WE LEAD
Code.Hub

FEBRUARY 2025

Predicting the City Cycle Fuel Consumption in mpg of a Car

A Classification Problem

Team 2

Alexaki Erofili
Diamanti Ioli
Karagianni Ioanna
Neokosmidou Simoni Maria
Charana Aikaterini


MADE WITH
KAPWING

WE LEAD
Code.Hub

Contents

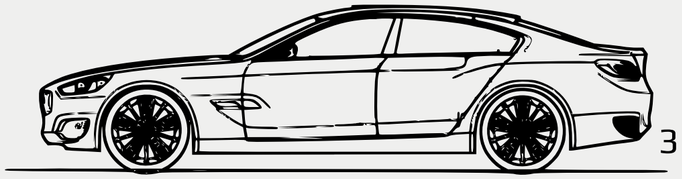
1 **Goal**
Finding the best classification model that would predict the fuel efficiency categories of different cars, in order to assist a car rental company to make data - driven fleet decisions.

2 **Exploratory Data Analysis**
Tried out different data analysis techniques for understanding our data.

3 **Preprocessing**
Cleaning and preproccesing the dataset before applying classification algorithms.

4 **Classification Algorithm Development**
Worked with 8 classification models, as we aimed to explore different algorithms and compare them based on performance.

5 **Results and Discussion**



WE LEAD
Code.Hub

Exploratory Data Analysis

Overview & Basic information about the dataset

	mpg	cylinders	displayments	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.0	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	1	ford torino

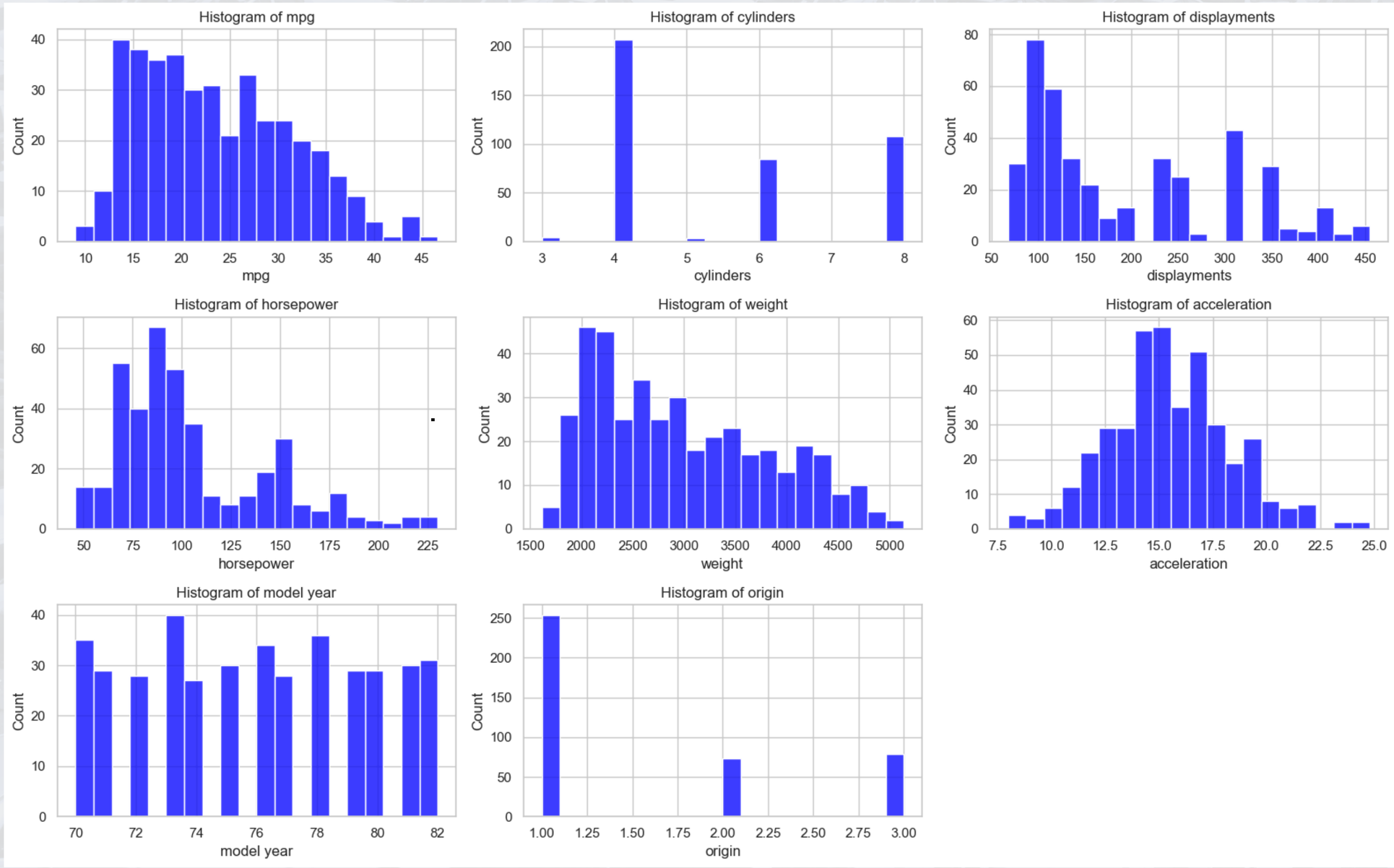
1.1. Dataset

```
mpg             8
cylinders       0
displayments    0
horsepower      6
weight          0
acceleration    0
model year      0
origin          0
car name        0
dtype: int64
```

1.2. Missing Values

- 406 instances
- 8 features & 1 target
- Detecting N/As & duplicates
- Descriptive statistics
- Range and value counts for each column

WE LEAD
Code.Hub

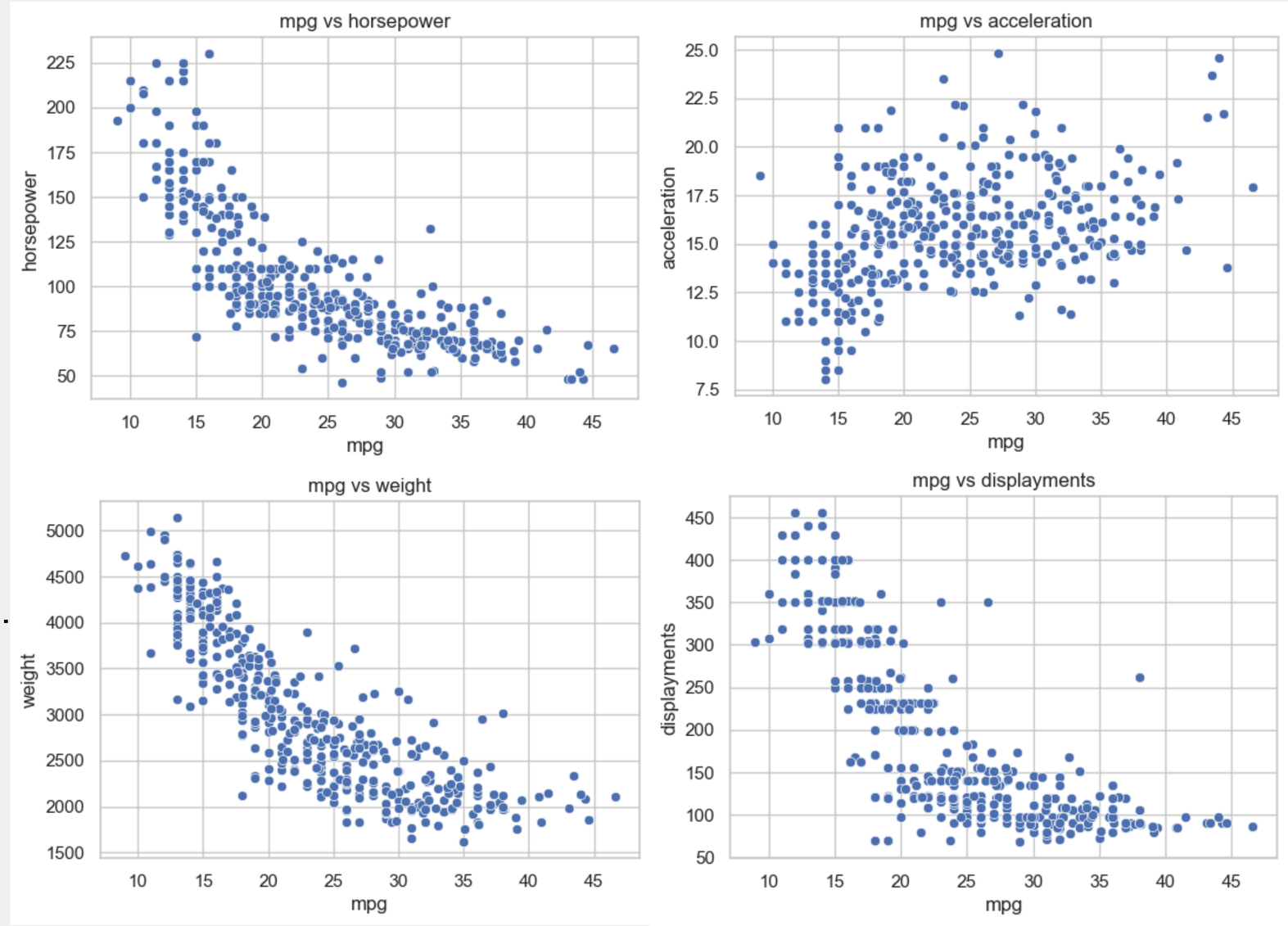


1.3. Histograms of counts in the variables

WE LEAD
Code.Hub

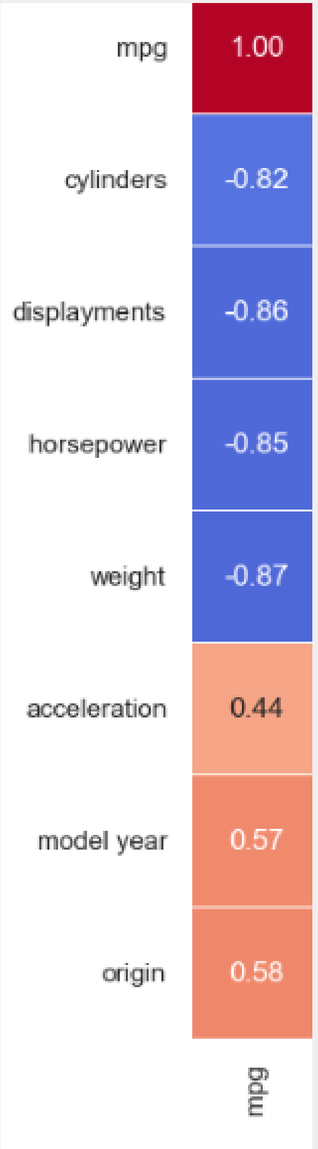
- The relationships between "mpg" with "displayments", "horsepower", "weight", are **strong negative monotonic.**

- Between the "mpg" and "acceleration" there seems to be a **weak positive monotonic** trend, although the scatter plot shows significant variability.

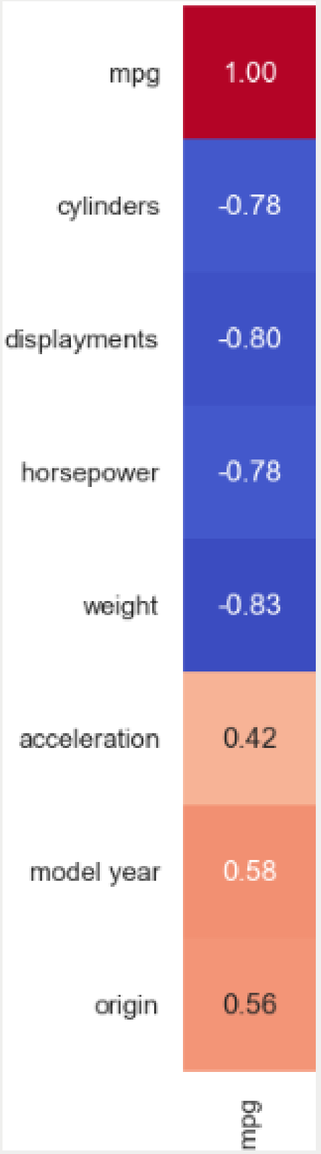


1.4. Scatterplots for defining relationships

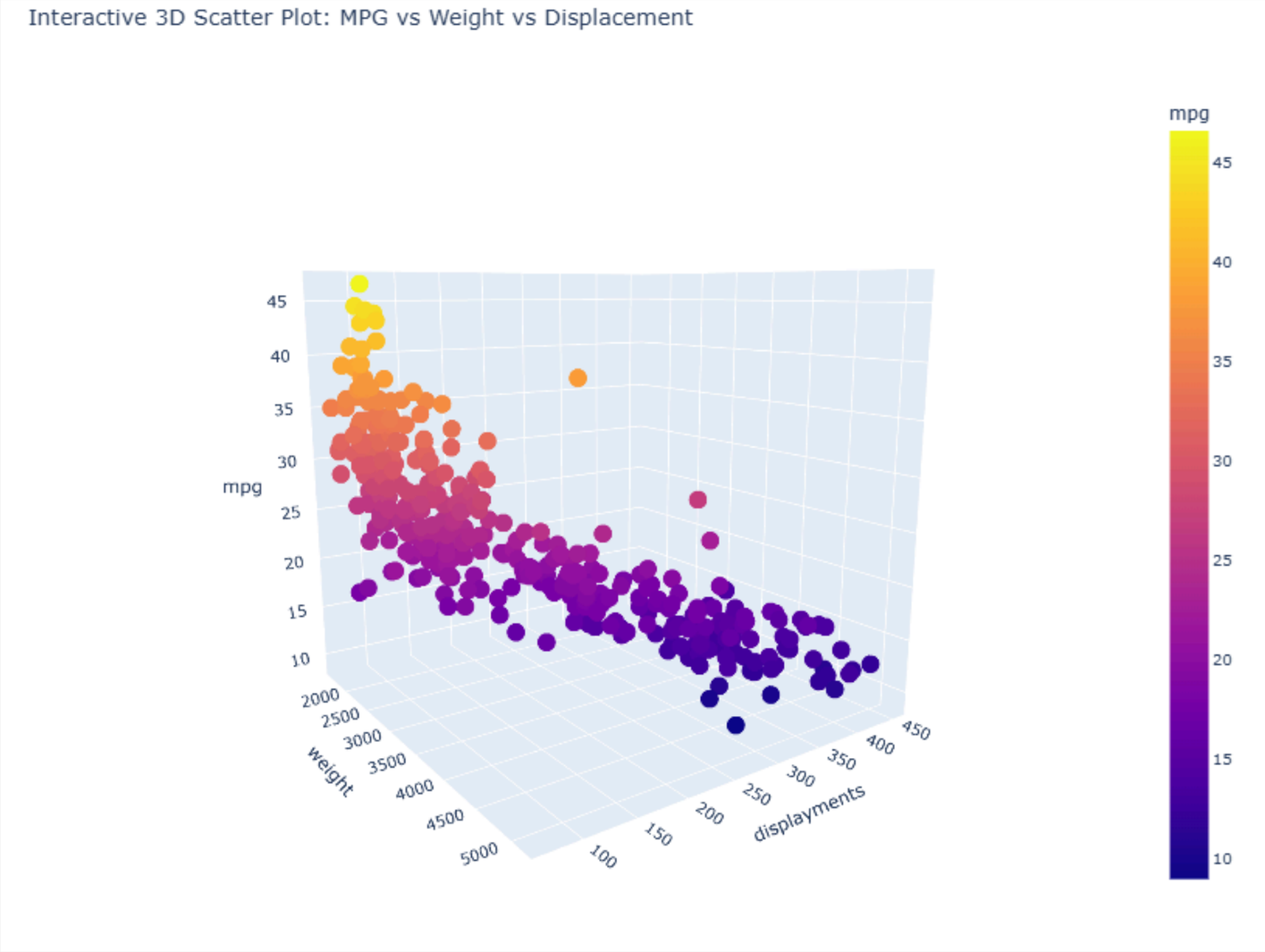
WE LEAD
Code.Hub



1.5. Spearman (monotonous) correlation between the variables



1.6. Pearson (linear) correlation between the variables

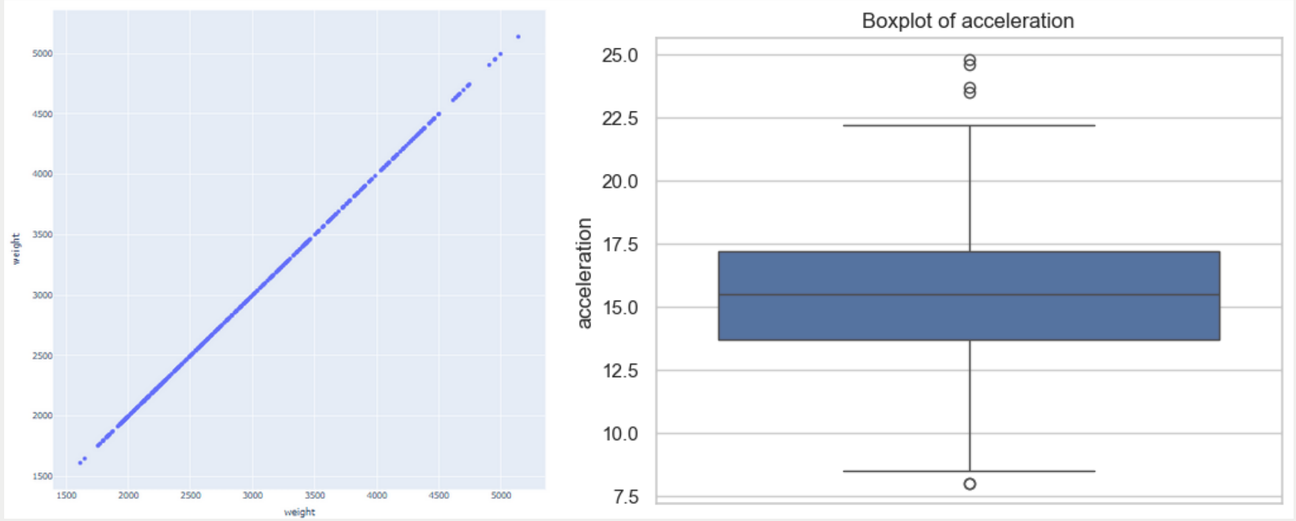


1.7. 3D correlation between mpg/weight/displacements

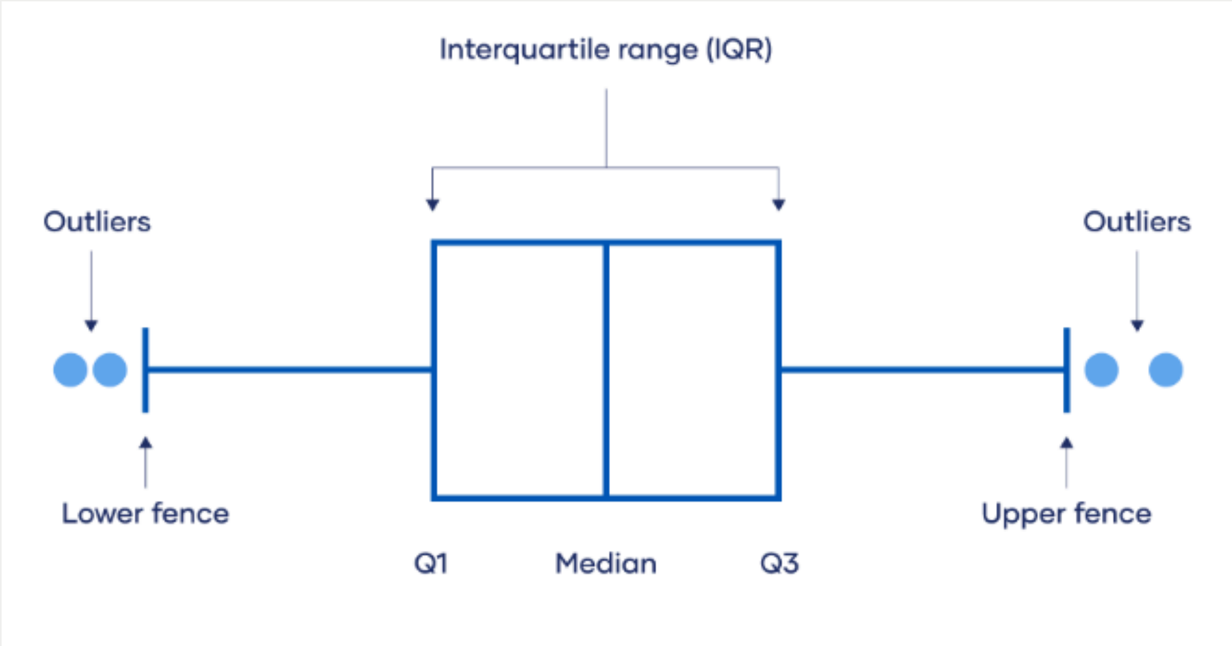
- **Lightweight Cars with Small Engines** (left side, lower displacement, red points): These cars tend to have high MPG values (fuel-efficient).
- **Heavyweight Cars with Large Engines** (right side, higher displacement, blue points): These cars are less fuel-efficient, as seen by their low MPG.

WE LEAD
Code.Hub

Detecting possible outliers



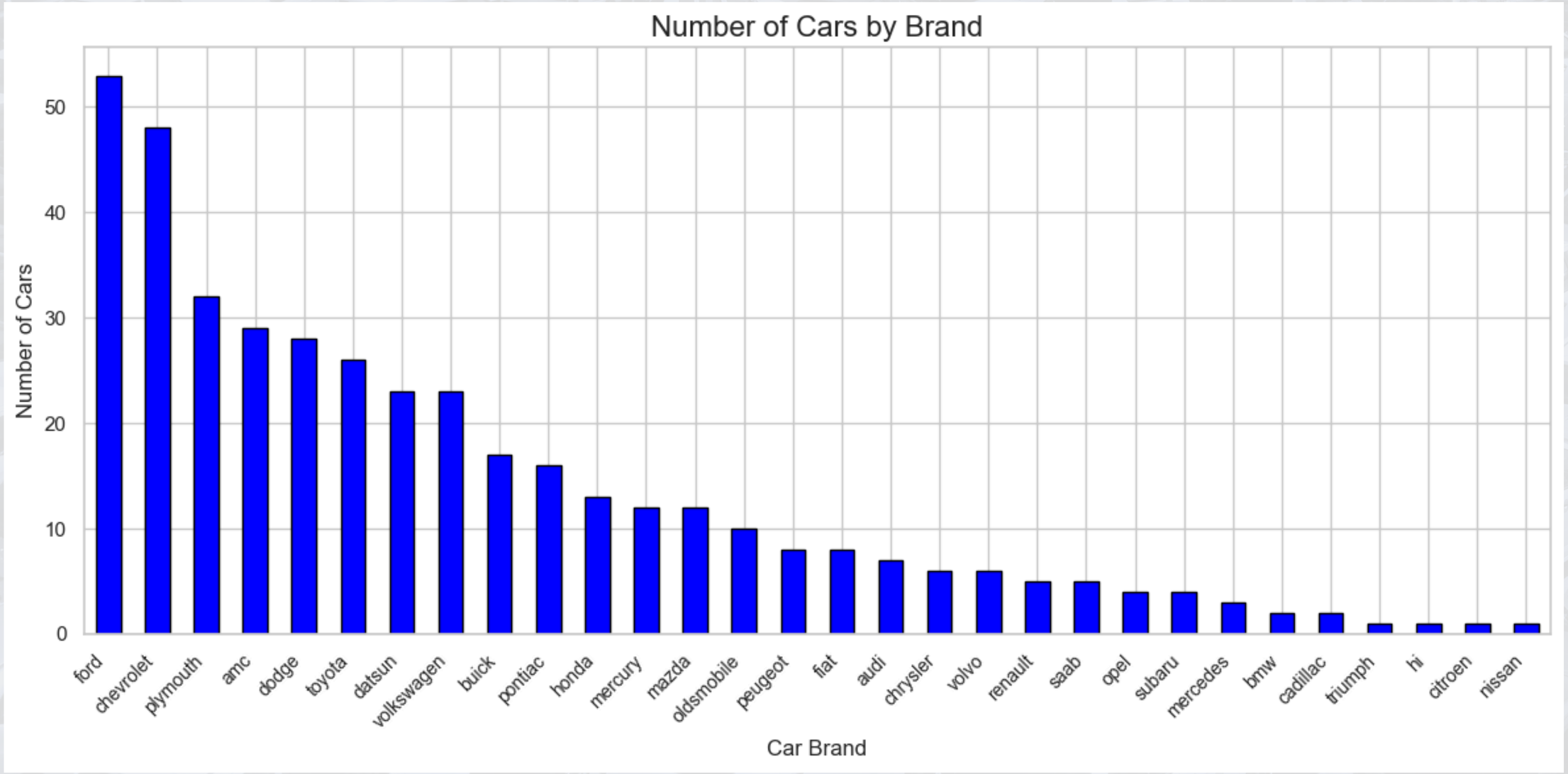
1.8. Scatterplots &Boxplots for detecting outliers



1.9. IQR method for collecting outliers

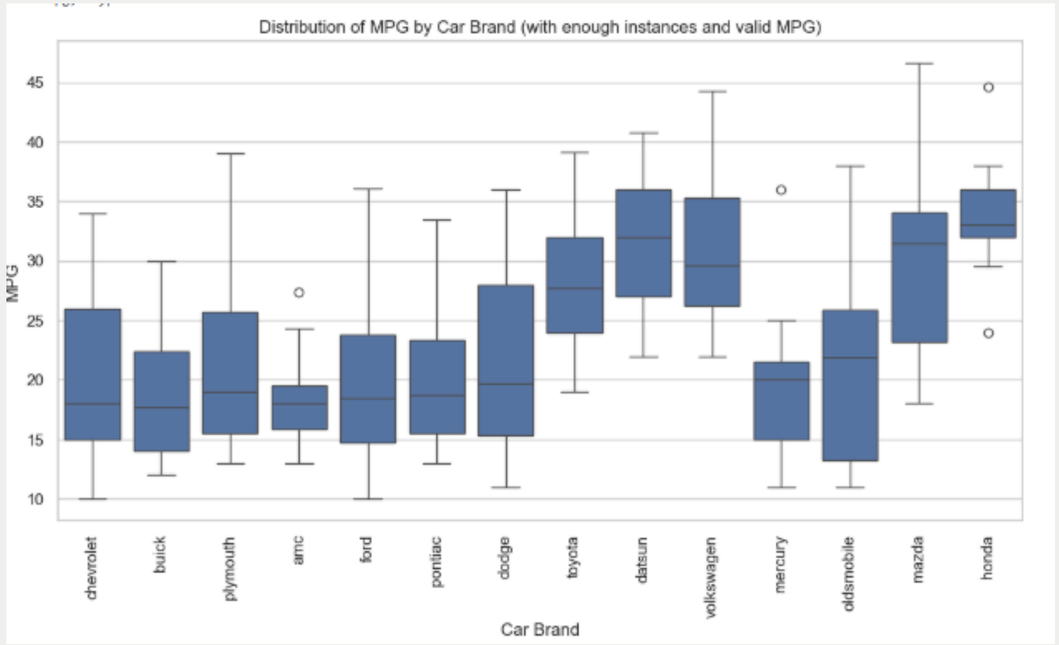
15 outliers

	mpg	cylinders	displayments	horsepower	weight	acceleration
6	14.0	8	454.0	220.0	4354	9.0
7	14.0	8	440.0	215.0	4312	8.5
8	14.0	8	455.0	225.0	4425	10.0
16	14.0	8	340.0	160.0	3609	8.0
17	NaN	8	302.0	140.0	3353	8.0
19	14.0	8	455.0	225.0	3086	10.0
31	10.0	8	360.0	215.0	4615	14.0
66	23.0	4	97.0	54.0	2254	23.5
101	13.0	8	440.0	215.0	4735	11.0
102	12.0	8	455.0	225.0	4951	11.0
123	16.0	8	400.0	230.0	4278	9.5
306	27.2	4	141.0	71.0	3190	24.8
329	46.6	4	86.0	65.0	2110	17.9
333	43.4	4	90.0	48.0	2335	23.7
402	44.0	4	97.0	52.0	2130	24.6

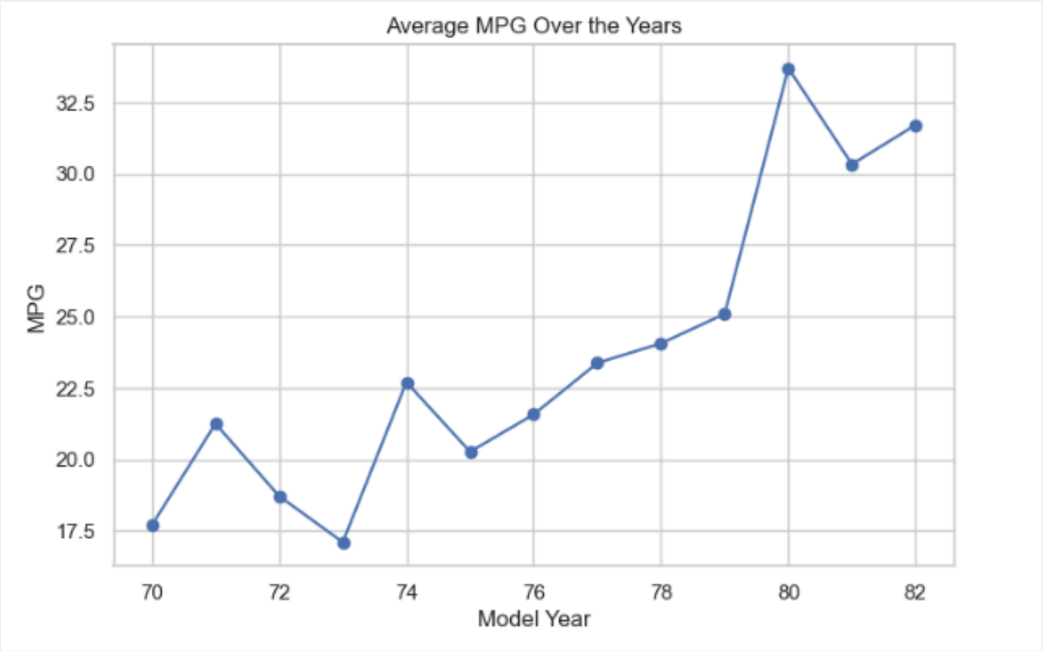


1.10. Analysis of Categorical Values: Counts of different car brands

WE LEAD
Code.Hub

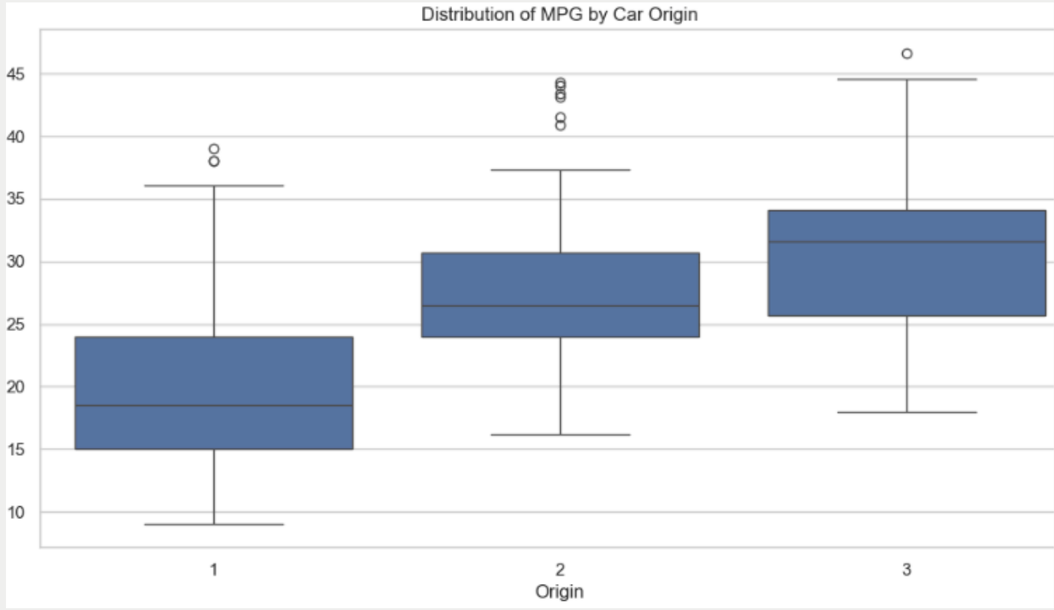


1.12. MPG by car brands



1.11. Average MPG over the years

- **Volkswagen, Mazda, Datsun and Honda** showing the highest average MPG values
- Brands like **Ford, Mercury, Buick, and Amc** tend to have lower MPG values.
- Increasing MPG trend over the years (1.11)
- **Origin 3 (Asia) likely corresponds to more fuel-efficient cars on average**, while **Origin 1 (USA) cars tend to have lower MPG**. Origin 2 (Europe) lies in the between. (1.13)



1.13. Distribution of MPG by origin

WE LEAD
Code.Hub

Dealing with Missing Values

- 6 missing MPG values: dropped
- 8 missing hp values : fill them using custom "similar car" function
  1. scale numerical features
  2. for each missing hp instance
     - compute absolute differences for each feature
     - row-wise sum all absolute differences to form a car similarity score
     - find the index of the most similar car
  3. assign the target column from the most similar car

	Feature	Row 38	Row 62	Absolute Difference
1	mpg	0.190289	0.446497	0.256208
2	cylinders	-0.856321	-0.856321	0.0
3	displacements	-0.916334	-0.925936	0.009602
4	horsepower		-1.156791	
5	weight	-1.092988	-1.343645	0.2506569999999999
6	acceleration	1.246054	1.246054	0.0
7	Total Absolute Difference			0.5164669999999999

1.14. Example of similar car function

```
Imputed Values for Missing Entries in 'horsepower':
Row 38: Imputed Value = -1.1567907473949541 (Based on Similar Car at Row 62)
Row 133: Imputed Value = -0.42842135527660774 (Based on Similar Car at Row 373)
Row 337: Imputed Value = -1.2088171325462647 (Based on Similar Car at Row 350)
Row 343: Imputed Value = -0.16828942952005543 (Based on Similar Car at Row 186)
Row 361: Imputed Value = -0.68855328103316 (Based on Similar Car at Row 324)
Row 382: Imputed Value = -0.37639497012529727 (Based on Similar Car at Row 322)
```

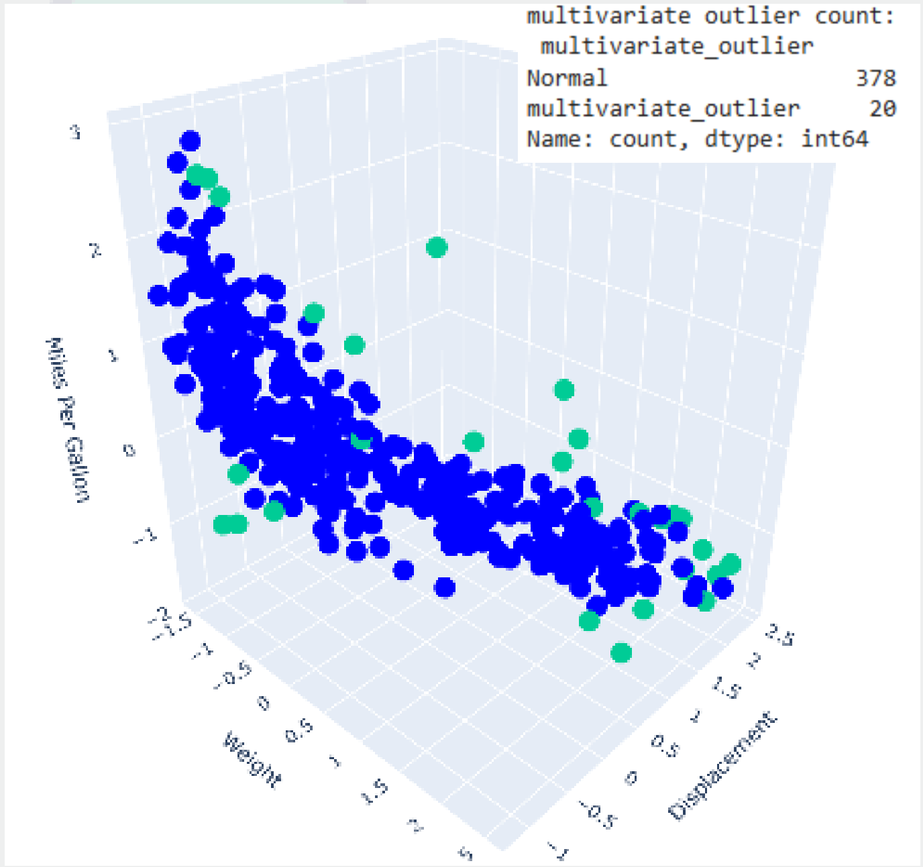
1.15. Imputed values for missing hp based on similar car function

WE LEAD
Code.Hub

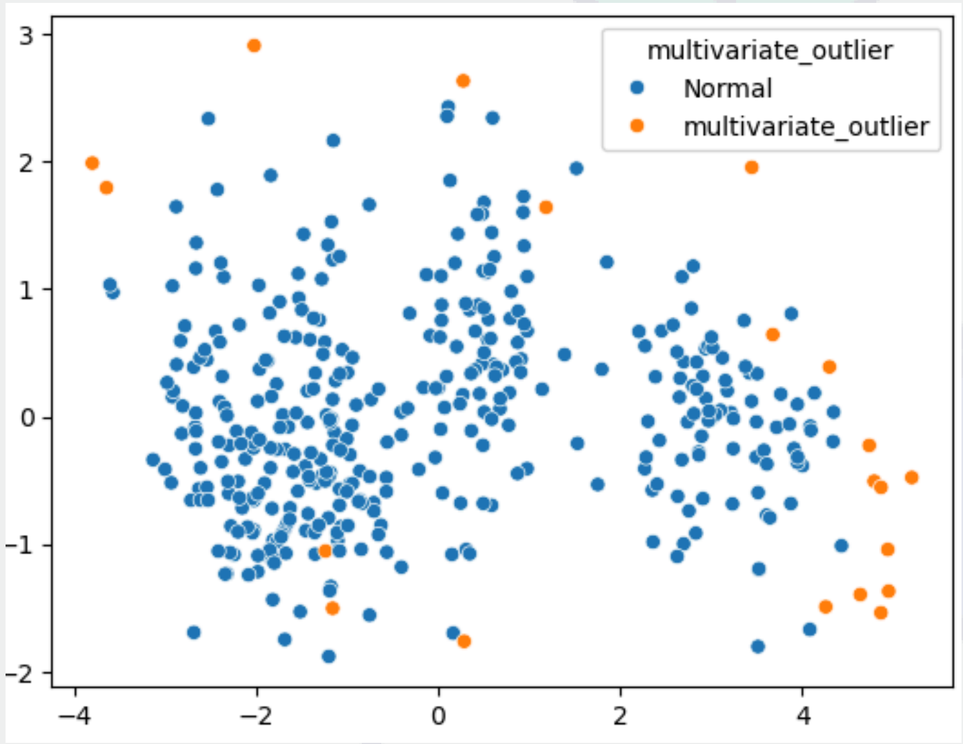
Detecting Multivariate Outliers

Indices of multivariate outliers: [6, 7, 8, 19, 31, 32, 34, 74, 97, 101, 102, 118, 123, 306, 307, 333, 340, 341, 372, 402]

Indices of feature outliers: [6, 7, 8, 16, 19, 31, 66, 101, 102, 123, 306, 329, 333, 402]



1.16. Mutlivariate outliers using Isolation Forest



1.17. Outliers projected in a 2dimensional space using PCA

23 total outliers

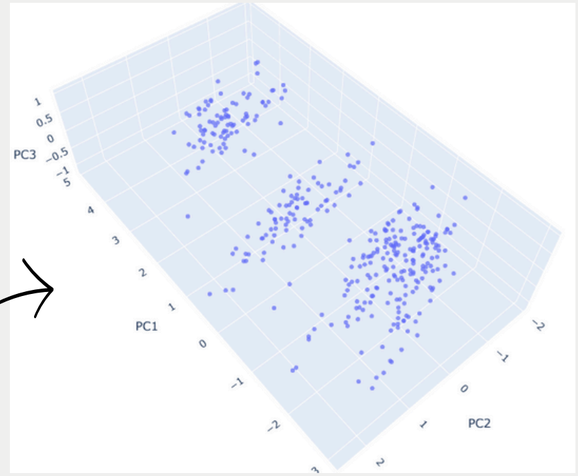
WE LEAD
Code.Hub

MPG Categorization

1. Perform Principal Component Analysis

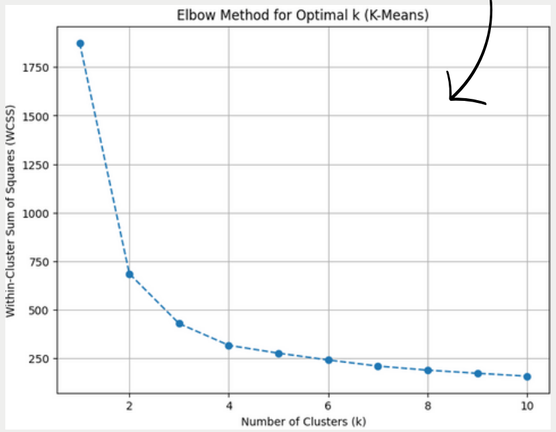
2. Find optimal number of Components using Elbow Method (98.21% explained variance for 3 components)

3. Project the data in the 3D space formed by the PCs



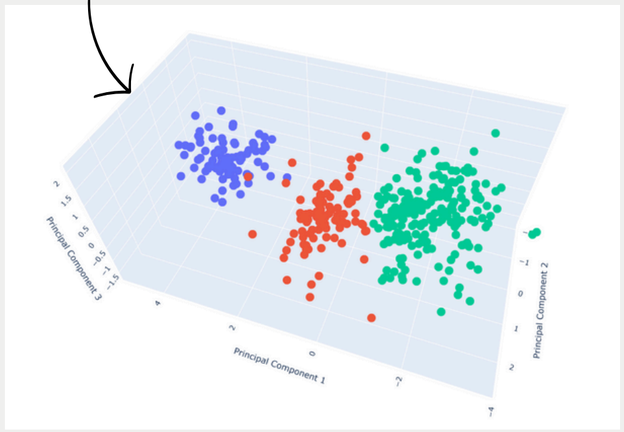
1.18. Projection on <PC1,PC2,PC3>

4. Elbow method for optimal number of clusters (k=3)



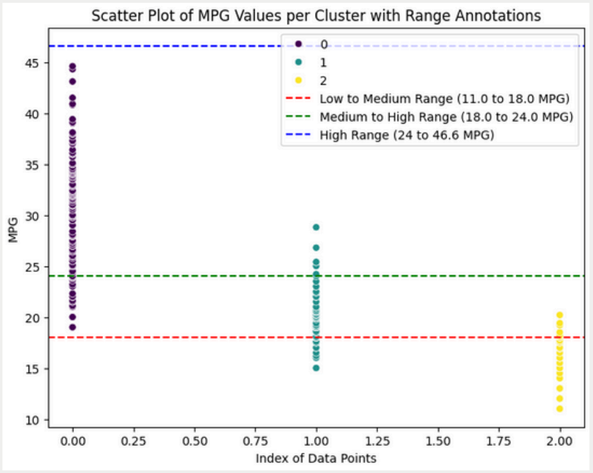
1.19. Elbow for optimal number of clusters

5. K-means clustering



1.20. K-means results

1.21. MPG per cluster with range



6. scatterplot mpg values for each cluster

7. choose mpg labels range and split the dataset accordingly

Low <= 18.0
18< Medium <=24
24 < High

```
mpg_label
High      167
Low       110
Medium     98
dtype: int64
```

WE LEAD
Code.Hub

MPG Categorization

Dataset labels were distributed in order for a relatively **balanced split** of the three categories to be acquired.

The acquisition was done based on **quantiles**:

Low= 0 (threshold 19.0)

Medium= 1 (threshold 27.0)

High= 2 (anything above > 27.0)

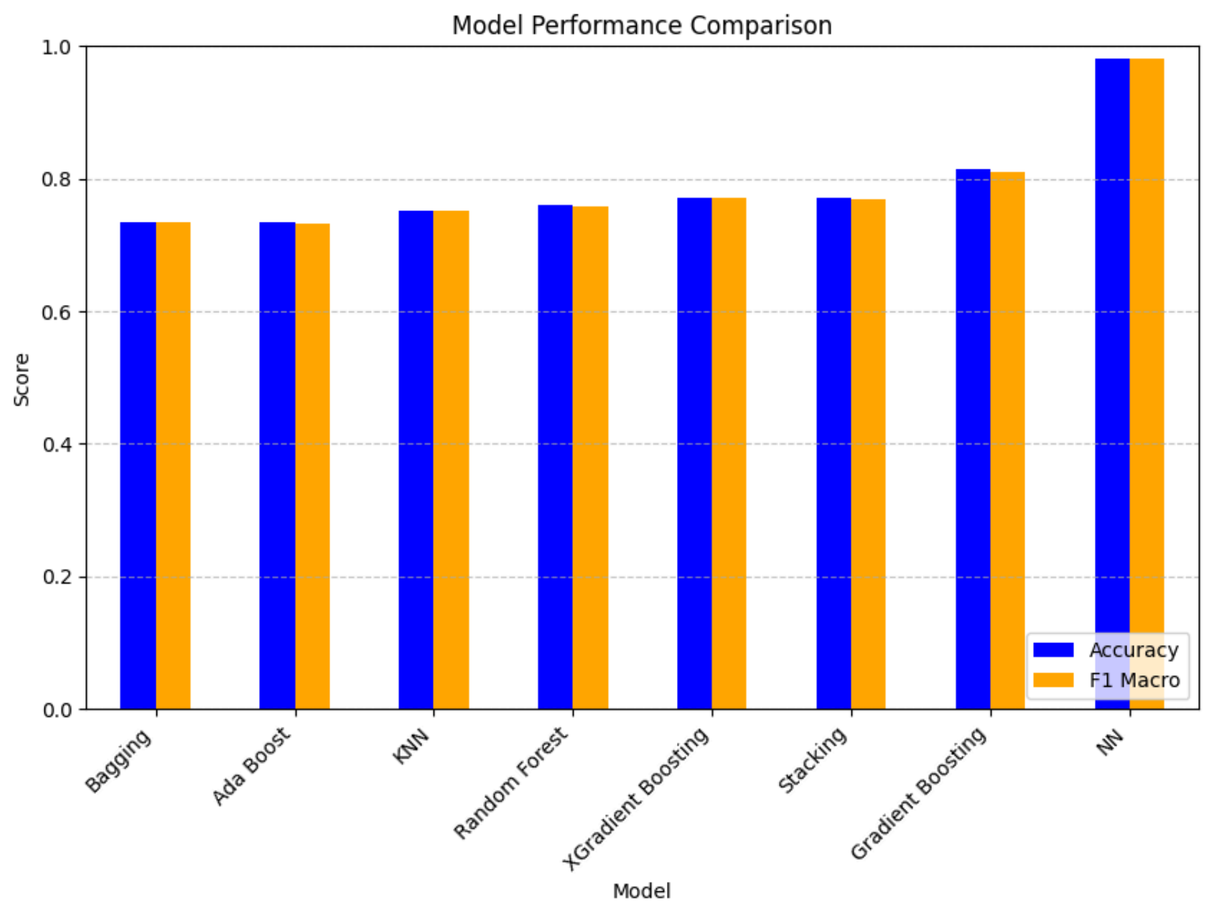
```
Quantile thresholds:
q1: 19.0
q2: 27.0

Label distribution:
mpg_classes
0    129
1    127
2    119
Name: count, dtype: int64
```

1.22. The final labeling categorization based on quantiles

WE LEAD
Code.Hub

Classification Models/ Approaches



1.23. Bar chart comparison of classification models based on Accuracy and F1-score.

Models	*Accuracy*	*F_1 score*
Ada Boost	0.7345	0.7330
Gradient Boosting	0.8142	0.8103
XGradient Boosting	0.7699	0.7708
Bagging Classifier	0.7345	0.7348
Random Forest	0.7611	0.7581
Stacking Classifier	0.7699	0.7684
NN	**0.9813**	**0.9815**

1.24. Accuracy and F1-score results for different classification models

Feature Engineering & Preprocessing:

- **Standardization** with StandardScaler
- Stratified **train-test split** (70%-30%)

Hyperparameter Optimization (GridSearchCV)

- n_estimators
- learning_rate
- max_depth
- max_samples
- colsample_bytree
- gamma
- reg_lambda
- subsample

WE LEAD
Code.Hub

Neural Network - MLP (Multi Layer Perceptron)

- **Data split:** Training and test sets using **5-fold cross-validation**

Basic Architecture

Layer Structure:

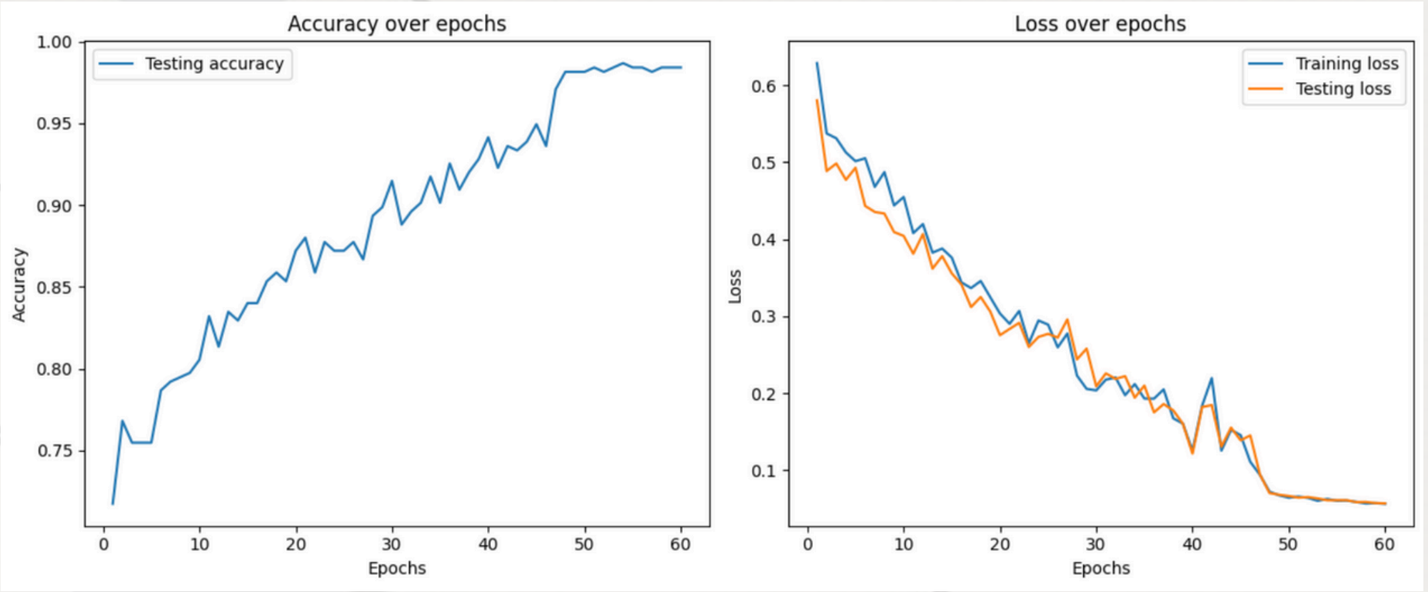
- Input Layer (5 features)
- Hidden Layers: 64 → 32 → 16 neurons
- Output Layer: 3 neurons

Parameters (weights + biases): 3043

Activation Function: Tanh – Hyperbolic Tangent (range of input values between –1 & 1)

Hyperparameters

- **Learning Rate**: 0.01
- **Epochs**: 60
- **Batch Size**: 32
- **Optimizer**: Adam
- **Loss Function**: CrossEntropyLoss (for multi-class classification)
- **Scheduler**: ReduceLROnPlateau (learning rate adjustment)
- **Early Stopping**: Stops training if no improvement after 7 epochs



1.25 Accuracy over epochs and training - testing loss during epochs

Final Model Evaluation:

	precision	recall	f1-score	support
Low	0.98	0.98	0.98	129
Medium	0.97	0.98	0.97	127
High	0.99	0.98	0.99	119
accuracy			0.98	375
macro avg	0.98	0.98	0.98	375
weighted avg	0.98	0.98	0.98	375

WE LEAD
Code.Hub

Thank you for your attention!

Any questions?