In [1]:

```python
from numpy import mean
from sklearn.datasets import make_regression
from sklearn.model_selection import cross_val_score, RepeatedKFold
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.ensemble import StackingRegressor

```

In [2]:

```python
X,y=make_regression(n_samples=100,n_features=20,random_state=1)
```

In [3]:

```python
def get_stacking():
    level0=list()
    level0.append(('knn',KNeighborsRegressor()))
    level0.append(('svm',SVR()))
    level1=LinearRegression()
    model=StackingRegressor(estimators=level0,final_estimator=level1)
    return model


#level0= base model
#level1=meta model
```

In [5]:

```python
def get_models():
    models=dict()
    models['knn']=KNeighborsRegressor()
    models['cart']=DecisionTreeRegressor()
    models['svm']=SVR()
    models['stacking']=get_stacking()
    return models
```

In [6]:

```python
def evaluate_model(model,X,y):
    cv=RepeatedKFold(n_splits=10,n_repeats=3,random_state=1)
    scores=cross_val_score(model,X,y,scoring='neg_mean_absolute_error
    return scores

```

In [7]:

```python
models=get_models()
results,names=list(),list()
for name,model in models.items():
    scores=evaluate_model(model,X,y)
    results.append(scores)
    names.append(model)
    print(name,mean(scores)) #the least is the best and high is worst

```

```
knn -103.77473576335751
cart -139.8829767739784
svm -141.4454992633503
stacking -99.49681805536376
```