Bagging Regression (Bootstrap Aggregation)

```
# from google.colab import files
# up = files.upload()
```

import dataset

```
import pandas as pd
df = pd.read_csv('df.csv')
df.head(3)
```

	f1	f2	f3	f4	T
0	16.5	202.0	865.500000	1880.0	50.000000
1	18.0	204.0	688.000000	1738.5	44.000000
2	18.0	203.0	583.666667	1470.0	66.666667

```
# df.info()
```

cleaning

```
# clean the data
```

encoding

```
# encode the data
```

define x , y

```
import numpy as np
x = df[['f1', 'f2', 'f3']].values
y = df['T'].values
```

spliting

```
### finding best random state



from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
```

scaling

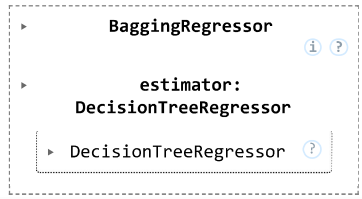
```
# do not typically require data scaling
```

fit the model

```
### K-fold cross validation

# from sklearn.ensemble import BaggingRegressor
# from sklearn.tree import DecisionTreeRegressor
# from sklearn.model_selection import GridSearchCV
```

```
# parameters = {
#     '': [],
#     '': []
# }

# br = BaggingRegressor(random_state=42)
# gs = GridSearchCV(estimator=br, param_grid=parameters, cv=5)

# gs.fit(x_train, y_train)

# best_params = gs.best_params_
# print(best_params)

# def param
#
```

```
from sklearn.ensemble import BaggingRegressor
from sklearn.tree import DecisionTreeRegressor

base_model = DecisionTreeRegressor()

br = BaggingRegressor(estimator=base_model, n_estimators=100, random_state=42)
br.fit(x_train, y_train)
```



predict test data

```
yhat_test = br.predict(x_test)
```

evaluate the model

```
from sklearn.metrics import r2_score
print("r2-score (train data): %0.4f" % r2_score(y_train, br.predict(x_train)))
print("r2-score (test data): %0.4f" % r2_score(y_test, yhat_test))
```

```
r2-score (train data): 0.9068
r2-score (test data): 0.3846
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
print(f"MSE (train data): {mean_squared_error(y_train, br.predict(x_train))}")
print(f"MAE (train data): {mean_absolute_error(y_train, br.predict(x_train))}")
print(f"MSE (test data): {mean_squared_error(y_test, yhat_test)}")
print(f"MAE (test data): {mean_absolute_error(y_test, yhat_test)}")
```

```
MSE (train data): 17.594789759359262
MAE (train data): 3.1879666666773336
MSE (test data): 90.11624066750916
MAE (test data): 7.838800000028
```

save the model

```
# import joblib
# joblib.dump(br, 'br_model.pkl')
```

load the model

```
# import joblib
# br = joblib.load('br_model.pkl')
```