```
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

Next steps: Generate code with df View recommended plots New interactive sheet

```
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 RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score

# Initialize model
rf = RandomForestRegressor(random_state=42)

# Train
rf.fit(X_train, y_train)

# Predict
y_train_pred = rf.predict(X_train)
y_test_pred = rf.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(rf, X, y, cv=5, scoring='r2')
```

```
print("Cross-validation R² scores:", cv_scores)
print("Mean CV R² score:", cv_scores.mean())
```

```
Train R²: 0.9558684229720562
Test R²: 0.8918851685022822
Train MSE: 22254754017.95725
Test MSE: 49155901439.30192
Cross-validation R² scores: [0.80217462 0.87467664 0.88807976 0.89361595 0.72405685]
Mean CV R² score: 0.8365207636902584
```

```
from sklearn.model_selection import GridSearchCV

# Hyperparameter grid
param_grid = {
    'n_estimators': [1, 2],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2]
}

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

# Train
grid.fit(X_train, y_train)

# Best model
best_rf = grid.best_estimator_

# Predict
y_train_pred = best_rf.predict(X_train)
y_test_pred = best_rf.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_rf, 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 24 candidates, totalling 120 fits
🔧 Best Parameters: {'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimato
Train R²: 0.9206326597464329
Test R²: 0.879101461458048
Train MSE: 40023510451.12758
Test MSE: 54968190417.51255
Cross-validation R² scores: [0.76552368 0.8454939  0.8666347  0.86270514 0.68885915]
Mean CV R² score: 0.8058433142099147
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.