```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder

# Load the dataset (replace the path with your actual file path)
url = "your_dataset_url_or_file_path.csv"  # Put the dataset URL or file path here
df = pd.read_csv(url)

# Display the first few rows of the dataset
print(df.head())

# Basic Information about the dataset
print(df.info())

# Checking for missing values
print(df.isnull().sum())

# Handle missing values (for example, filling missing numerical columns with median)
df.fillna(df.median(), inplace=True)

# For categorical columns, let's use LabelEncoder to convert them to numerical values
categorical_columns = df.select_dtypes(include=['object']).columns
encoder = LabelEncoder()
```

```
for col in categorical_columns:
    df[col] = encoder.fit_transform(df[col])

# Now, let's define the features (X) and target (y)
X = df.drop('price', axis=1)  # Assuming 'price' is the target column
y = df['price']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a model (Linear Regression or Random Forest)
# Example with Linear Regression
linear_regressor = LinearRegression()
linear_regressor.fit(X_train, y_train)

# Predictions
y_pred_linear = linear_regressor.predict(X_test)

# Example with Random Forest Regressor
random_forest_regressor = RandomForestRegressor(n_estimators=100, random_state=42)
random_forest_regressor.fit(X_train, y_train)

# Predictions
y_pred_rf = random_forest_regressor.predict(X_test)

# Evaluate the model performance
print("Linear Regression Model Evaluation:")
print(f"Mean Absolute Error: {mean_absolute_error(y_test, y_pred_linear)}")
print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred_linear)}")
print(f"R² Score: {r2_score(y_test, y_pred_linear)}")
```

```
print("\nRandom Forest Model Evaluation:")

print(f"Mean Absolute Error: {mean_absolute_error(y_test, y_pred_rf)}")

print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred_rf)}")

print(f"R² Score: {r2_score(y_test, y_pred_rf)}")


# Visualize actual vs predicted prices

plt.figure(figsize=(10, 6))

plt.scatter(y_test, y_pred_rf, color='blue', label='Random Forest Predicted')

plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--',
label='Perfect Prediction Line')

plt.title("Actual vs Predicted Car Prices (Random Forest)")

plt.xlabel("Actual Price")

plt.ylabel("Predicted Price")

plt.legend()

plt.show()
```