Name: Raghav Sharma, Roll no: 25201313

Q1) Implement a stacking ensemble model using the 'sklearn' library in python. Your task is to use a base model consisting of a decision tree and a logistic regression model and then stack them using a logistic regression model as the final estimator. Train your ensemble model on a dataset of your choice (e.g., Iris or any suitable dataset) and evaluate its accuracy on a test set. Ensure to properly split the data into training and testing sets and report the accuracy of the model.

In [10]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import StackingClassifier
from sklearn.metrics import accuracy_score, classification_report
```

In [11]:
```
iris = load_iris()
X, y = iris.data, iris.target
```

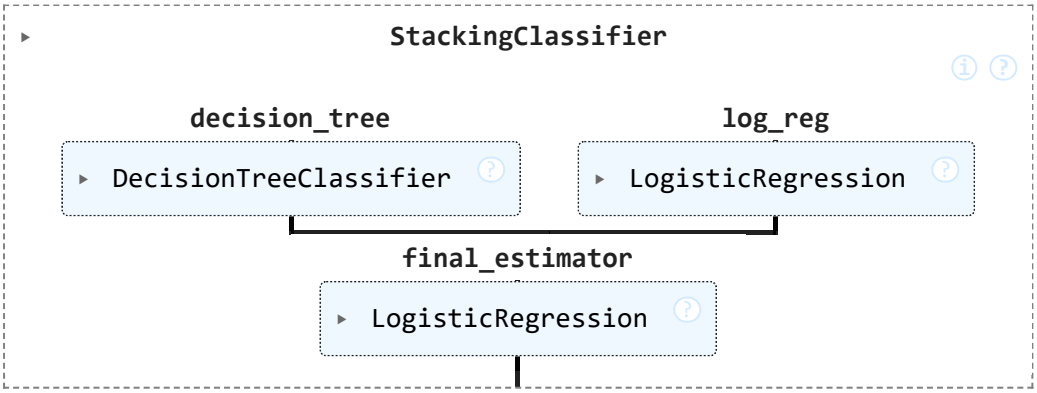
In [12]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

In [13]:
```
base_models = [
("decision_tree", DecisionTreeClassifier(random_state=42)),
("log_reg", LogisticRegression(max_iter=200, random_state=42))
]
```

In [14]:
```
stacking_model = StackingClassifier(
estimators=base_models,
final_estimator=LogisticRegression(max_iter=200, random_state=42)
)
```

In [15]:
```
stacking_model.fit(X_train, y_train)
```

Out[15]:



In [16]:
```
y_pred = stacking_model.predict(X_test)
```

In [17]:
```
accuracy = accuracy_score(y_test, y_pred)
print(f"Stacking Ensemble Model Accuracy: {accuracy:.4f}")
```

```
Stacking Ensemble Model Accuracy: 0.9667
```

In [23]:
```
# Train & evaluate individual models for comparison
dt = DecisionTreeClassifier(random_state=42)
```

```
lr = LogisticRegression(max_iter=200, random_state=42)

models = {
    "Decision Tree": dt,
    "Logistic Regression": lr,
    "Stacking Ensemble": stacking_model
}

results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    results[name] = accuracy_score(y_test, y_pred)

print("\nModel Performance Comparison:")
for name, acc in results.items():
    print(f"{name}: {acc:.4f}")
```

```
Model Performance Comparison:
Decision Tree: 0.9333
Logistic Regression: 0.9667
Stacking Ensemble: 0.9667
```

In [ ]:

In [ ]:

In [ ]: