
model.score(X_test,y_test)
```

```
In [ ]: # to use cross_val_score method

model = BaggingClassifier(base_estimator=DecisionTreeClassifier(),n_estimators = 100,
                          max_samples=0.8,oob_score=True,random_state=0)
results = cross_val_score(model,X,y,cv=5)
results.mean()
```

```
In [ ]: # define a random forest classifier...

from sklearn.ensemble import RandomForestClassifier

results = cross_val_score(RandomForestClassifier(),X,y,cv=5)
results.mean()
```

```
In [ ]: #source code ---> end
#Done by Rohit Naresh Saktel
```