```
import pandas as pd
import numpy as np
import re
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error, r2_score
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
```

```
df = pd.read_csv('/dataset.csv', dtype=str)
```

```
# 1. Convert 'price' column to numeric
df.loc[:, 'price'] = pd.to_numeric(df['price'], errors='coerce')
df = df.dropna(subset=['price'])

# 2. Clean 'mileage' column
df.loc[:, 'mileage'] = pd.to_numeric(df['mileage'], errors='coerce', downcast='integer').fillna(0)

# 3. Convert 'cylinders' strings like "4 cylinders" to numeric
df.loc[:, 'cylinders'] = df['cylinders'].str.extract(r'(\d+)')  # Extract digits only
df.loc[:, 'cylinders'] = pd.to_numeric(df['cylinders'], errors='coerce', downcast='integer')
df.loc[:, 'cylinders'] = df['cylinders'].fillna(df['cylinders'].median())

# 4. Similarly clean 'doors'
df.loc[:, 'doors'] = pd.to_numeric(df['doors'], errors='coerce', downcast='integer')
df.loc[:, 'doors'] = df['doors'].fillna(df['doors'].median())

# 5. Fix year
df.loc[:, 'year'] = pd.to_numeric(df['year'], errors='coerce', downcast='integer')
df.loc[:, 'year'] = df['year'].fillna(2024)  # Use default year if missing
```

```
/usr/local/lib/python3.11/dist-packages/numpy/lib/_nanfunctions_impl.py:1231: RuntimeWarning: Mean of empty slice
  return np.nanmean(a, axis, out=out, keepdims=keepdims)
/tmp/ipython-input-7-271560095.py:11: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and wi
  df.loc[:, 'cylinders'] = df['cylinders'].fillna(df['cylinders'].median())
/tmp/ipython-input-7-271560095.py:15: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and wi
  df.loc[:, 'doors'] = df['doors'].fillna(df['doors'].median())
/tmp/ipython-input-7-271560095.py:19: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and wi
  df.loc[:, 'year'] = df['year'].fillna(2024)  # Use default year if missing
```

```
# Feature engineering
df['age'] = 2025 - df['year']
df['interior_color'] = df['interior_color'].fillna('Unknown')
df['exterior_color'] = df['exterior_color'].fillna('Unknown')
df['has_leather'] = df['description'].str.lower().str.contains('leather', na=False).astype(int)
df['has_navigation'] = df['description'].str.lower().str.contains('navigation', na=False).astype(int)
```

```
# Extract engine size from engine column
def extract_engine_size(engine_str):
    if pd.isna(engine_str):
        return np.nan
    match = re.search(r'(\d+\.\d+L)', engine_str)
    if match:
        return float(match.group(1).replace('L', ''))
    return np.nan

df['engine_size'] = df['engine'].apply(extract_engine_size)
df['engine_size'] = df['engine_size'].fillna(df['engine_size'].median())
```

```
# Define features and target
numerical_features = ['mileage', 'cylinders', 'doors', 'age', 'engine_size', 'has_leather', 'has_navigation']
categorical_features = ['make', 'model', 'trim', 'body', 'fuel', 'transmission', 'exterior_color', 'interior_color', 'drivetrain']
X = df[numerical_features + categorical_features]
y = df['price']
```

```
# Create preprocessing pipeline
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False), categorical_features)
    ])
```

```
# Create and train the model pipeline
model = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42))
])
```

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error: {mae:.2f}")
print(f"R-squared: {r2:.2f}")

# Optionally, perform cross-validation
cv_scores = cross_val_score(model, X, y, cv=5, scoring='neg_mean_absolute_error')
cv_mae_scores = -cv_scores
print(f"Cross-validated MAE scores: {cv_mae_scores}")
print(f"Average Cross-validated MAE: {cv_mae_scores.mean():.2f}")
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction**2 / new_sample_count
Mean Absolute Error: 6219.84
R-squared: 0.73
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction**2 / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction**2 / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction**2 / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction**2 / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction**2 / new_sample_count
Cross-validated MAE scores: [6202.00763735 6452.47596064 6634.70205225 6628.91342391 7453.58028774]
Average Cross-validated MAE: 6674.34
```

```
# Perform cross-validation
cv_scores = cross_val_score(model, X, y, cv=5, scoring='r2')
mean_cv_r2 = cv_scores.mean()
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction**2 / new_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
/usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
```

```
      T = new_sum / new_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
      new_unnormalized_variance -= correction**2 / new_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
      updated_mean = (last_sum + new_sum) / updated_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
      T = new_sum / new_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
      new_unnormalized_variance -= correction**2 / new_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
      updated_mean = (last_sum + new_sum) / updated_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
      T = new_sum / new_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
      new_unnormalized_variance -= correction**2 / new_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1101: RuntimeWarning: invalid value encountered in divide
      updated_mean = (last_sum + new_sum) / updated_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1106: RuntimeWarning: invalid value encountered in divide
      T = new_sum / new_sample_count
    /usr/local/lib/python3.11/dist-packages/sklearn/utils/extmath.py:1126: RuntimeWarning: invalid value encountered in divide
      new_unnormalized_variance -= correction**2 / new_sample_count
```

```python
# Get feature importance
feature_names = numerical_features + list(model.named_steps['preprocessor']
                                          .named_transformers_['cat']
                                          .get_feature_names_out(categorical_features))
importances = model.named_steps['regressor'].feature_importances_
feature_importance = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
top_features = feature_importance.sort_values(by='Importance', ascending=False).head(5)
```

```python
# Print results
print("Vehicle Price Prediction Model Results")
print("-" * 40)
print(f"Mean Absolute Error (MAE): ${mae:.2f}")
print(f"R² Score (Test Set): {r2:.4f}")
print(f"Mean Cross-Validation R² Score: {mean_cv_r2:.4f}")
print("\nTop 5 Most Important Features:")
for _, row in top_features.iterrows():
    print(f"{row['Feature']}: {row['Importance']:.4f}")
```

```
Vehicle Price Prediction Model Results
----------------------------------------
Mean Absolute Error (MAE): $6219.84
R² Score (Test Set): 0.7259
Mean Cross-Validation R² Score: 0.6910

Top 5 Most Important Features:
fuel_Diesel: 0.0803
make_Mercedes-Benz: 0.0758
make_BMW: 0.0633
model_Compass: 0.0603
drivetrain_Front-wheel Drive: 0.0561
```

```python
# Save the model (optional)
import joblib
joblib.dump(model, 'vehicle_price_model.pkl')
```

```
['vehicle_price_model.pkl']
```

```python
# Example prediction for a single vehicle
example_vehicle = X.iloc[0:1]
predicted_price = model.predict(example_vehicle)[0]
print(f"\nExample Prediction for {df.iloc[0]['name']}: ${predicted_price:.2f}")
```

```

Example Prediction for 2024 Jeep Wagoneer Series II: $72616.47
```