## **Car Price Prediction – Beginner-Friendly Coding Test**

Dataset: [Car Price Prediction](https://www.kaggle.com/datasets/hellbuoy/car-price-prediction)

## **Data Cleaning & Preprocessing (1–15)**

**1.** *Load the dataset into a Pandas DataFrame and display the first 5 rows.* Explain what each column means in your own words.  
 Resource: [Pandas read\_csv() – Basic Usage](https://www.geeksforgeeks.org/python-pandas-dataframe-read_csv/)

**2.** *Check for missing values in the dataset.* Show both the number of missing values per column and the percentage.  
 Resource: [How to Check Missing Values in Pandas](https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/)

**3.** *Drop any rows where the target variable (selling price) is missing.* Explain why we can’t train a model with missing target values.  
 Resource: [Pandas dropna() – Remove Missing Values](https://www.geeksforgeeks.org/python-pandas-dataframe-dropna/)

**4.** *Fill missing values in the “mileage” column with the column’s mean.* Explain why filling missing values can sometimes be better than dropping rows.  
 Resource: [Fill Missing Data in Pandas](https://www.geeksforgeeks.org/fillna-in-pandas/)

**5.** *Remove duplicate rows from the dataset.* Explain how duplicate rows can affect model training.  
 Resource: [Drop Duplicates in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-drop_duplicates/)

**6.** *Convert the “year” column into the car’s age.* Example: If the car’s year is 2015 and the current year is 2025, the age is 10 years.  
 Resource: [Pandas DateTime – Extract Year](https://www.geeksforgeeks.org/python-pandas-to_datetime/)

**7.** *Check the unique values in the “fuel type” column.* Explain why knowing all possible values in a categorical column is important before encoding.  
 Resource: [Unique Values in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-nunique/)

**8.** *Replace inconsistent values in the “transmission” column.* For example, replace “manual” and “Manual” with “Manual” (consistent casing).  
 Resource: [Replace Values in Pandas](https://www.geeksforgeeks.org/python-pandas-replace-method/)

**9.** *Check if there are any outliers in the “selling price” column using a boxplot.* Explain why outliers can affect model accuracy.  
 Resource: [Boxplot in Python](https://www.geeksforgeeks.org/box-plot-in-python-using-matplotlib/)

**10.** *Remove cars that are priced below 10,000 or above 5,000,000 (possible outliers).* Explain why setting realistic price limits can help improve model performance.  
 Resource: [Conditional Filtering in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-loc/)

**11.** *Standardize the column names so they are all lowercase and contain underscores instead of spaces.* Resource: [Rename Columns in Pandas](https://www.geeksforgeeks.org/rename-columns-in-pandas-dataframe/)

**12.** *Check if any numerical columns are stored as strings and convert them to numbers.* Resource: [Convert Strings to Numbers in Pandas](https://www.geeksforgeeks.org/convert-pandas-dataframe-column-to-numeric/)

**13.** *Create a new column for “price per kilometer” by dividing the selling price by the mileage.* Resource: [Create New Columns in Pandas](https://www.geeksforgeeks.org/create-a-new-column-in-pandas-dataframe-based-on-the-existing-columns/)

**14.** *Ensure the dataset index is properly reset after cleaning.* Resource: [Reset Index in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-reset_index/)

**15.** *Save your cleaned dataset as a new CSV file called cleaned\_cars.csv.*Resource: [Save DataFrame to CSV](https://www.geeksforgeeks.org/exporting-a-pandas-dataframe-to-csv-file/)

## **Exploratory Data Analysis (EDA) (16–30)**

**16.** *Find the average selling price of cars in the dataset.* Resource: [Mean in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-mean/)

**17.** *Find the most common fuel type in the dataset.* Resource: [Value Counts in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-value_counts-method/)

**18.** *Plot a histogram of the selling prices.* Resource: [Matplotlib Histogram](https://www.geeksforgeeks.org/plot-histogram-using-matplotlib/)

**19.** *Plot the relationship between car age and selling price using a scatter plot.* Resource: [Scatter Plot in Python](https://www.geeksforgeeks.org/scatter-plot-using-matplotlib/)

**20.** *Group cars by fuel type and find the average selling price for each group.* Resource: [GroupBy in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-groupby/)

**21.** *Plot a bar chart showing the number of cars per transmission type.* Resource: [Bar Chart in Matplotlib](https://www.geeksforgeeks.org/bar-plot-in-matplotlib/)

**22.** *Find the car with the highest mileage in the dataset.* Resource: [idxmax in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-idxmax/)

**23.** *Calculate the correlation between mileage and selling price.* Resource: [Correlation in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-corr/)

**24.** *Plot a heatmap showing the correlations between all numeric columns.* Resource: [Heatmap in Seaborn](https://www.geeksforgeeks.org/heatmap-in-python/)

**25.** *Check if manual cars are generally cheaper or more expensive than automatic cars.* Resource: [Filtering and Grouping in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-query/)

**26.** *Find the average selling price for each year of manufacture.* Resource: [GroupBy and Mean in Pandas](https://www.geeksforgeeks.org/python-pandas-groupby-method/)

**27.** *Plot a line chart showing the trend of selling prices over the years.* Resource: [Line Plot in Matplotlib](https://www.geeksforgeeks.org/matplotlib-pyplot-plot-in-python/)

**28.** *Find the most expensive car for each fuel type.* Resource: [GroupBy with idxmax in Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-idxmax/)

**29.** *Find out which brand appears most frequently in the dataset.* Resource: [Pandas String Operations](https://www.geeksforgeeks.org/python-pandas-string-methods/)

**30.** *Check the top 5 most common combinations of fuel type and transmission.* Resource: [GroupBy Multiple Columns](https://www.geeksforgeeks.org/python-pandas-dataframe-groupby/)

## **Machine Learning (31–40)**

**31.** *What are the main assumptions of Linear Regression, and how can you check them using this dataset?* Explain the assumptions (Linearity, Homoscedasticity, Normality of residuals, No multicollinearity, Independence of errors). Then, write code to verify at least **three** of these assumptions using visualizations or statistics.  
 Resource: [Linear Regression Assumptions Made Simple](https://towardsdatascience.com/assumptions-of-linear-regression-why-you-should-care-1c1aab8c12f1)

**32.** *Train a Linear Regression model to predict the car’s selling price using selected features.* Split the dataset into train/test sets, train the model, and print the coefficients and intercept.  
 Resource: [Linear Regression in Python – Step-by-Step Guide](https://www.geeksforgeeks.org/linear-regression-python-implementation/)

**33.** *Evaluate your Linear Regression model using R² Score and Mean Squared Error (MSE).* Also, explain in your own words what each metric tells you about the model’s performance.  
 Resource: [Regression Evaluation Metrics](https://www.analyticsvidhya.com/blog/2021/05/evaluation-metrics-for-regression-models/)

**34.** *Apply Lasso Regression on the dataset and compare its performance with Linear Regression.* Explain what Lasso Regression does differently and why it can be useful when you have many features.  
 Resource: [Lasso Regression – Explained Simply](https://www.geeksforgeeks.org/lasso-regression/)

**35.** *Apply Ridge Regression on the dataset and compare its performance with both Linear and Lasso Regression.* Discuss why Ridge might perform better or worse in certain cases.  
 Resource: [Ridge Regression – Explained Simply](https://www.geeksforgeeks.org/ridge-regression/)

**36.** *Use cross-validation to evaluate your Ridge Regression model.* Explain why cross-validation gives a better estimate of model performance than a single train-test split.  
 Resource: [Cross-Validation in Machine Learning](https://www.geeksforgeeks.org/cross-validation-machine-learning/)

**37.** *Plot the predicted vs actual prices for your best-performing model.* Discuss what the plot reveals about the accuracy of your predictions.  
 Resource: [Predicted vs Actual Plot in Python](https://www.geeksforgeeks.org/how-to-plot-actual-vs-predicted-values/)

**38.** *Use GridSearchCV to find the best alpha value for Ridge Regression.* Explain what alpha does in Ridge/Lasso regression and why tuning it is important.  
 Resource: [GridSearchCV Explained with Example](https://www.geeksforgeeks.org/ml-hyperparameter-tuning-using-gridsearchcv/)

**39.** *Try Polynomial Regression on the dataset.* Compare the results with Linear Regression and explain when polynomial regression might be useful.  
 Resource: [Polynomial Regression in Python](https://www.geeksforgeeks.org/python-implementation-of-polynomial-regression/)

**40.** *Summarize your findings:* Which model performed best (Linear, Lasso, Ridge, or Polynomial)?  
 Explain why you think it performed better and what you would do next to improve predictions.  
 Resource: [Choosing the Right Regression Model](https://towardsdatascience.com/choosing-the-right-regression-model-19d1dce18401)