```
import pandas as pd
import numpy as np
import sklearn
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import BaggingClassifier,
GradientBoostingClassifier
from mlxtend.classifier import StackingCVClassifier
from sklearn.metrics import r2_score
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import MinMaxScaler
```

EDA

```
iris = load_iris()
df = pd.DataFrame(iris.data, columns = iris.feature_names)
df['target'] = iris.target
```

```
df.head()
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width
(cm)  \
0                5.1               3.5                1.4
0.2
1                4.9               3.0                1.4
0.2
2                4.7               3.2                1.3
0.2
3                4.6               3.1                1.5
0.2
4                5.0               3.6                1.4
0.2

   target
0       0
1       0
2       0
3       0
4       0
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   target             150 non-null    int32
dtypes: float64(4), int32(1)
memory usage: 5.4 KB
```

```
y = df["target"]
x = df.drop(["target"], axis = 1)
```

```
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(x)
```

```
x_train, x_test, y_train, y_test = train_test_split(scaled_data, y, 
test_size = 0.2, random_state = 0)
```

Bagging

```
bc = BaggingClassifier(n_estimators = 5, oob_score = True, 
random_state = 10)
bc_prediction = bc.fit(x_train, y_train).predict(x_test)
r2_score(y_test, bc_prediction)
```

```
C:\Users\sofya\anaconda3\lib\site-packages\sklearn\ensemble\
_bagging.py:789: UserWarning: Some inputs do not have OOB scores. This 
probably means too few estimators were used to compute any reliable 
oob estimates.
  warn(
C:\Users\sofya\anaconda3\lib\site-packages\sklearn\ensemble\
_bagging.py:795: RuntimeWarning: invalid value encountered in 
true_divide
  oob_decision_function = predictions / predictions.sum(axis=1)[:, 
np.newaxis]
```

```
0.9381443298969072
```

Boosting

```
gb = GradientBoostingClassifier(random_state = 1)
gb_prediction = gb.fit(x_train, y_train).predict(x_test)
r2_score(y_test, gb_prediction)
```

```
1.0
```

Stacking

```
clf1 = BaggingClassifier(random_state = 42)
clf2 = GradientBoostingClassifier(random_state = 42)
clf3 = LogisticRegression()
meta_learner = LogisticRegression()
st_prediction = StackingCVClassifier(classifiers = [clf1, clf2, clf3],
meta_classifier = meta_learner).fit(x_train, y_train).predict(x_test)
r2_score(y_test, st_prediction)
```

```
1.0
```