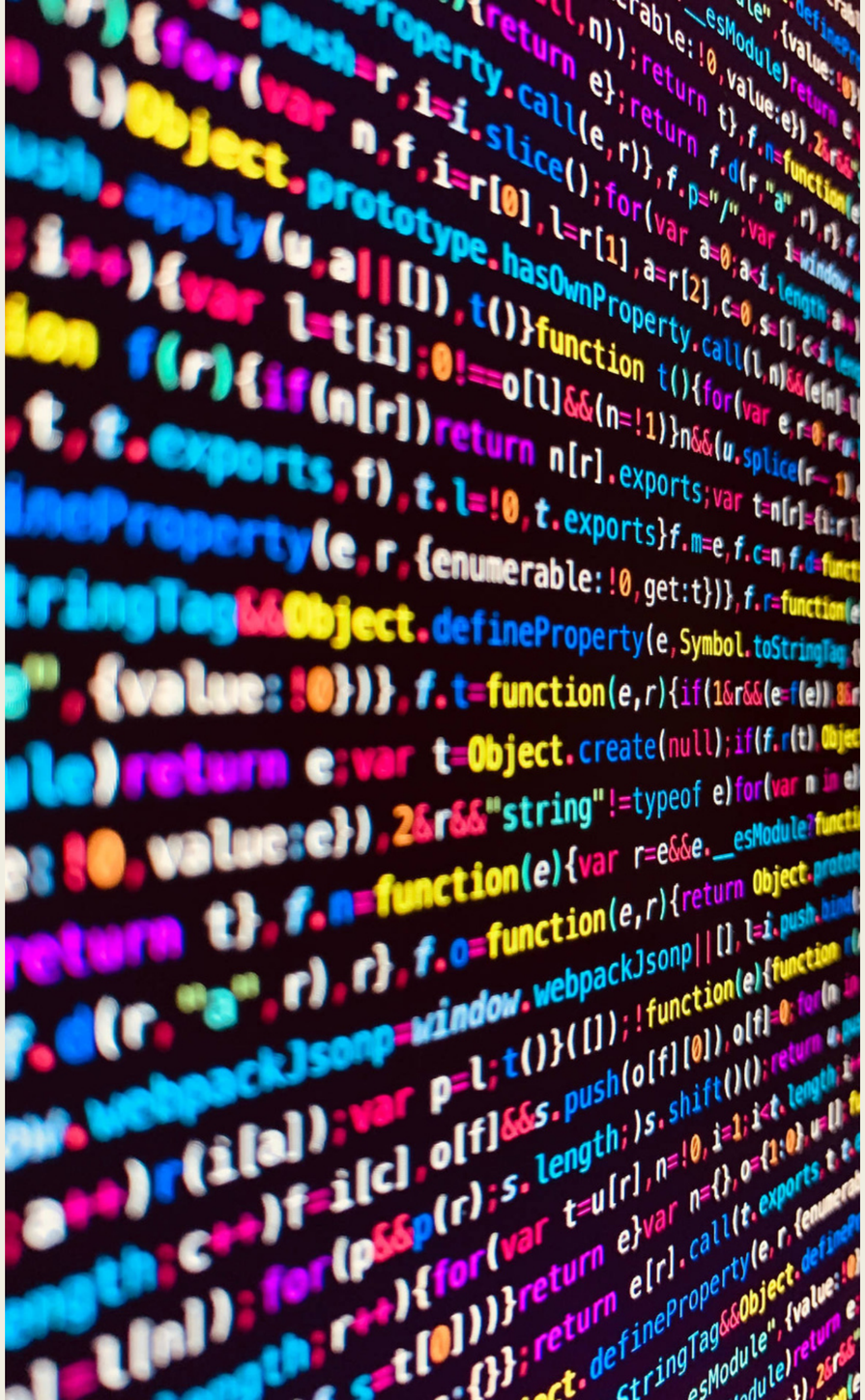
CAR SALES DATASET

TEAM SPEC

Histre Matéo
MMN3

Krika Camila
DIA4



24/12/23
Krika Camila
Histre Matéo

WORKFLOW

1. Introduction

2. Data Preprocessing

3. Data Visualization

4. Data Modeling

5. Comparative Analysis

24/12/23
Krika Camila
Histre Matéo

"HOW CAN MACHINE LEARNING IMPROVE CAR PRICE PREDICTION USING A DIVERSE DATASET IN THE COMPETITIVE AUTOMOTIVE MARKET?"

INTRODUCTION

The automotive market is a sector embroiled in fierce competition. Consequently, we have undertaken a machine learning project focused on predicting car prices. Leveraging a comprehensive dataset encompassing sales data from various brands, our objective is to develop a robust model that accurately forecasts the pricing dynamics within the automobile industry.

24/12/23
Krika Camila
Histre Matéo

OVERVIEW

1. Data Loading
2. Data Cleaning
3. Data pre-processing: Imputation of missing data
4. Data pre-processing: Normalizing Data
5. Data pre-processing: Data Encoding

24/12/23
Krika Camila
Histre Matéo

6

100 rows | 100 rows × 16 columns

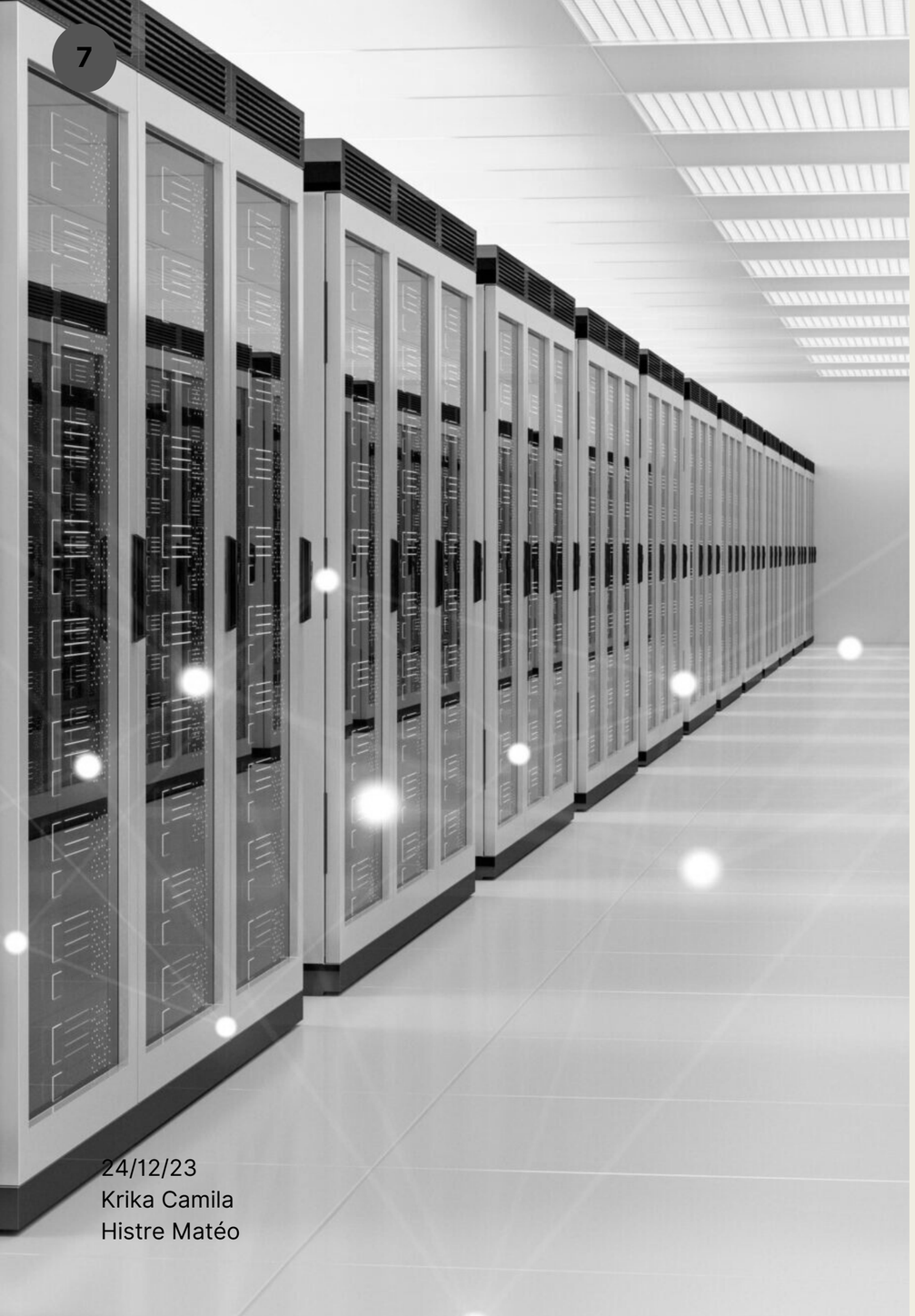
	anufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length
0	Acura	Integra	16.919	16.360	Passenger	21.500	1.8	140	101.2	67.3	172.4
1	Acura	TL	39.384	19.875	Passenger	28.400	3.2	225	108.1	70.3	192.9
3	Acura	RL	8.588	29.725	Passenger	42.000	3.5	210	114.6	71.4	196.6
4	Audi	A4	20.397	22.255	Passenger	23.990	1.8	150	102.6	68.2	178.0
5	Audi	A6	18.780	23.555	Passenger	33.950	2.8	200	108.7	76.1	192.0
6	Audi	A8	1.380	39.000	Passenger	62.000	4.2	310	113.0	74.0	198.2
7	BMW	323i	19.747	NaN	Passenger	26.990	2.5	170	107.3	68.4	176.0
8	BMW	328i	9.231	28.675	Passenger	33.400	2.8	193	107.3	68.5	176.0
9	BMW	528i	17.527	36.125	Passenger	38.900	2.8	193	111.4	70.9	188.0
10	Buick	Century	91.561	12.475	Passenger	21.975	3.1	175	109.0	72.7	194.6
11	Buick	Regal	39.350	13.740	Passenger	25.300	3.8	240	109.0	72.7	196.2
12	Buick	Park Avenue	27.851	20.190	Passenger	31.965	3.8	205	113.8	74.7	206.8
13	Buick	LeSabre	83.257	13.360	Passenger	27.885	3.8	205	112.2	73.5	200.0
14	Cadillac	DeVille	63.729	22.525	Passenger	39.895	4.6	275	115.3	74.5	207.2
16	Cadillac	Eldorado	6.536	25.725	Passenger	39.665	4.6	275	108.0	75.5	200.6
17	Cadillac	Catera	11.185	18.225	Passenger	31.010	3.0	200	107.4	70.3	194.8
18	Cadillac	Escalade	14.785	NaN	Car	46.225	5.7	255	117.5	77.0	201.2
19	Chevrolet	Cavalier	145.519	9.250	Passenger	13.260	2.2	115	104.1	67.9	180.9

DATA LOADING

24/12/23
Krika Camila
Histre Matéo

Python csv loading

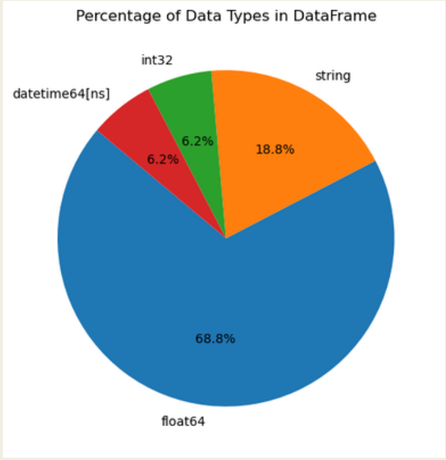
In the realm of data analysis with Python, efficient CSV data loading is a fundamental skill. Leveraging the versatility of the Pandas library, Python enthusiasts can seamlessly handle CSV datasets for analysis. With a concise code snippet, users can effortlessly read a CSV file into a Pandas DataFrame, providing an immediate glimpse of the dataset's structure. Beyond basic loading, Pandas offers robust mechanisms for handling challenges such as missing data, specifying data types for optimization, and efficiently managing large datasets through techniques like chunking. Python's prowess in data manipulation extends to advanced libraries like Dask and NumPy, ensuring that users have a comprehensive toolkit for diverse data loading needs. The simplicity and power of Python make it a preferred choice for data professionals navigating the intricacies of CSV data in their analytical endeavors.



DATA PRE-PROCESSING: DATA CLEANING

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Manufacturer         157 non-null    object
 1   Model                157 non-null    object
 2   Sales_in_thousands   157 non-null    object
 3   __year_resale_value  121 non-null    object
 4   Vehicle_type         157 non-null    object
 5   Price_in_thousands   155 non-null    object
 6   Engine_size          156 non-null    object
 7   Horsepower           156 non-null    object
 8   Wheelbase            156 non-null    object
 9   Width                156 non-null    object
 10  Length               156 non-null    object
 11  Curb_weight          155 non-null    object
 12  Fuel_capacity        156 non-null    object
 13  Fuel_efficiency      154 non-null    object
 14  Latest_Launch        157 non-null    object
 15  Power_perf_factor    155 non-null    object
dtypes: object(16)
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 152 entries, 0 to 156
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Manufacturer         152 non-null    string
 1   Model                152 non-null    string
 2   Sales_in_thousands   152 non-null    float64
 3   __year_resale_value  152 non-null    float64
 4   Vehicle_type         152 non-null    string
 5   Price_in_thousands   152 non-null    float64
 6   Engine_size          152 non-null    float64
 7   Horsepower           152 non-null    int32
 8   Wheelbase            152 non-null    float64
 9   Width                152 non-null    float64
 10  Length               152 non-null    float64
 11  Curb_weight          152 non-null    float64
 12  Fuel_capacity        152 non-null    float64
 13  Fuel_efficiency      152 non-null    float64
 14  Latest_Launch        0 non-null      datetime64[ns]
 15  Power_perf_factor    152 non-null    float64
dtypes: datetime64[ns](1), float64(11), int32(1), string(3)
memory usage: 19.6 KB
```



24/12/23
Krika Camila
Histre Matéo

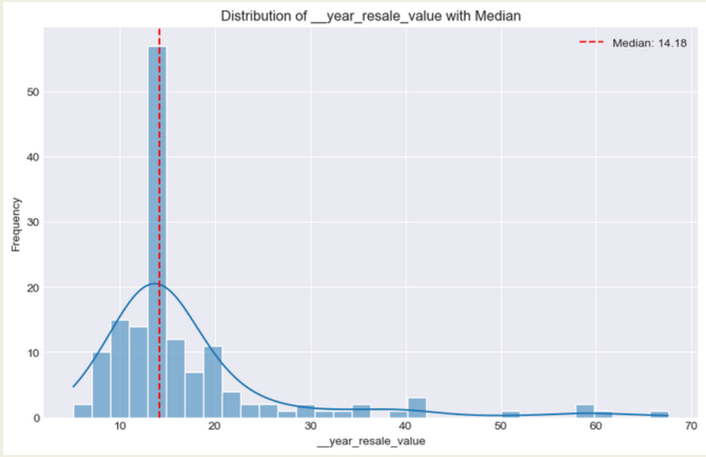
DATA PRE-PROCESSING: IMPUTATION OF MISSING DATA

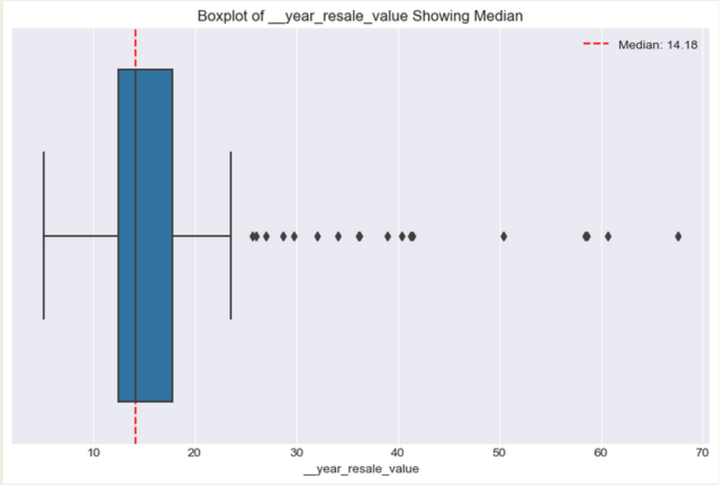
	0
Manufacturer	0
Model	0
Sales_in_thousands	0
__year_resale_value	36
Vehicle_type	0
Price_in_thousands	2
Engine_size	1
Horsepower	1
Wheelbase	1
Width	1
Length	1
Curb_weight	2
Fuel_capacity	1
Fuel_efficiency	3
Latest_Launch	0
Power_perf_factor	2

dtype: int64

	0
Manufacturer	0
Model	0
Sales_in_thousands	0
__year_resale_value	0
Vehicle_type	0
Price_in_thousands	0
Engine_size	0
Horsepower	0
Wheelbase	0
Width	0
Length	0
Curb_weight	0
Fuel_capacity	0
Fuel_efficiency	0
Latest_Launch	0
Power_perf_factor	0

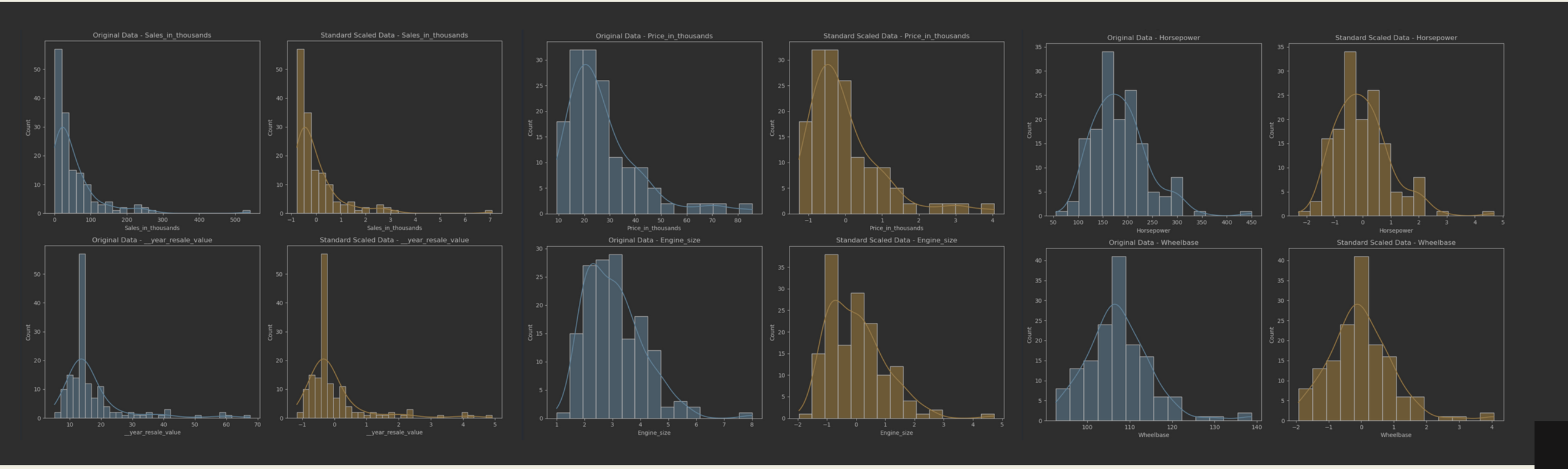
dtype: int64





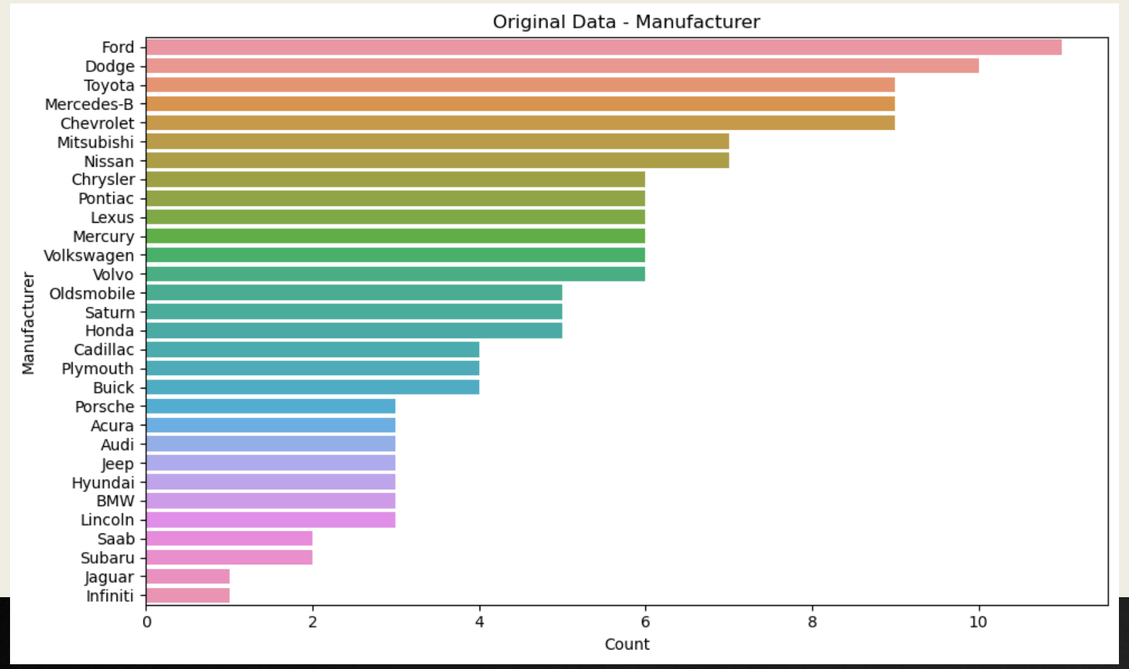
24/12/23
Krika Camila
Histre Matéo

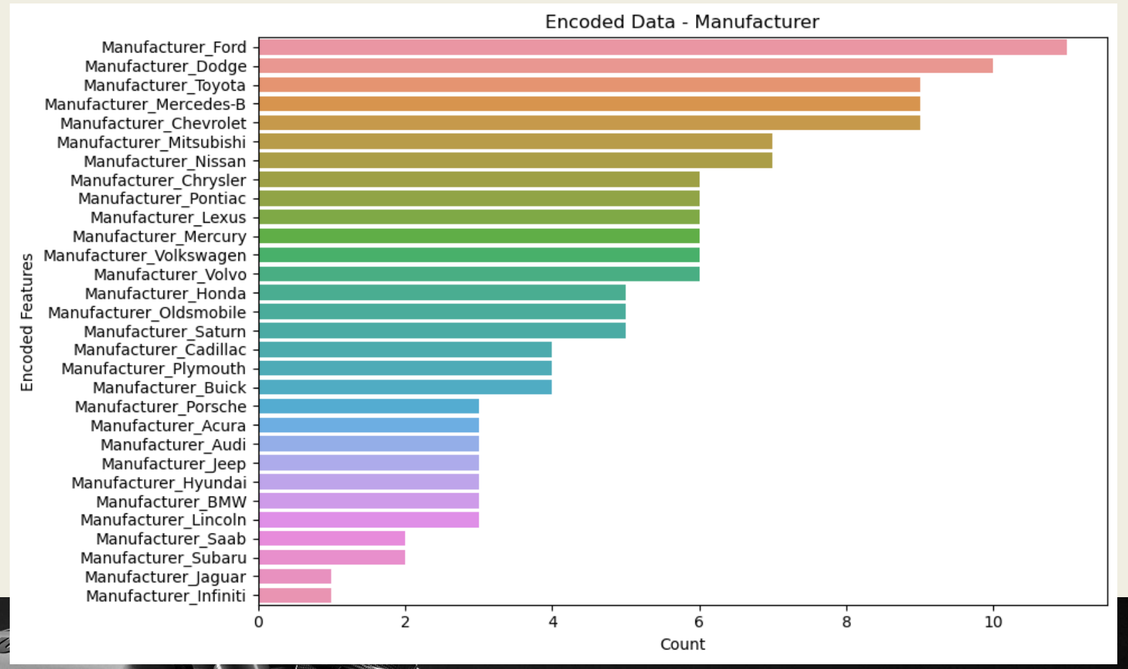
DATA PRE-PROCESSING: NORMALIZING DATA

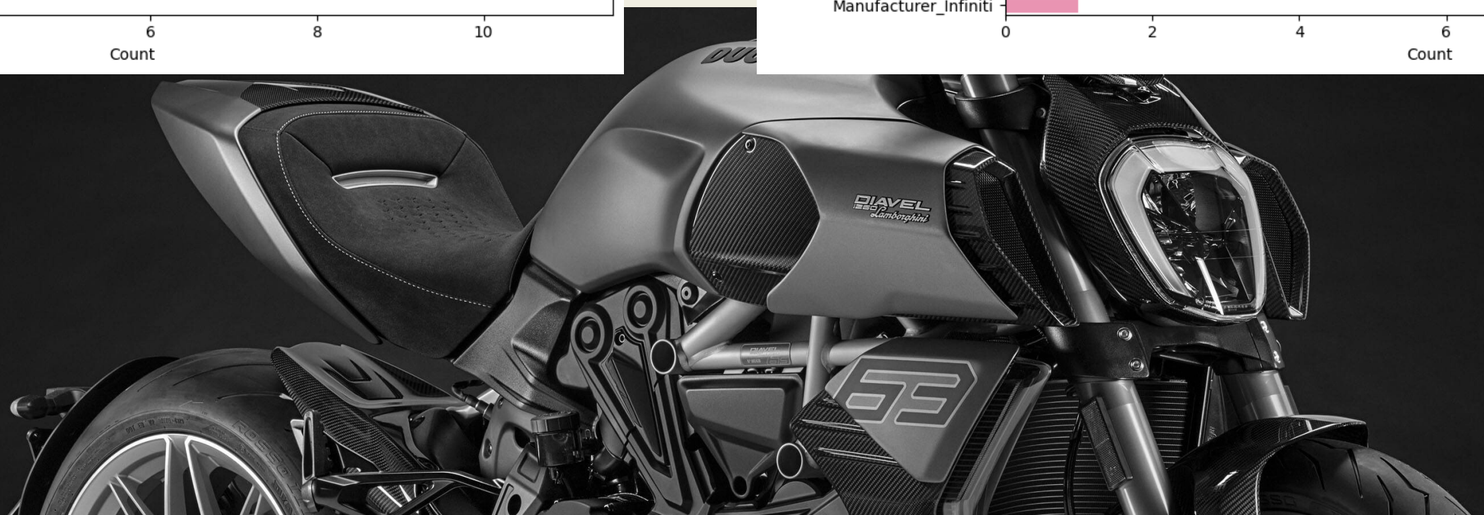


24/12/23
Krika Camila
Histre Matéo

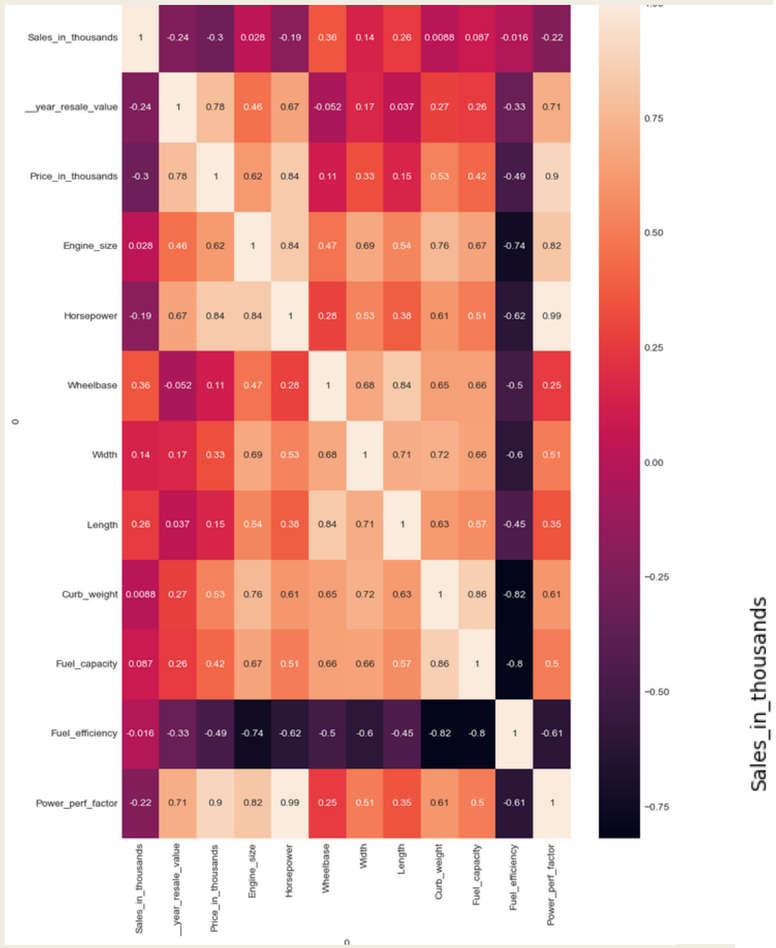
DATA PRE-PROCESSING: DATA ENCODING

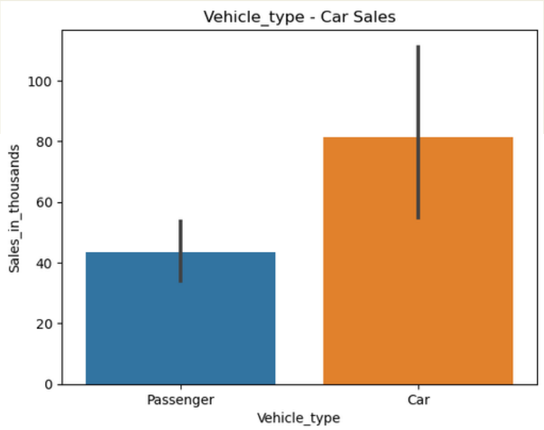


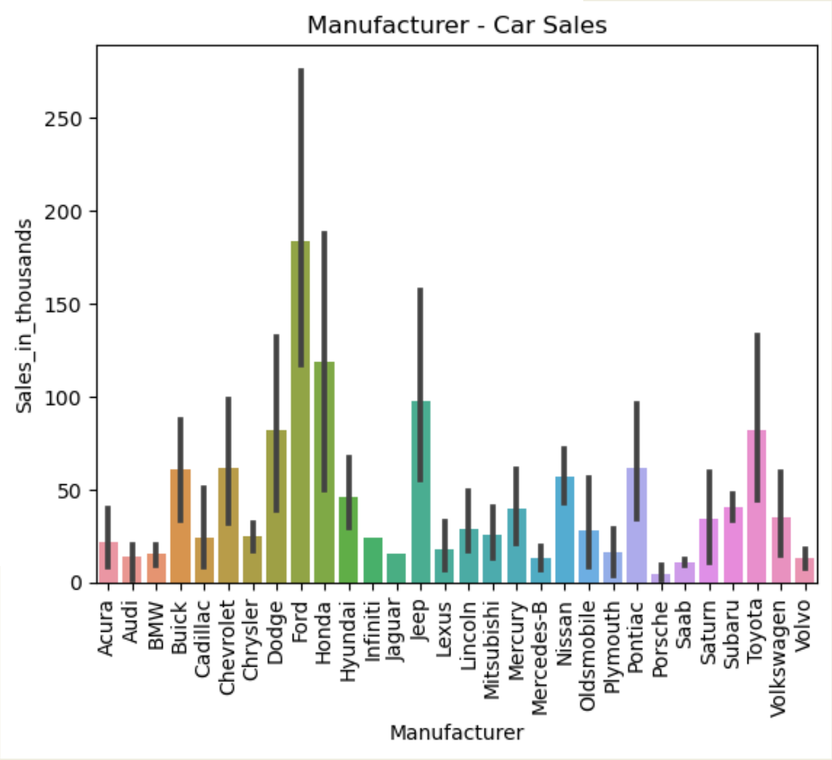




24/12/23
Krika Camila
Histre Matéo



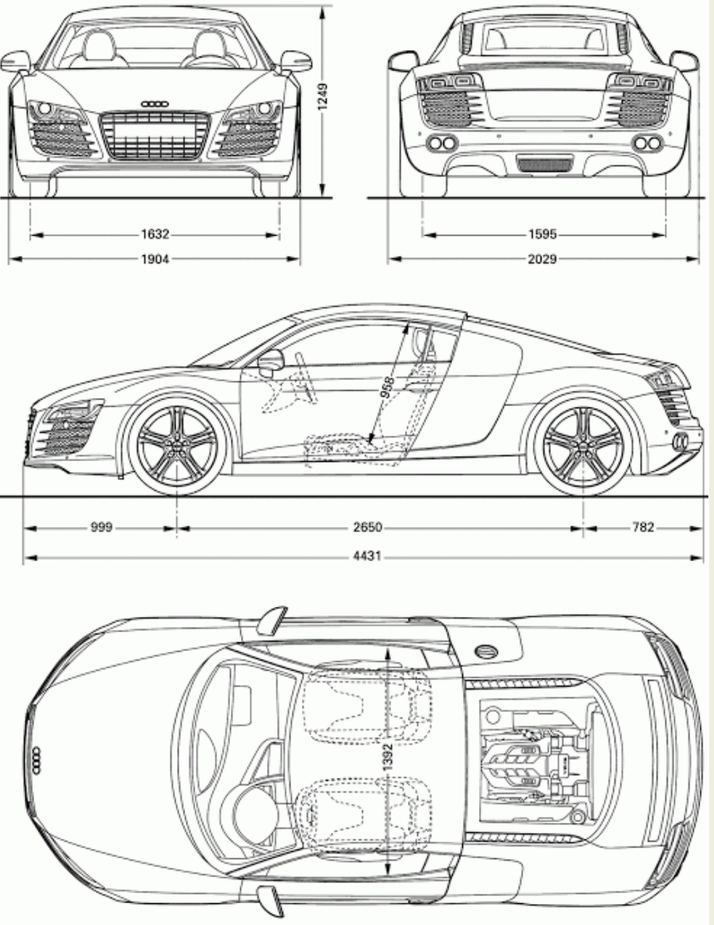




DATA VISUALIZATION

Data visualization is a powerful tool for unraveling the insights hidden within a dataset. By representing complex information in visual formats such as charts, graphs, and maps, data visualization transforms raw data into a comprehensible and meaningful narrative. It allows analysts, researchers, and decision-makers to quickly grasp patterns, trends, and outliers, facilitating more informed and strategic decision-making. Visualization not only simplifies the communication of findings but also enhances the ability to identify correlations and relationships that might be challenging to discern in tabular or textual formats. Ultimately, data visualization serves as a bridge between data and understanding, enabling more effective exploration and interpretation of datasets across various domains, from business analytics to scientific research.

24/12/23
Krika Camila
Histre Matéo



DATA MODELING

We aim to predict the "price_in_thousands," our designated target variable in the dataset. Employing the scikit-learn library, we experiment with various algorithms, fine-tune hyperparameters, conduct grid searches, and analyze model performance through graphical comparisons.

The algorithms under consideration include:

1. RandomForestRegressor
2. SVR (Support Vector Regressor)

For each model, we will evaluate and compare the following metrics:

1. R2 (Coefficient of Determination)
2. RMSE (Root Mean Square Error)

24/12/23
Krika Camila
Histre Matéo

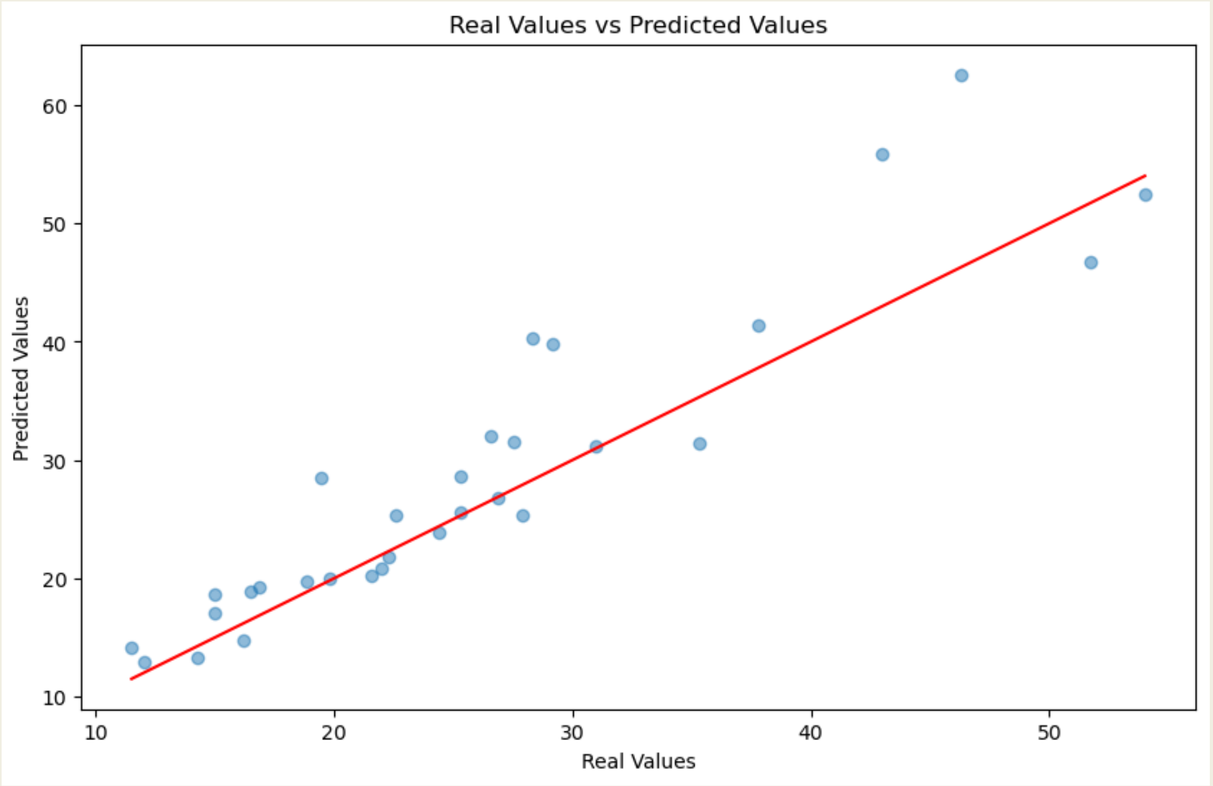
RANDOM FOREST REGRESSOR

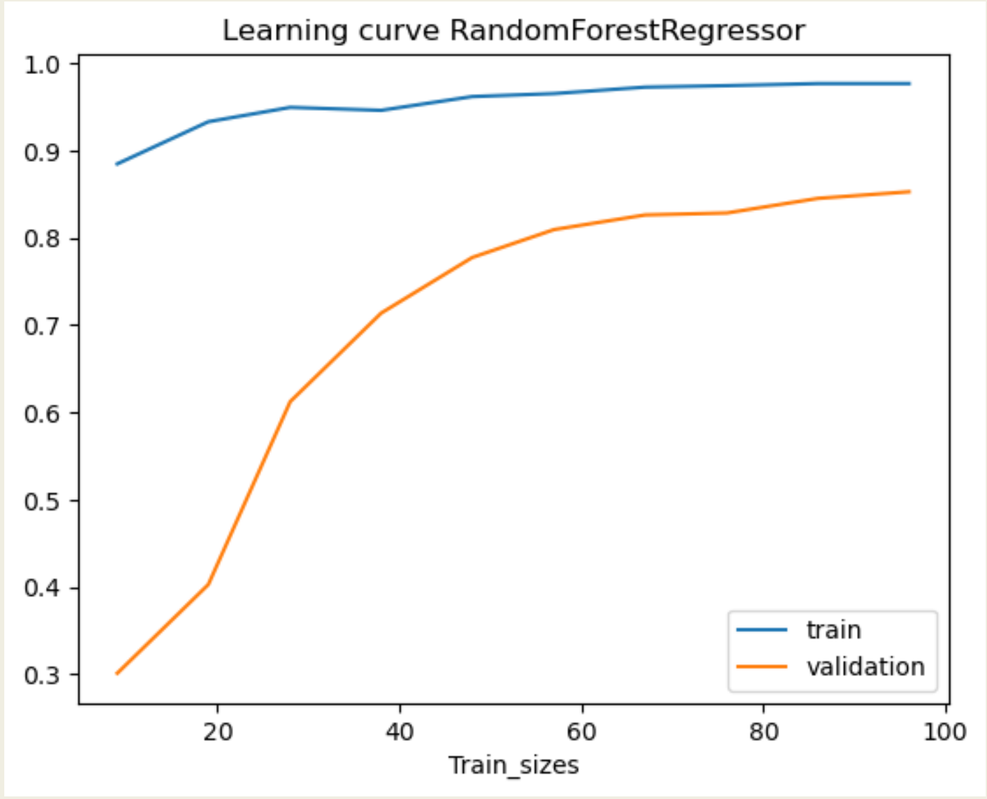
Grid Search :

```
Best hyperparameters: {'max_depth': 30, 'n_estimators': 100}
```

Regression Metrics:

```
Root Mean Squared Error (RandomForestRegressor): 5.500149292121607
R-squared (RandomForestRegressor): 0.7442740374437234
```





24/12/23
Krika Camila
Histre Matéo

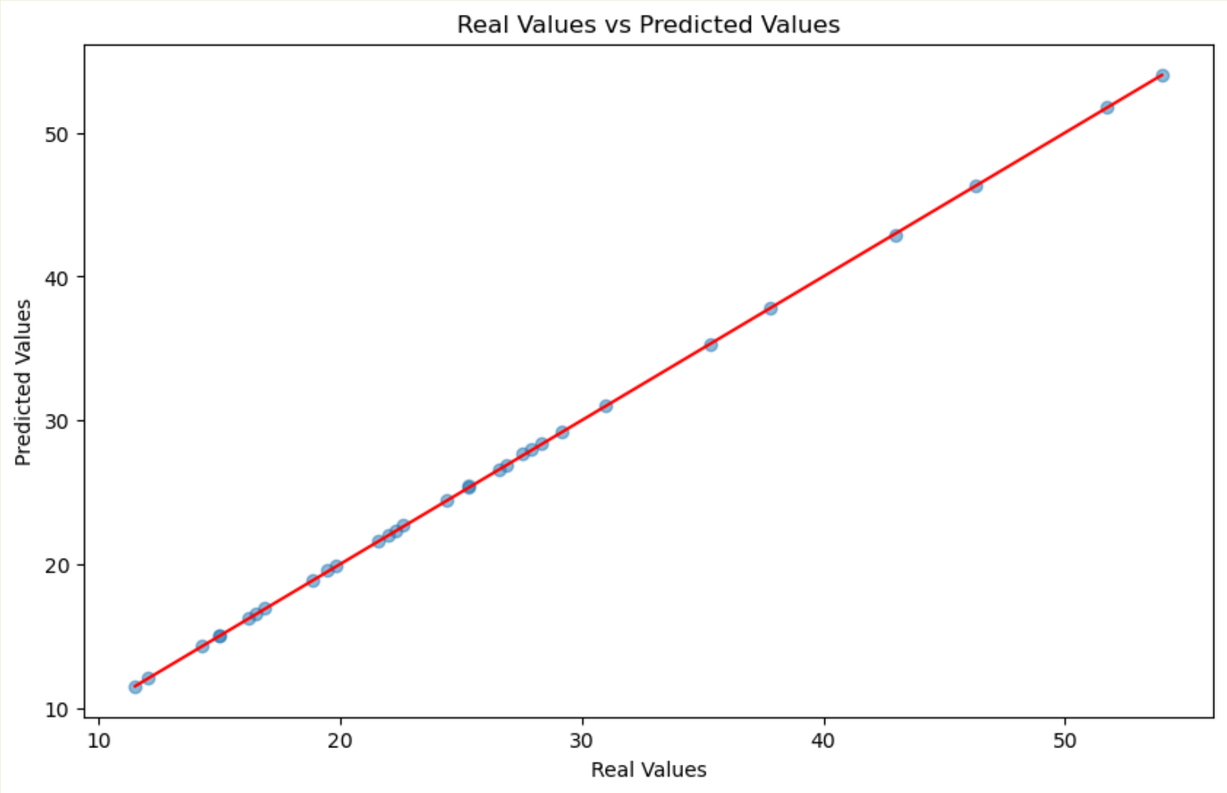
SVR

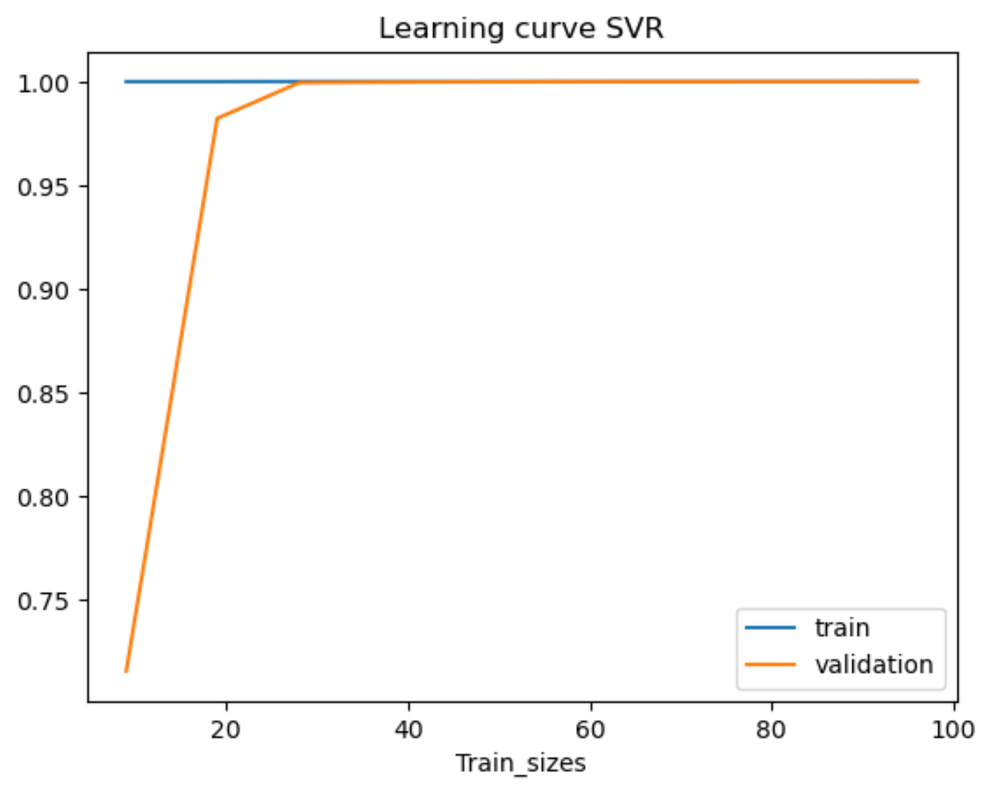
Grid Search :

```
Best hyperparameters SVR: {'C': 1, 'gamma': 'scale', 'kernel': 'linear'}
```

Regression Metrics:

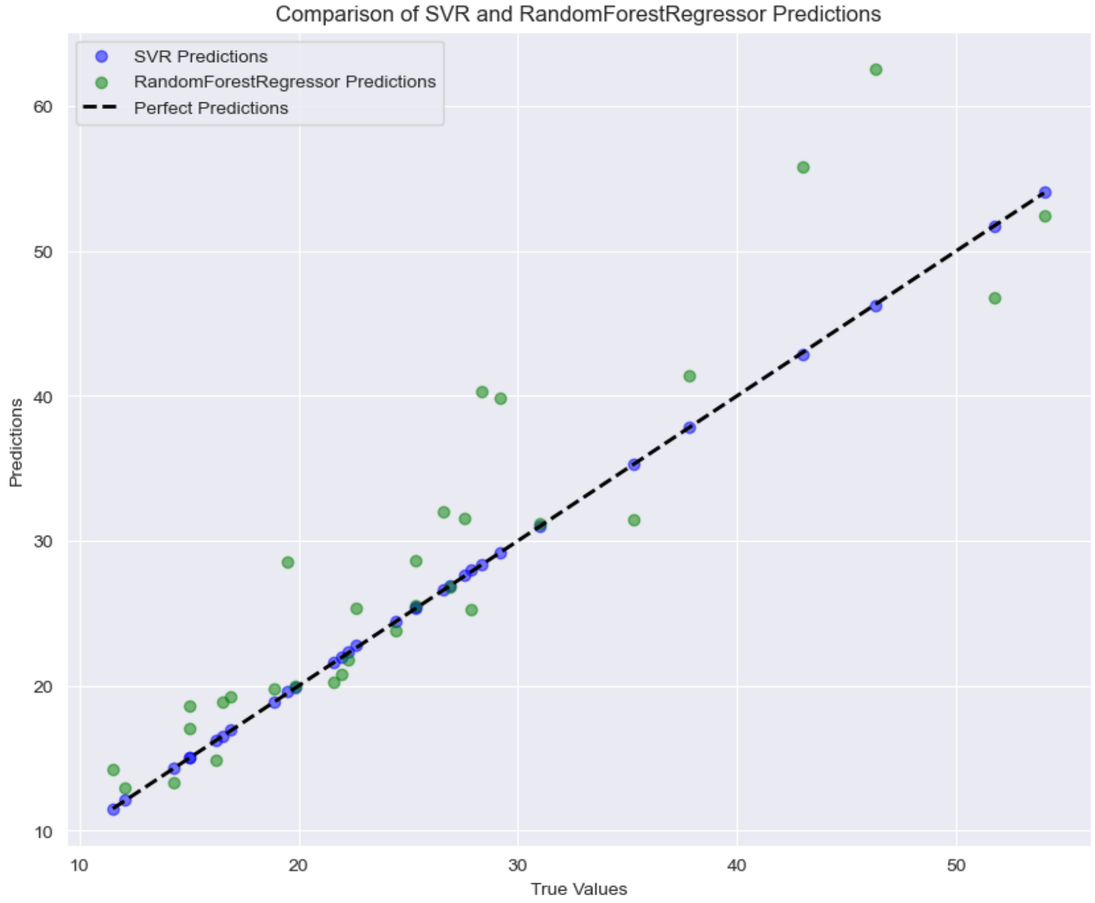
```
Root Mean Squared Error (SVR): 0.06883855913188511
R-squared (SVR): 0.9999599419864361
```

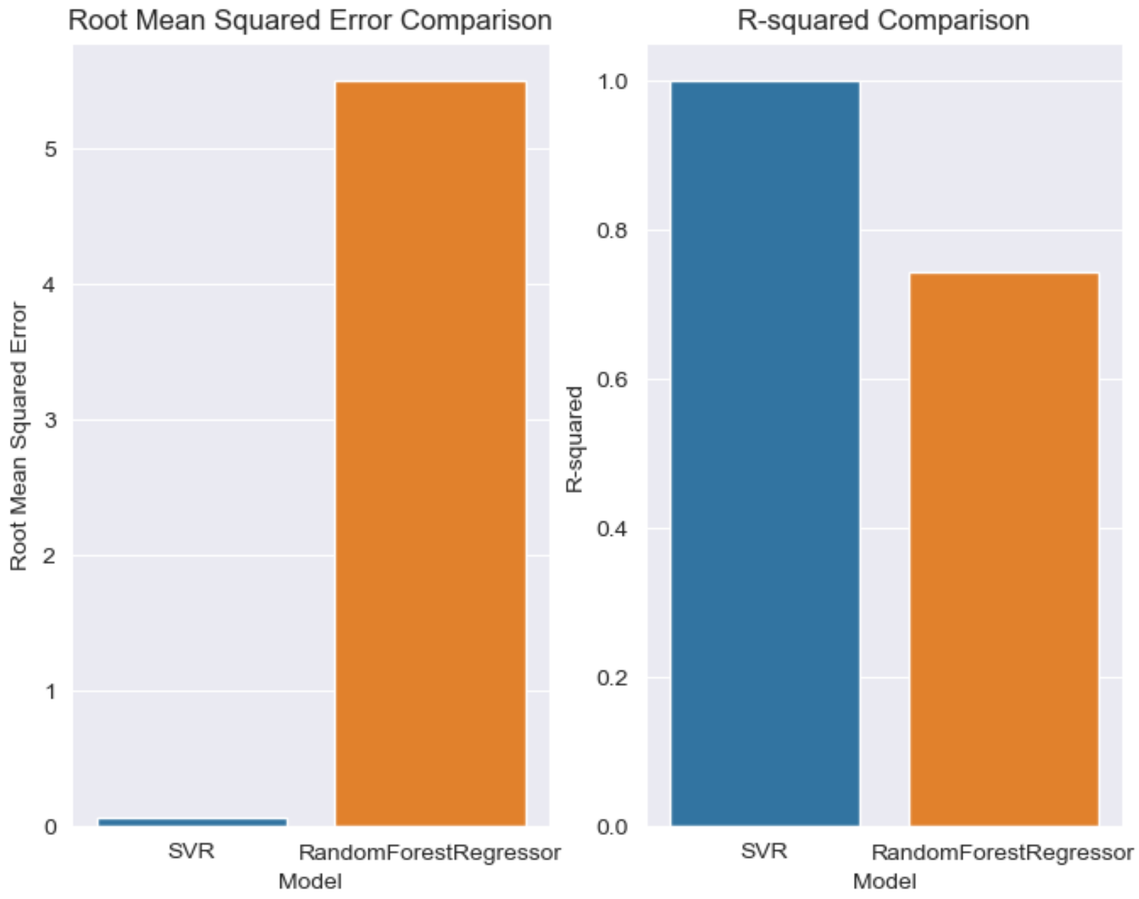




24/12/23
Krika Camila
Histre Matéo

COMPARATIVE ANALYSIS OF MODEL PERFORMANCE METRICS

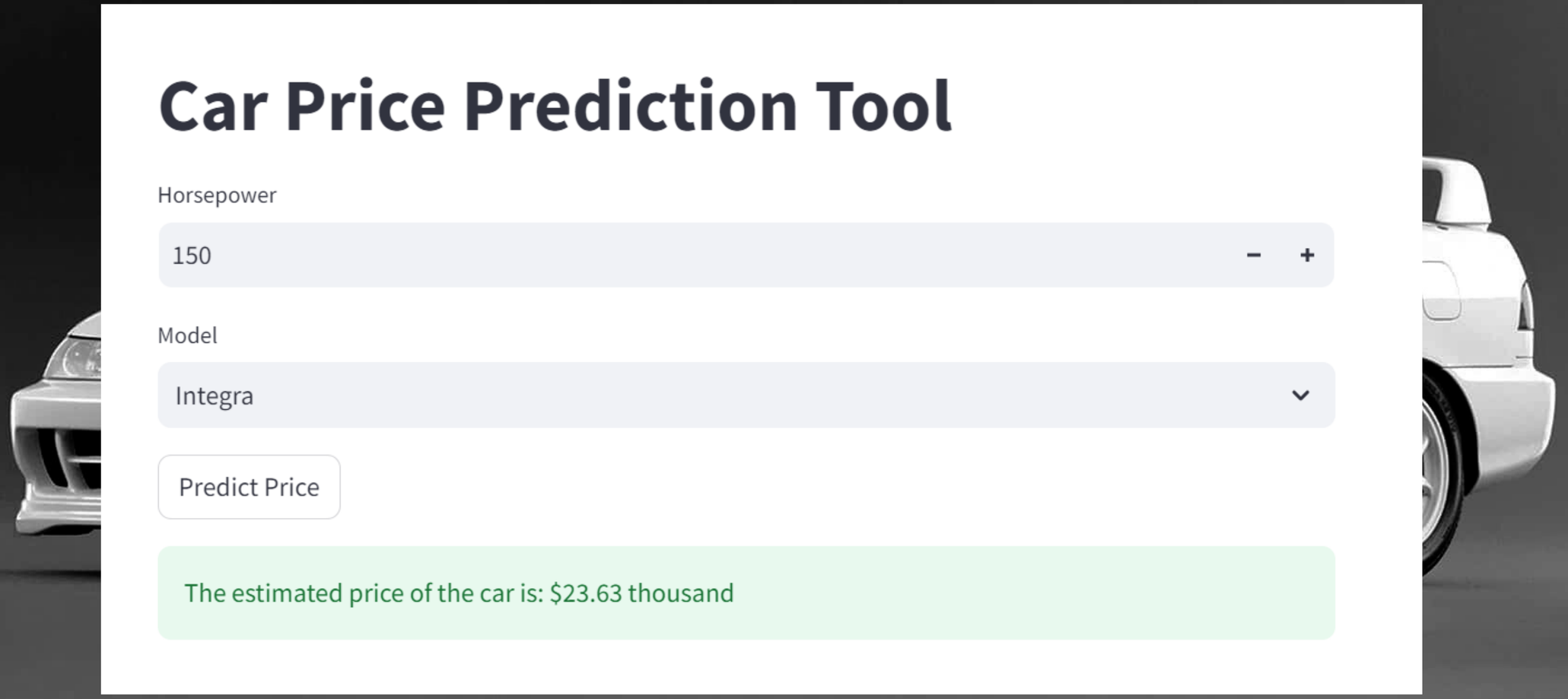




24/12/23
Krika Camila
Histre Matéo

Based on the visualizations, the **SVR** model appears to outperform the Random Forest Regressor in terms of accuracy and the ability to capture the variance of the test data. The predictions from the **SVR** are closer to the actual values and exhibit a lower mean squared error, as indicated by a lower RMSE and an R^2 closer to 1. This suggests that SVR might be the preferred model for this specific dataset.

BONUS



24/12/23
Krika Camila
Histre Matéo

THANK YOU

24/12/23
Krika Camila
Histre Matéo