In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import learning_curve
```

In [7]:
```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winec
data = pd.read_csv(url, sep=';')
data
```

Out[7]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcol
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.45	
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.49	
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.44	1
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	
...	...	...	...	...	...	...	...	...	...	...	
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	0.50	1
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	0.46	
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	0.46	
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	0.38	1
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	0.32	1

4898 rows × 12 columns

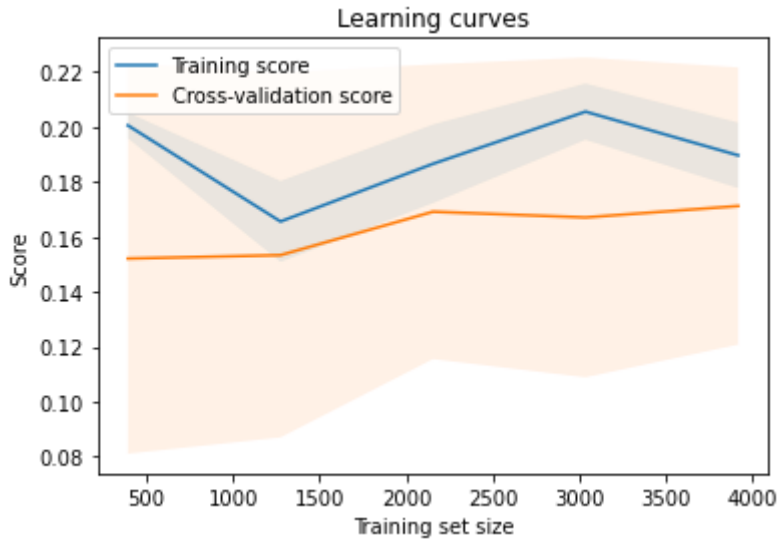
In [8]:
```
# Load the wine quality dataset
#data = pd.read_csv('winequality-white.csv', delimiter=';')

# Split the dataset into training and testing sets
X = data[['alcohol']].values
y = data['quality'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_st
```

In [9]:
```
# Create a linear regression object and fit the model to the training data
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predict the quality of the wine for the test data
y_pred = regressor.predict(X_test)

# Print the model's parameters
print('Coefficients: ', regressor.coef_)
print('Intercept: ', regressor.intercept_)

# Print the performance metrics
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

```
print('Mean Squared Error: ', mse)
print('Root Mean Squared Error: ', rmse)
print('R-squared: ', r2)
```

```
Coefficients:  [0.31100587]
Intercept:  2.620696868564679
Mean Squared Error:  0.7306442340192562
Root Mean Squared Error:  0.8547773008329457
R-squared:  0.1710201454832172
```

In [10]:
```
# Plot the learning curves
train_sizes, train_scores, test_scores = learning_curve(regressor, X, y, cv=5)
train_mean = np.mean(train_scores, axis=1)
train_std = np.std(train_scores, axis=1)
test_mean = np.mean(test_scores, axis=1)
test_std = np.std(test_scores, axis=1)
plt.plot(train_sizes, train_mean, label='Training score')
plt.plot(train_sizes, test_mean, label='Cross-validation score')
plt.fill_between(train_sizes, train_mean - train_std, train_mean + train_std, alpha
plt.fill_between(train_sizes, test_mean - test_std, test_mean + test_std, alpha=0.
plt.xlabel('Training set size')
plt.ylabel('Score')
plt.title('Learning curves')
plt.legend()
plt.show()
```



In [ ]: