



3s



```
import pandas as pd
from sklearn.datasets import load_breast_cancer

# Load the breast cancer dataset
data = load_breast_cancer()

# Convert the dataset into a pandas DataFrame
df = pd.DataFrame(data.data, columns=data.feature_names)

# Add the target variable to the DataFrame
df['target'] = data.target

# Split the data into X and y
X = df.drop('target', axis=1)
y = df['target']
```

```

54s ✓ from sklearn.model_selection import KFold, StratifiedKFold, ShuffleSplit, RepeatedKFold
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier, RandomForestClassifier
from sklearn.metrics import accuracy_score

# Define the SVM model
svm_model = SVC(kernel='linear')

# Define the cross-validation methods
cv_methods = [
    KFold(n_splits=5),
    StratifiedKFold(n_splits=5),
    ShuffleSplit(n_splits=5),
    RepeatedKFold(n_splits=5, n_repeats=2)
]

# Define the models to be evaluated
models = [
    svm_model,
    DecisionTreeClassifier(),
    AdaBoostClassifier(n_estimators=100),
    RandomForestClassifier(n_estimators=100)
]

# Define the metric to be used for performance evaluation
metric = accuracy_score

# Loop through the cross-validation methods and models, and evaluate their performance
for cv_method in cv_methods:
    for model in models:
        scores = []
        for train_idx, test_idx in cv_method.split(X, y):
            X_train, y_train = X.iloc[train_idx], y.iloc[train_idx]
            X_test, y_test = X.iloc[test_idx], y.iloc[test_idx]
            model.fit(X_train, y_train)
            y_pred = model.predict(X_test)
            score = metric(y_test, y_pred)
            scores.append(score)
        print(f"{cv_method.__class__.__name__} - {model.__class__.__name__}: {sum(scores)/len(scores):.3f}")

```

```

KFold - SVC: 0.953
KFold - DecisionTreeClassifier: 0.900
KFold - AdaBoostClassifier: 0.963
KFold - RandomForestClassifier: 0.954
StratifiedKFold - SVC: 0.946
StratifiedKFold - DecisionTreeClassifier: 0.924
StratifiedKFold - AdaBoostClassifier: 0.975
StratifiedKFold - RandomForestClassifier: 0.967
ShuffleSplit - SVC: 0.937
ShuffleSplit - DecisionTreeClassifier: 0.919
ShuffleSplit - AdaBoostClassifier: 0.951
ShuffleSplit - RandomForestClassifier: 0.968
RepeatedKFold - SVC: 0.953
RepeatedKFold - DecisionTreeClassifier: 0.926
RepeatedKFold - AdaBoostClassifier: 0.967
RepeatedKFold - RandomForestClassifier: 0.957

```

The two cross - validation methods that I want to compare are Stratified K Fold and Repeated K Fold. They seem to have very high accuracy values, especially for the adaboost model, for which they have an accuracy value of 0.975 and 0.967 respectively. For SVC the best cross-validation method is tied between KFold and RepeatedKFold. For the Decision tree model the value seems to be lower than the other models across the board but the best cross-validation method for it is RepeatedKFold. And last the Shuffle split is the best cross v validation method for the random forest model. The metric that I chose for performance evaluation is essentially accuracy percentage