אוניברסיטת
אריאל
בשומרון

Analyzing Trends and Characteristics of Vehicle Recalls

Course: Advanced Topics in Machine Learning

Lecturer: Chen Hajaj

Team Members: Lilya Shvedskaya & Lior Vanounou

Github link:

Abstract:

The primary objective of this project is to understand and predict vehicle trends in Israel, providing valuable insights for policymakers, manufacturers, and insurance providers. This analysis can play a key role in enhancing road safety, optimizing fuel policies, and supporting the adoption of eco-friendly vehicles. Given the global shift towards sustainable energy, insights into fuel type trends and vehicle weight distributions are of significant importance.

The data for this project was sourced from the data.gov.il website through an API call. The data processing steps involved removing irrelevant columns and cleaning the data to ensure its quality. To handle anomalies, we employed Isolation Forest and pattern analysis within groups. Afterward, categorical variables were encoded, and once the data preparation process was completed, we proceeded to the analysis phase.

The study involved analyzing results obtained through algorithms such as Random Forest, Gradient Boosting, Neural Networks, SVM, and KNN. This process included model training, model evaluation, and comparison of model performance. In the next phase, unsupervised analysis was conducted using Clustering techniques such as KMeans and DBSCAN, followed by visualization of the resulting clusters.

Introduction:

Despite significant advancements in automotive technologies, including renewable energy, autonomy, and smart transportation, there remains a need to understand market trends in Israel, particularly regarding privately imported vehicles, both new and used. Analyzing these trends can improve road safety, optimize fuel policies, and promote eco-friendly vehicles. In the transition to sustainable energy, it is essential to evaluate new fuel technologies, vehicle weight distribution, and demand for technological features.

This research aims to explore the relationships between the characteristics of privately imported vehicles and factors influencing their usage patterns. We seek insights into anomaly detection for these vehicles, identifying potential risks unique to this group. This analysis will compare privately imported vehicles with standard imports, assessing their impact on usage, demand, and maintenance needs.

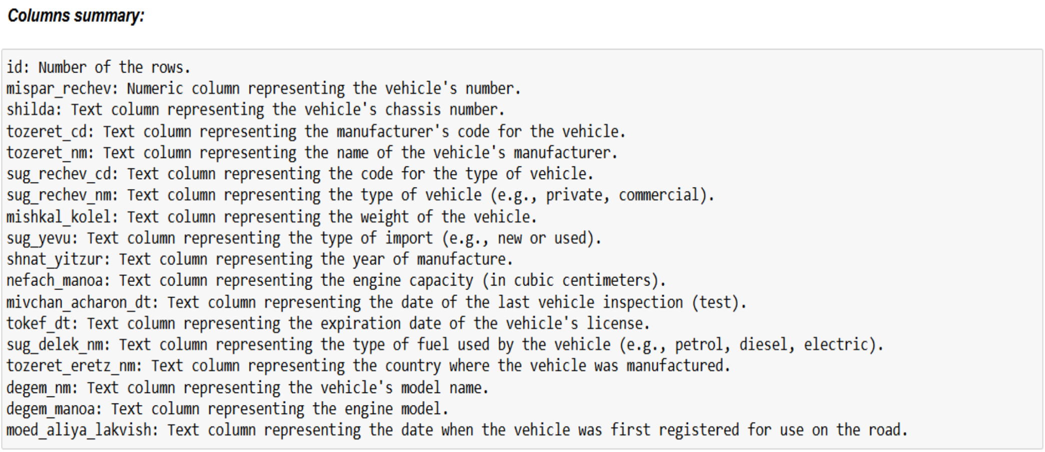
Additionally, we will predict the characteristics of privately imported vehicles to support regulatory and market analysis, understanding the evolving needs of this segment. Clustering techniques will be applied to group vehicles by various characteristics, enabling more targeted policies and strategies for manufacturers, service providers, and regulators. These insights will help stakeholders across the automotive industry make smarter decisions, enhance products and services, and shape regulations aligned with current trends in the privately imported vehicle market.

Dataset and Features:

We used a simple API call to retrieve data from data.gov.il. Our data is updated every 24 hours based on the private vehicle imports in Israel. The initial feature list includes 18 columns and 28,465 rows. The content of the columns in the data is

described in Figure 1.

Fig 1:



Columns summary:

```
id: Number of the rows.
mispar_rechev: Numeric column representing the vehicle's number.
shilda: Text column representing the vehicle's chassis number.
tozeret_cd: Text column representing the manufacturer's code for the vehicle.
tozeret_nm: Text column representing the name of the vehicle's manufacturer.
sug_rechev_cd: Text column representing the code for the type of vehicle.
sug_rechev_nm: Text column representing the type of vehicle (e.g., private, commercial).
mishkal_kolel: Text column representing the weight of the vehicle.
sug_yevu: Text column representing the type of import (e.g., new or used).
shnat_yitzur: Text column representing the year of manufacture.
nefach_manoa: Text column representing the engine capacity (in cubic centimeters).
mivchan_acharon_dt: Text column representing the date of the last vehicle inspection (test).
tokef_dt: Text column representing the expiration date of the vehicle's license.
sug_delek_nm: Text column representing the type of fuel used by the vehicle (e.g., petrol, diesel, electric).
tozeret_eretz_nm: Text column representing the country where the vehicle was manufactured.
degem_nm: Text column representing the vehicle's model name.
degem_manoa: Text column representing the engine model.
moed_aliya_lakvish: Text column representing the date when the vehicle was first registered for use on the road.
```

During data processing, we reviewed each column to identify values that could distort or were irrelevant to the analysis. This step was essential for ensuring accuracy. We removed, corrected, and normalized columns and values that added noise to the data. For instance, in the *tozeret_nm* column, we removed country names, which were irrelevant, and standardized car brand names to English.

We also removed columns like *_id*, *mispar_rechev*, and *shilda*, which had unique, non-contributory values. To reduce redundancy, we developed functions that checked for matching values between columns. For example, we identified a 100% match between *sug_rechev_cd* and *sug_rechev_nm* and removed the former. Similarly, we removed *tozeret_cd* and *engine_type_numeric* based on high matching percentages. For the *degem_manoa* and *sug_delek_nm* columns, we retained *degem_manoa*, which was more informative.

A challenge arose with the *degem_nm* column, which had over 17,500 unique values. To handle this, we created a function to normalize model names, reducing the unique values to 13,790. Although still high, we chose to retain the column due to its importance.

We also removed the *mivchan_acharon_dt* column, as it was derived from *tokef_dt*, to reduce redundancy. In the *mishkal_kolel* column, over 50% of values were 0, which could distort the analysis. We created a function to replace these zeros with the average of matching vehicles based on *degem_nm*, *tozeret_nm*, and *nefach_manoa*. This reduced 0 values to 7.26%, after which we removed rows with 0 values, ensuring cleaner data for analysis.

Fig 2:

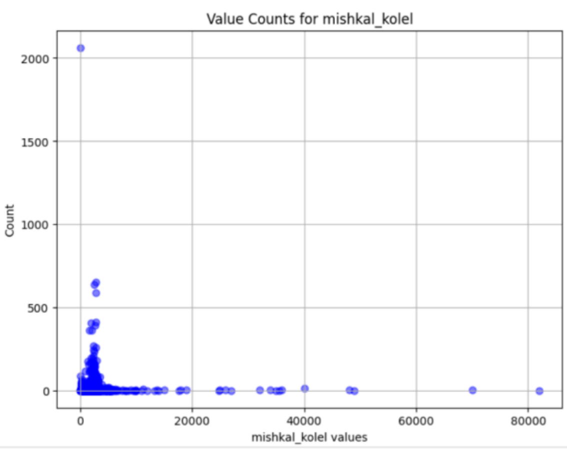
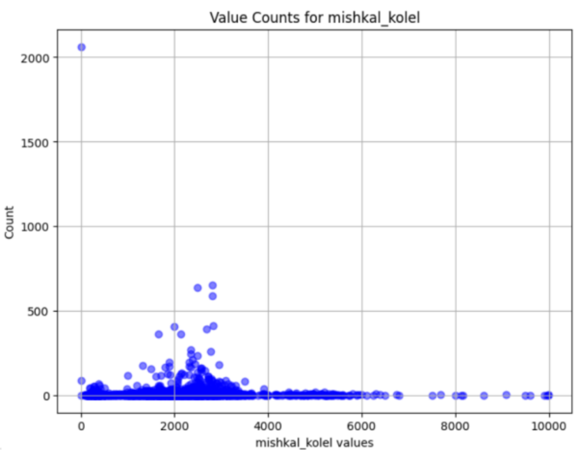
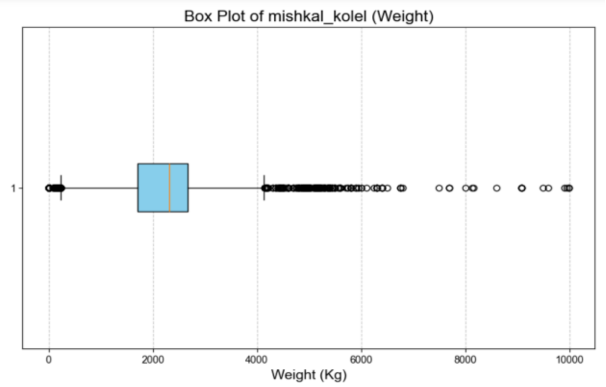


Fig 3:



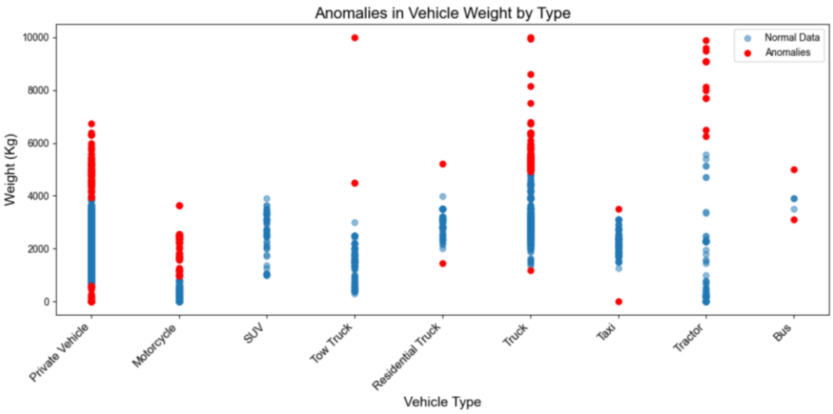
We used the **Interquartile Range (IQR)** method to detect outliers in the dataset, focusing on columns such as **mishkal_kolel** (vehicle weight) and **shnat_yitzur** (year of manufacture). The **IQR method** involves calculating the 25th and 75th percentiles for each column and using them to define upper and lower bounds. Any values outside these bounds were flagged as outliers. We then visualized these outliers using **Box Plots**, which clearly displayed values outside the expected range(fig 4).

Fig 4:



In addition to the IQR method, we employed the **Isolation Forest** algorithm, a machine learning-based outlier detection method that works by isolating data points through randomly constructed decision trees. This method was effective in detecting anomalies in the **mishkal_kolel** column. We visualized these anomalies using **scatter plots**(fig 5), where outliers were marked in red to differentiate them from normal data points.

Fig 5:



We also implemented **group-based anomaly detection** by analyzing the data based on vehicle type. This allowed us to apply the **IQR method** to each vehicle type group individually and identify any outliers within specific vehicle categories. The results were visualized using scatter plots, showing anomalies in vehicle weight for each category(fig 6+7).

Fif 6:

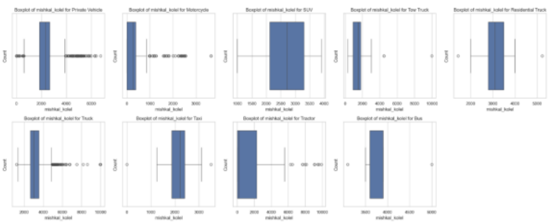
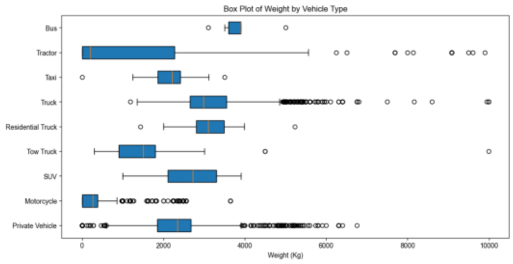


fig 7:



After identifying and visualizing the outliers, we removed extreme ones, such as replacing zero values in the *mishkal_kolel* column with average weights based on vehicle type, brand, and engine volume to maintain dataset consistency. We also split the *tokef_dt* column into separate year and month columns for better temporal analysis.

Throughout data processing, we focused on maintaining data quality, using functions like *detect_outliers_iqr* for outlier detection, and functions like *detect_outliers_year* and *detect_outliers_engine* to handle anomalies in the *shnat_yitzur* and *nefach_manoa* columns. These functions helped refine the dataset for more precise analysis.

We also used visualizations, such as box plots and scatter plots, to highlight anomalies and the distribution of data (fig 8+9). These steps ensured we were working with a high-quality dataset, providing a solid foundation for deeper analysis and informed decision-making in the automotive industry.

Fig 8:

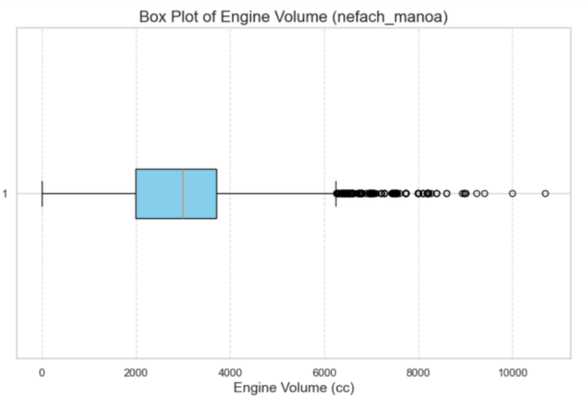
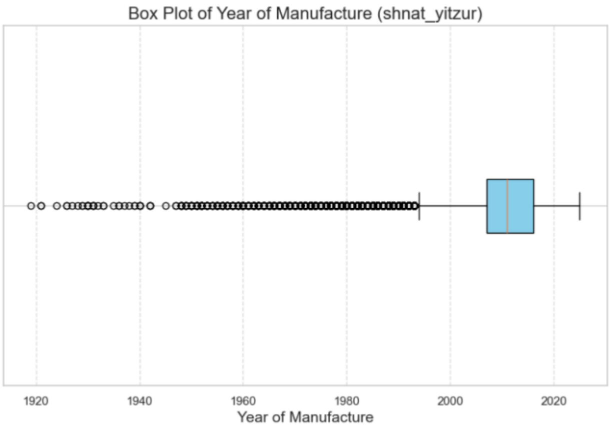


fig 9:



Methodology:

In this section, we will describe the methods used for preprocessing, model training, clustering, and evaluation. The goal of this project is to build predictive models and perform clustering on vehicle data to better understand vehicle characteristics and identify meaningful patterns.

Data Preprocessing and Feature Selection

First, we encoded categorical variables using LabelEncoder. The selected categorical columns include 'tozeret_nm', 'sug_rechev_nm_reduced', 'tozeret_eretz_nm', and 'sug_yevu'. This transformation ensures that categorical variables are represented numerically for the models to process.

We then selected relevant features for training the models: 'tozeret_nm', 'sug_rechev_nm_reduced', 'mishkal_kolel', 'shnat_yitzur', 'nefach_manoa', 'tozeret_eretz_nm', 'tokef_dt_year', and 'tokef_dt_month'. The target variable (y) was defined as 'degem_manoa', representing the label we want to predict.

Before dimensionality reduction, we ensured the numerical columns ('mishkal_kolel', 'shnat_yitzur', 'nefach_manoa') were converted to float type. These columns were then normalized using StandardScaler to ensure all features had the same scale.

Model Training

We performed classification using several models:

- Random Forest: This model was trained on a 10% sample of the training set (X_train_sample and y_train_sample).
- Gradient Boosting: Similar to the Random Forest, Gradient Boosting was trained on the same training subset.

- Neural Network: A Multi-layer Perceptron (MLP) classifier was used with a maximum of 1000 iterations.
- SVM: A Support Vector Machine model was trained on the sample.
- KNN: K-Nearest Neighbors classifier was also trained on the sample.

Each model was then tested on the full test set (X_test) to evaluate its performance.

Model Evaluation

To evaluate the models, we used accuracy as the primary metric. The models were assessed using the evaluate_model function, which calculates the accuracy of predictions (y_pred) against the true values (y_true). The performance of each model was compared to determine which achieved the best accuracy.

Additionally, we compared models based on their performance using a variety of metrics such as precision, recall, and confusion matrix.

Clustering and Dimensionality Reduction

For unsupervised analysis, we performed clustering using KMeans and DBSCAN algorithms.

- KMeans: We set the number of clusters (n_clusters=4) and used the KMeans algorithm to assign labels to each data point.
- DBSCAN: We chose eps=1.5 and min_samples=10 to identify clusters and noise in the data. The Silhouette Score for DBSCAN was computed to assess the quality of the clustering.

Both algorithms were evaluated based on their ability to identify meaningful groups in the data. The Silhouette Score for DBSCAN, for instance, was 0.71, which indicates well-defined clusters.

Clustering Visualization

To visualize the clustering results, we used the plot_clusters function. This function generates scatter plots for both KMeans and DBSCAN clustering results. The clusters were visualized in 2D, using the two most relevant features for each model.

Results & Discussion:

Model Performance:

Model	Accuracy
Random Forest	0.409083
Gradient Boosting	0.008097
Neural Network	0.332864
SVM	0.049463
KNN	0.242211

The Random Forest and Gradient Boosting models performed well, with Random Forest achieving an accuracy of 0.41. However, models like SVM and KNN struggled, likely due to the high dimensionality and imbalanced distribution of the target variable (degem_manoa).

Clustering Analysis

- KMeans clustering resulted in four clusters, with some degree of meaningful grouping in terms of vehicle characteristics (fig 10).
- DBSCAN achieved a high Silhouette Score of 0.71, suggesting that the algorithm was able to effectively separate the data into distinct clusters (fig 11).

Fig 10:

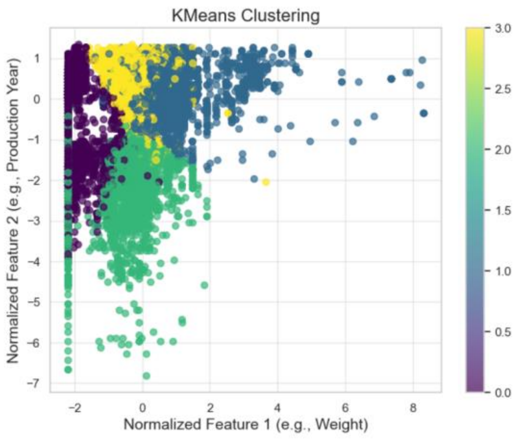
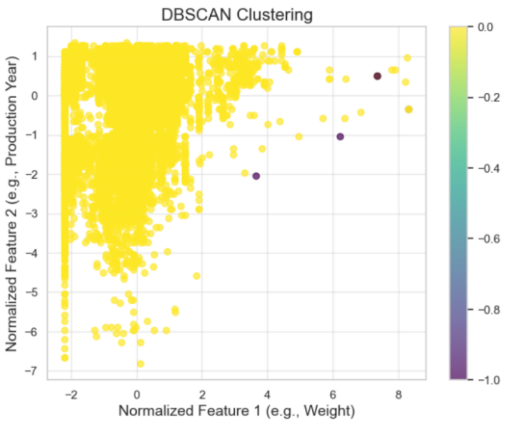


fig 11:



The clusters identified by both algorithms seemed to correspond with known patterns in the data, such as vehicle types and sizes.

Conclusions and Future Work

This project demonstrates the use of classification models and clustering techniques to analyze vehicle data. Random Forest and Gradient Boosting proved to be the best models for classification tasks, while KMeans and DBSCAN provided valuable insights into the data's underlying structure.

Impact:

- The models can predict vehicle attributes like *degem_manoa* and identify anomalies in vehicle data.
- Clustering results can group vehicles based on shared characteristics, aiding business strategies or regulatory analysis.

For future work, further model tuning, additional features, and larger datasets could enhance performance. Exploring advanced clustering techniques or integrating both classification and clustering models could lead to more effective decision-making tools for vehicle manufacturers or consumers.

Contributions

The work was completed collaboratively, with alternating contributions between team members. Each person focused on areas where they felt most confident. The division of labor was as follows:

- Lilia handled data extraction, presentation, and missing value handling.
- Lior performed the scaling process, cleaned columns, normalized data, and handled outliers, particularly in the *mishkal_kolel* column.
- Lilia encoded categorical variables, trained models, compared results, and worked on clustering and visual presentations.
- Both contributed to the conclusion, and Lior worked on the technical report and presentation.