```
import pandas as pd
```

```
df = pd.read_excel('/content/car.xlsx')
df
```

	brand	model	transmission	age	fuel	price	mileage	power	seats
0	18	244	1	4	1	1231000.0	19.01	4.496471	5
1	10	263	1	6	4	786000.0	19.01	4.496471	5
2	31	123	1	2	1	1489000.0	19.01	4.496471	5
3	9	55	0	1	4	1227000.0	19.01	4.496471	5
4	8	82	1	3	1	887000.0	19.01	4.496471	5
...	...	...	...	...	...	...	...	...	...
32009	5	199	1	6	4	292000.0	19.01	4.496471	5
32010	32	295	1	6	4	534000.0	19.01	4.496471	5
32011	33	25	1	8	4	424000.0	19.01	4.496471	5
32012	10	120	0	5	4	685000.0	19.01	4.496471	5
32013	31	247	1	2	4	392000.0	19.01	4.496471	5

32014 rows × 9 columns

```
X = df.drop('price',axis=1)
y = df['price']
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=True)
```

```
from sklearn.ensemble import AdaBoostRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score

# Initialize model
ada = AdaBoostRegressor(random_state=42)

# Train
ada.fit(X_train, y_train)

# Predict
y_train_pred = ada.predict(X_train)
y_test_pred = ada.predict(X_test)

# Evaluate
print("Train R²:", r2_score(y_train, y_train_pred))
print("Test R²:", r2_score(y_test, y_test_pred))
print("Train MSE:", mean_squared_error(y_train, y_train_pred))
print("Test MSE:", mean_squared_error(y_test, y_test_pred))

# Cross-validation
cv_scores = cross_val_score(ada, X, y, cv=5, scoring='r2')
print("Cross-validation R² scores:", cv_scores)
print("Mean CV R² score:", cv_scores.mean())
```

```
Train R²: 0.5976538825389944
Test R²: 0.5507899179022089
Train MSE: 202895850934.69754
Test MSE: 204239568385.2668
Cross-validation R² scores: [ 0.71147855  0.54838461  0.40965438  0.39251297 -0.56204891]
Mean CV R² score: 0.29999632146036975
```

```
from sklearn.model_selection import GridSearchCV

# Hyperparameter grid
param_grid = {
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.01, 0.1, 1]
}

# GridSearchCV setup
grid = GridSearchCV(AdaBoostRegressor(random_state=42), param_grid, cv=5, scoring='r2', verbose=1, n_jobs=-1)

# Train
```

```
grid.fit(X_train, y_train)

# Best model
best_ada = grid.best_estimator_

# Predict
y_train_pred = best_ada.predict(X_train)
y_test_pred = best_ada.predict(X_test)

# Evaluate
print("🔧 Best Parameters:", grid.best_params_)
print("Train R²:", r2_score(y_train, y_train_pred))
print("Test R²:", r2_score(y_test, y_test_pred))
print("Train MSE:", mean_squared_error(y_train, y_train_pred))
print("Test MSE:", mean_squared_error(y_test, y_test_pred))

# Cross-validation on best model
cv_scores = cross_val_score(best_ada, X, y, cv=5, scoring='r2')
print("Cross-validation R² scores:", cv_scores)
print("Mean CV R² score:", cv_scores.mean())
```

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
🔧 Best Parameters: {'learning_rate': 0.01, 'n_estimators': 200}
Train R²: 0.7077784433295954
Test R²: 0.6934496127055665
Train MSE: 147362031914.8475
Test MSE: 139377367705.0321
Cross-validation R² scores: [ 0.64983936  0.70626173  0.70594039  0.70734182 -0.15875902]
Mean CV R² score: 0.5221248578083586
```