Cars4You ML Pipeline Group 2

Bussines Needs

- Identify the problem and solution: build a model that can predicts car prices without the need of a person to see the car

Data Integration

- Import the provided training and test datasets.

Data Exploration and Understanding

- Check for duplicates, missing values, data types, and null entries.
- Perform descriptive statistics (minimum, maximum, mean, median, and standard deviation).
- Clean categorical variables, correcting spelling errors.
- Conduct Exploratory Data Analysis (EDA), including: distribution plots (histograms, boxplots), frequency plots (bar charts, stacked bar charts), correlation heatmaps, relationship plots (pair plots)
- Remove the following variables:
  - hasDamage → contains no variation (all values are 0).
  - paintQuality% → requires mechanic input, and we aim for a model that does not depend on mechanics.
  - Brand and model → contain too many categories and have missing values that are difficult to impute accurately.

Data Preparation

- **Handle inconsistencies (Outliers):** Convert impossible negative values to their absolute values, and remove cars registered after 2020. A total of 1,849 observations (≈2.4%) were removed.
- **Split the data:** Divide the dataset into training (80%) and validation (20%) sets using a Hold-Out (HO) approach.
- **Impute missing categorical values:** Fill missing values in fuelType and transmission using their respective modes.
- **Transform features:** Simplify fuelType into three categories — diesel, petrol, and other — by merging the original five, as they showed similar behavior during the EDA phase. Convert previousOwners into a boolean variable (had previous owners or not).
- **Encode categorical variables:** Create dummy variables for fuelType and transmission, dropping one dummy to avoid multicollinearity.
- **Feature scaling:** Apply RobustScaler to numerical features to reduce the influence of outliers.
- **Impute missing numerical values:** Use a KNN Imputer with the best parameters founded.
- **Feature selection:** Perform RFE on Linear Regression, and apply Lasso Regression to identify the most relevant features, **Concluding** that is better to keep all the features because the RFE results showed that the difference in MAE between using 10 and 9 (suggested) features was less than £1, indicating minimal improvement. Additionally, the Lasso coefficients for previousOwners and tax were around |60|, suggesting that these variables still contribute useful information and are not entirely irrelevant.
- **Note:** All preprocessing steps — including mode imputation, scaling, imputation, and model parameter tuning — were fitted only on the training set and later applied to the validation set to prevent data leakage.

Modelling and Assement

- Establish a benchmark using a Linear Regression model.
- Train a Random Forest model for comparison.
- Evaluate performance using MAE (Mean Absolute Error), as it is less sensitive to outliers and aligns with the Kaggle competition's evaluation metric.
- Compare alternative models (Random Forest) against the benchmark to identify which one achieves a lower MAE on the validation dataset and exhibits less overfitting.
- Select the best model and train it on the entire dataset (training + validation) before making final predictions.

Deployment

- Analyze the required preprocessing steps (cleaning, encoding, scaling, and imputing).
- Build a pipeline that receives the test dataset and performs all necessary steps — preprocessing, prediction, and file creation.
- Output the predictions as a .csv file suitable for Kaggle submission.