

Voting Classifier using Sklearn

Voting Classifier:

The voting classifier is a machine learning model that trains the integration of multiple models and predicts the output (class) according to their higher chances for the selected class as a result. It simply summarized the findings of each divider transmitted to the voting divider and predicted the output phase according to the number of votes. The idea is that instead of making separate models dedicated and finding each other's accuracy, we create a single model that trains these types and predicts output based on their combined votes for each of the output classes.

Take example breast cancer dataset for classification using different classifiers like KNeighborsClassifier, SVC, DecisionTreeClassifier and use voting classifiers as follows,

```
# importing libraries
from sklearn.ensemble import VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_breast_cancer
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split

# loading breast cancer dataset
cancer = load_breast_cancer()
X = cancer.data
Y = cancer.target

# train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X, Y, test_size = 0.20, random_state = 42)

# ensemble of models
models = []
models.append(('KNN', KNeighborsClassifier()))
models.append(('SVC', SVC(gamma='auto', probability=True)))
models.append(('DTC', DecisionTreeClassifier()))

# Voting Classifier with hard voting
vot_hard = VotingClassifier(estimators = models, voting
='hard')
vot_hard.fit(X_train, y_train)
y_pred = vot_hard.predict(X_test)

# using accuracy_score metric to predict accuracy
h_score = accuracy_score(y_test, y_pred)
```

```
print("HARD SCORE = % d" % h_score)

# Voting Classifier with soft voting
vot_soft = VotingClassifier(estimators = models, voting
='soft')
vot_soft.fit(X_train, y_train)
y_pred = vot_soft.predict(X_test)

# using accuracy_score
s_score = accuracy_score(y_test, y_pred)
print("SOFT SCORE =% d" % s_score)
```

Output:

Accuracy score using hard and soft scoring is given,

HARD SCORE = 1

SOFT SCORE = 1