Car Price Estimation Model: Ridge & Lasso Techniques 🚗 💰

I developed a Car Price Estimation Model employing **Ridge** and **Lasso** regression techniques to accurately predict car prices based on various features. Below, I outline the key steps, methodologies, and insights derived from this project.

---

1. Import Libraries 📦

I began by importing essential libraries for data manipulation, visualisation, and model building:

- **Pandas** for data handling
- **NumPy** for numerical operations
- **Seaborn** and **Matplotlib** for visualisation
- **Scikit-learn** for model selection, evaluation, and regression techniques

---

2. Load Dataset & Display Overview 📊

I loaded the dataset from a CSV file containing car attributes and their corresponding prices. Initial exploration involved displaying the first few rows, which provided insight into the structure of the data.

---

3. Data Preprocessing 🔄

I undertook a comprehensive data cleaning process, which included:

- **Missing Values**: I identified and dropped any rows with missing values to ensure data integrity.
- **Duplicates**: I eliminated duplicate entries to maintain a unique dataset.
- **Outlier Detection**: Using the **Interquartile Range (IQR)** method, I detected and removed outliers in the Driven_kms feature, ensuring the model wasn't skewed by extreme values.
- **Logarithmic Transformation**: I applied logarithmic transformations to the Selling_Price and Present_Price features to normalise their distributions, aiding in model performance.

- **Label Encoding**: Categorical features such as Fuel_Type, Selling_type, and Transmission were converted into numeric format using label encoding for compatibility with machine learning algorithms.
- **Feature Engineering**: I transformed the Year feature into Age to represent the car's age, providing a more relevant feature for prediction.
- **Dropping Irrelevant Columns**: I removed unnecessary columns (Car_Name, Year) that wouldn't contribute to the prediction task.

I also performed data visualisations to understand distributions and correlations, revealing valuable insights about feature relationships.

---

4. Define Features and Target 🎯

In this step, I established the target variable (Selling_Price) and the feature set (X), which included all other relevant attributes. This clear definition is crucial for training the model effectively.

---

5. Split Data into Training and Testing Sets 🔀

I partitioned the dataset into training (80%) and testing (20%) sets. This separation ensures that I can evaluate the model's performance on unseen data.

---

6. Feature Scaling ⚖️

To standardise the feature values, I utilised **StandardScaler**. This step is crucial for models like Ridge and Lasso, as they are sensitive to the scale of input data.

---

7. Model Training 🏋️

I trained both **Ridge** and **Lasso** regression models, implementing **cross-validation** to evaluate their performance on the training data. This approach allowed me to ensure the robustness of the models and mitigate overfitting.

---

8. Make Predictions and Evaluate Model 📈

I computed the **Root Mean Squared Error (RMSE)** and **R-squared (R^2)** metrics to gauge model performance:

- **Ridge RMSE**: 0.1875
- **Lasso RMSE**: 0.2480
- **Test RMSE**: 0.1584
- **Test R^2 Score**: 0.9869

These results indicated that my Ridge model explained approximately 98.69% of the variance in the selling prices, showcasing strong predictive capability.

---

9. Feature Importance and Visualization of Predictions 📊

I visualised the feature coefficients from the Ridge model, which highlighted that Present_Price had the most significant positive impact on Selling_Price, while other features exhibited negative coefficients. This insight helps to understand which features are most influential in determining car prices.

Additionally, I created visualisations comparing actual vs. predicted prices, demonstrating that the majority of predicted points aligned closely with the actual values, indicating model accuracy.

Interpretation of Results 🔍

The residual plot showed a random distribution of residuals around zero, confirming no bias in predictions. This randomness indicates that the model's errors are evenly spread, enhancing confidence in the model's reliability.

Skills Acquired:

Data Preprocessing, Feature Engineering, Model Evaluation, Data Visualisation, Machine Learning, Regression Analysis, Cross-Validation.

Hashtags:

#DataScience #MachineLearning #RidgeRegression #LassoRegression #DataPreprocessing #FeatureEngineering #ModelEvaluation #DataVisualisation #Python #Pandas #Numpy #Seaborn #Matplotlib #ModelTraining #CrossValidation #PredictiveAnalytics #DataAnalysis #Statistics #AI #CarPricePrediction