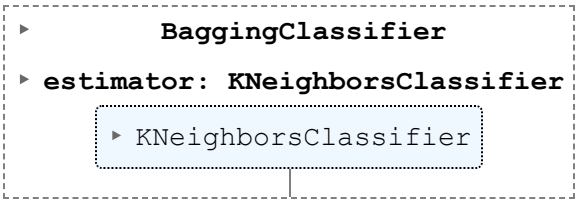
In [12]:
```
from sklearn.ensemble import BaggingClassifier,VotingClassifier,RandomForestRegressor,Ba
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier,DecisionTreeRegressor
from sklearn.linear_model import LogisticRegression
```

In [4]:
```
iris=load_iris()
X=iris.data
y=iris.target
```

In [7]:
```
weak_model=KNeighborsClassifier()
model=BaggingClassifier(estimator=weak_model,n_estimators=10)
model.fit(X,y)
```

Out[7]:



In [8]:
```
sample=[1.5,.8,2.8,.5]
model.predict([sample])
```

Out[8]:
```
array([1])
```

In [9]:
```
model.predict_proba([sample])
```

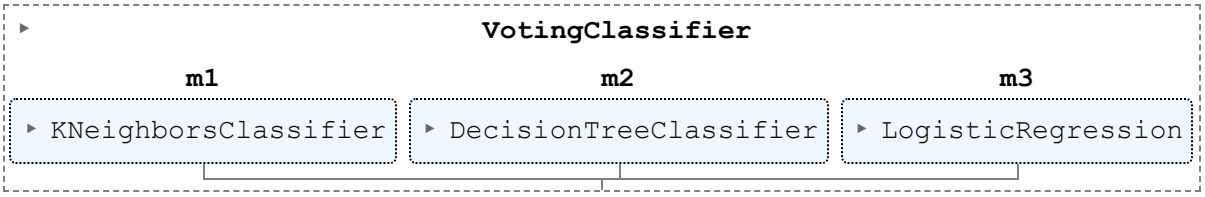
Out[9]:
```
array([[0.28, 0.72, 0.  ]])
```

In [14]:
```
#Stacking

weak_model_1=KNeighborsClassifier()
weak_model_2=DecisionTreeClassifier()
weak_model_3=LogisticRegression(max_iter=200)

model=VotingClassifier(estimators=[('m1',weak_model_1),('m2',weak_model_2),('m3',weak_mo
model.fit(X,y)
```

Out[14]:



In [15]:
```
sample=[1.5,.8,2.8,.5]
model.predict([sample])
```

Out[15]:
```
array([1])
```

In [18]:
```
#Boosting--->Adaboost

from sklearn.ensemble import AdaBoostClassifier,AdaBoostRegressor
model=AdaBoostClassifier() #weak model=Dtree depth=1
model.fit(X,y)
model.predict([sample])
```

Out[18]:
```
array([1])
```

In [19]:
```
wk_model=LogisticRegression(max_iter=200)
model=AdaBoostClassifier(estimator=wk_model)
model.fit(X,y)
model.predict([sample])
```

Out[19]:
```
array([1])
```

In [20]:
```
wk_model=KNeighborsClassifier()
model=AdaBoostClassifier(estimator=wk_model)
model.fit(X,y)
model.predict([sample])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[20], line 3
      1 wk_model=KNeighborsClassifier()
      2 model=AdaBoostClassifier(estimator=wk_model)
----> 3 model.fit(X,y)
      4 model.predict([sample])

File ~\anaconda3\Lib\site-packages\sklearn\ensemble\_weight_boosting.py:142, in BaseWeig
htBoosting.fit(self, X, y, sample_weight)
    139 sample_weight /= sample_weight.sum()
    141 # Check parameters
--> 142 self._validate_estimator()
    144 # Clear any previous fit results
    145 self.estimators_ = []

File ~\anaconda3\Lib\site-packages\sklearn\ensemble\_weight_boosting.py:525, in AdaBoost
Classifier._validate_estimator(self)
    517         raise TypeError(
    518             "AdaBoostClassifier with algorithm='SAMME.R' requires "
    519             "that the weak learner supports the calculation of class "
   (...)
    522             "algorithm='SAMME' instead."
    523         )
    524 if not has_fit_parameter(self.estimator_, "sample_weight"):
--> 525     raise ValueError(
    526         f"{self.estimator.__class__.__name__} doesn't support sample_weight."
    527     )

ValueError: KNeighborsClassifier doesn't support sample_weight.
```

In [21]:
```
wk_model=LogisticRegression(max_iter=200)
model=AdaBoostClassifier(estimator=wk_model,algorithm='SAMME.R') #weak model predicts pr
model.fit(X,y)
model.predict([sample])
```

Out[21]:
```
array([1])
```

In [22]:
```
from sklearn.svm import SVC
```

In [23]:
```
wk_model=SVC() #does not compute probs
model=AdaBoostClassifier(estimator=wk_model,algorithm='SAMME.R') #weak model predicts pr
model.fit(X,y)
model.predict([sample])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[23], line 3
      1 wk_model=SVC()
      2 model=AdaBoostClassifier(estimator=wk_model,algorithm='SAMME.R') #weak model pre
dicts probs
----> 3 model.fit(X,y)
```

```
      4 model.predict([sample])

File ~\anaconda3\Lib\site-packages\sklearn\ensemble\_weight_boosting.py:142, in BaseWeig
htBoosting.fit(self, X, y, sample_weight)
    139 sample_weight /= sample_weight.sum()
    141 # Check parameters
--> 142 self._validate_estimator()
    144 # Clear any previous fit results
    145 self.estimators_ = []

File ~\anaconda3\Lib\site-packages\sklearn\ensemble\_weight_boosting.py:517, in AdaBoost
Classifier._validate_estimator(self)
    515 if self.algorithm == "SAMME.R":
    516     if not hasattr(self.estimator_, "predict_proba"):
--> 517         raise TypeError(
    518             "AdaBoostClassifier with algorithm='SAMME.R' requires "
    519             "that the weak learner supports the calculation of class "
    520             "probabilities with a predict_proba method.\n"
    521             "Please change the base estimator or set "
    522             "algorithm='SAMME' instead."
    523         )
    524 if not has_fit_parameter(self.estimator_, "sample_weight"):
    525     raise ValueError(
    526         f"{self.estimator.__class__.__name__} doesn't support sample_weight."
    527     )

TypeError: AdaBoostClassifier with algorithm='SAMME.R' requires that the weak learner su
pports the calculation of class probabilities with a predict_proba method.
Please change the base estimator or set algorithm='SAMME' instead.
```

In [24]:
```
wk_model=SVC() #does not compute probs
model=AdaBoostClassifier(estimator=wk_model,algorithm='SAMME') #weak model predicts prob
model.fit(X,y)
model.predict([sample])
```

Out[24]:
```
array([0])
```

In [ ]: