

7. Write implementation of various Ensemble Techniques.

```
import pandas as pd
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier,
VotingClassifier, StackingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

```
# Load the Breast Cancer dataset
data = load_breast_cancer()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)
```

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```
# Feature scaling for algorithms sensitive to feature scales
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
# 1. Bagging - Random Forest Classifier
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
rf_clf.fit(X_train, y_train)
rf_pred = rf_clf.predict(X_test)
rf_acc = accuracy_score(y_test, rf_pred)
print(f"Random Forest Accuracy: {rf_acc * 100:.2f}%")
```

Random Forest Accuracy: 96.49%

```
# 2. Boosting - AdaBoost Classifier
ada_clf = AdaBoostClassifier(n_estimators=100, random_state=42)
ada_clf.fit(X_train, y_train)
ada_pred = ada_clf.predict(X_test)
ada_acc = accuracy_score(y_test, ada_pred)
print(f"AdaBoost Accuracy: {ada_acc * 100:.2f}%")
```

AdaBoost Accuracy: 97.37%

```
# 3. Voting Classifier
log_clf = LogisticRegression(max_iter=1000, random_state=42)
dt_clf = DecisionTreeClassifier(random_state=42)
svc_clf = SVC(probability=True, random_state=42)
```

```
voting_clf = VotingClassifier(estimators=[
    ('lr', log_clf),
    ('dt', dt_clf),
    ('svc', svc_clf)
], voting='soft')

voting_clf.fit(X_train_scaled, y_train)
voting_pred = voting_clf.predict(X_test_scaled)
voting_acc = accuracy_score(y_test, voting_pred)
print(f"Voting Classifier Accuracy: {voting_acc * 100:.2f}%")
```

Voting Classifier Accuracy: 96.49%

```
# 4. Stacking Classifier
stacking_clf = StackingClassifier(
    estimators=[
        ('lr', log_clf),
        ('dt', dt_clf),
        ('svc', svc_clf)
    ],
    final_estimator=RandomForestClassifier(n_estimators=100, random_state=42)
)

stacking_clf.fit(X_train_scaled, y_train)
stacking_pred = stacking_clf.predict(X_test_scaled)
stacking_acc = accuracy_score(y_test, stacking_pred)
print(f"Stacking Classifier Accuracy: {stacking_acc * 100:.2f}%")
```

Stacking Classifier Accuracy: 97.37%