Classification Algorithms, cleaning data, and cost functions

Chidimma Prisca Ogu

M.SC Data Science

Department of Computing & Electronic Engineering, ATU Sligo

Abstract

This project investigates the use of vehicle sensor data to classify traffic conditions in urban environments. The dataset comprises of four vehicle runs recorded from Peugeot 207 and Opel Corsa vehicles, capturing 14 sensor-derived features, including vehicle speed, engine load, RPM, acceleration, and fuel consumption.The dataset supports three classification tasks which include: Driving Style, Road Surface Condition, and Traffic Congestion—but this study focuses solely on Traffic Congestion, categorizing it into Low, Normal, and High levels.

Keywords: Classification, Machine Learning, Data, Exploratory Data Analysis (EDA), Cost function

1 Introduction

Traffic congestion poses significant challenges for modern transportation, affecting travel time, fuel use, and safety. This project uses vehicle sensor data to classify traffic congestion levels with machine learning, comparing Logistic Regression and Support Vector Machines (SVM). Data from four runs of Peugeot 207 and Opel Corsa vehicles, capturing 14 features such as speed, engine load, and acceleration, is used. Traffic congestion is chosen as the target for its real-world relevance.The methodology emphasizes: Exploratory Data Analysis (EDA) to identify patterns, missing values, and imbalances. Feature engineering and selection to improve model performance. Preprocessing, including scaling and SMOTE oversampling to balance classes. Classification using Logistic Regression and SVM. Evaluation via F1-score, confusion matrices, ROC-AUC, and cost functions. The findings highlight how sensor-based models can inform intelligent and connected vehicle systems for real-time traffic management.

2 State of the Art

Modern vehicles increasingly rely on high-frequency in-vehicle sensor data, with many cars containing more than seventy ECUs communicating via CAN, FlexRay, or automotive Ethernet (Abdeslam et al., 2022). These systems generate rich data streams that support vehicle behaviour analysis, road-condition assessment, and intelligent driver assistance.

Beyond internal networking, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication enable shared perception across connected transport networks. Although this project does not implement V2V/V2I, its methodology contributes to these goals by allowing vehicles to infer and potentially share local traffic congestion states.

Recent work integrates OBD-II signals with smartphone-based sensors to model the driver–vehicle–environment system, supporting tasks such as driving-style classification and road-condition detection (Bergenhem et al., 2024). This project adopts a similar workflow, using cleaned and SMOTE-balanced datasets to train ML models—such as Logistic Regression and SVM—for congestion classification. Train/test cost functions (log-loss and hinge loss) provide additional insight into model reliability.

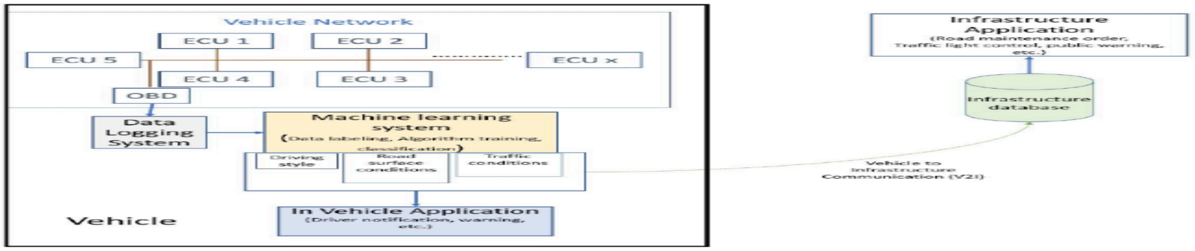
Vehicle Network
ECU 1
ECU 2
ECU 5
ECU x
ECU 4
ECU 3
OBD
Data Logging System
Machine learning system
(Data labeling, Algorithm training, classification)
Driving style
Road surface conditions
Traffic conditions
In Vehicle Application
(Driver notification, warning, etc.)
Vehicle
Infrastructure Application
(Road maintenance order, Traffic light control, public warning, etc.)
Infrastructure database
Vehicle to infrastructure Communication (V2I)

Figure 1. The figure shows the system architecture for road conditions and driving style prediction.

Figure 1

(Abdeslam et al., 2022) illustrates a typical in-vehicle prediction architecture.

3 My Work

3.1 Exploratory Data Analysis (EDA)

3.1.1 Missing Values

Minimal missing data (<0.2%) across 8 sensor features (Engine Load, Air Temperature, Throttle Position).

Rows with missing values were removed to avoid artificial patterns (Udacity, Predictive Analytics for Business). Irrelevant columns like Unnamed 0, and other classifications: Driving Style and Road Surface Condition were dropped.

3.1.2 Class Imbalance

The Target variable (traffic_congestion) heavily imbalanced (e.g., Opel Corsa: >85% Low Congestion). With the risk of the classifiers biased toward the majority class, poor detection of Normal/High congestion. To resolve this, SMOTE (Synthetic Minority Oversampling Technique) was used to generate minority-class samples (Chawla et al., 2002).

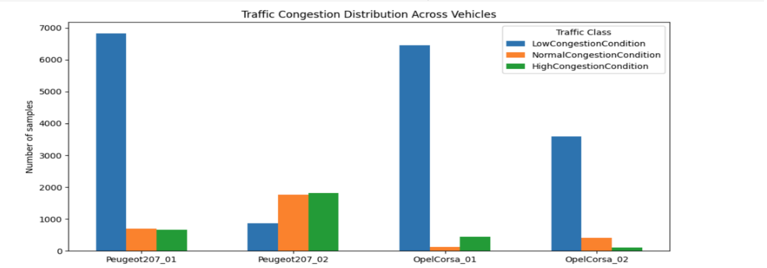

Figure 1. Traffic Congestion Distribution Before Balancing

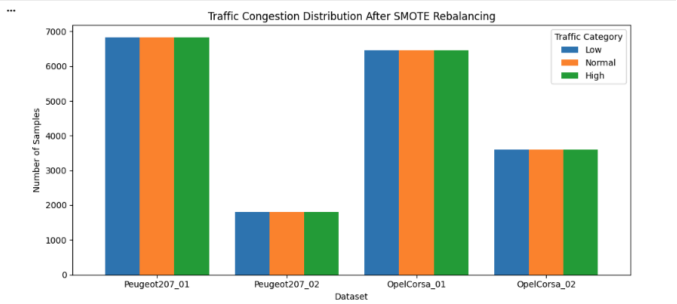

Figure 1.1. Traffic Congestion Distribution After SMOTE Rebalancing

3.1.3 Correlation & Multivariate Analysis

Correlation heatmap highlights multicollinearity: It shows high correlation between Vehicle Speed Instanteneous and Vehicle Speed Average, MassAirFlow and Manifold Absolute Pressure, Longitudinal Acceleration and Vertical Acceleration as well as Vehicle Speed Instanteneous and EngineRPM.

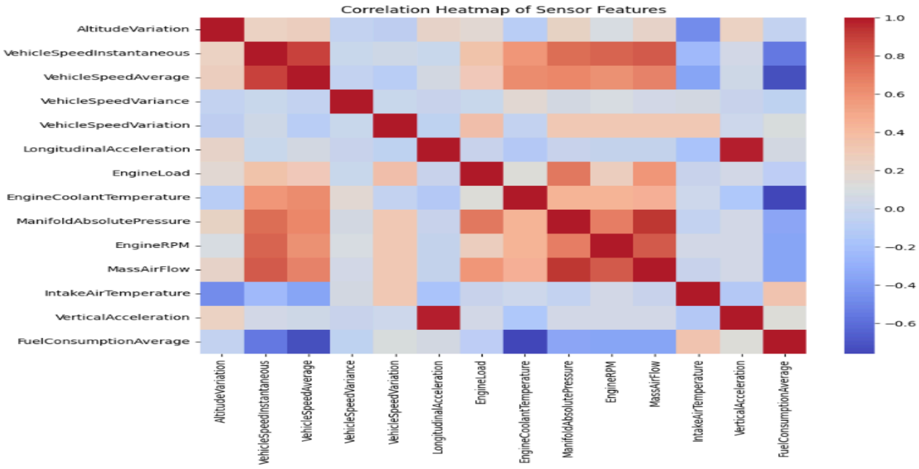

Figure 2. Feature Correlation Heatmap

3.2 Feature Engineering & Selection

This will take the following steps

- Encoding: One-hot/cyclical encoding for temporal & categorical features.
- Scaling: Standardized continuous variables (zero mean, unit variance).

- Feature Selection (ANOVA F-test): using the criteria F-statistic (degree of separation), p-value < 0.05 (significance).
- Evaluates whether feature means differ significantly between Low, Normal, High congestion classes.
- Cleaning: Removed highly correlated or low-variance features.
- Features retained: ['AltitudeVariation', 'VehicleSpeedVariance', 'LongitudinalAcceleration', 'EngineLoad', 'EngineCoolantTemperature', 'ManifoldAbsolutePressure', 'EngineRPM', 'IntakeAirTemperature', 'FuelConsumptionAverage', 'traffic']

4 Model

Model building and evaluation was performed using stratified train-test splitting and cross-validation techniques to ensure robust generalization (Scikit-learn, 2025).

4.1 Model Building

4.1.1 Classifiers Trained

- Logistic Regression (LR) – predicts probabilities of each traffic congestion class.
- Support Vector Machine (SVM) – classifies congestion based on separating hyperplanes.

4.1.2 Cost Function Calculation

Log Loss evaluates probability predictions, while Hinge Loss measures margin violations in class separation. Both quantify model performance during training and testing (Nama, 2023)

- **Logistic Regression (Log Loss):**

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^{N} -(y_i * \log(p_i) + (1-y_i) * \log(1-p_i))$$

- **Support Vector Machines(Hinge Loss):**

$$HL = \max(0, 1 - y\_actual * y\_pred)$$

Note: Cost functions were computed for both training and test sets to assess model generalization.

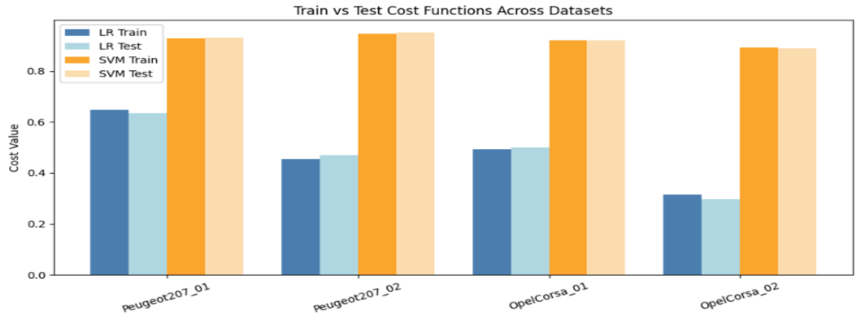

Fig 3. Showing comparison between cost function for the different models across datasets

4.2 Model Evaluation

4.2.1 Evaluation Metrics Used:

- F1-score (macro) – balances precision and recall across all classes.
- Confusion Matrix – shows correct vs misclassified samples per class.
- ROC Curve and Macro-average AUC – illustrates class separation for each classifier.

4.2.2 Observations:

- SVM consistently achieves higher F1-scores and Macro-AUC values, indicating stronger class discrimination.
- Logistic Regression provides interpretable probability outputs and performs well on majority classes but is slightly less robust for minority congestion categories.

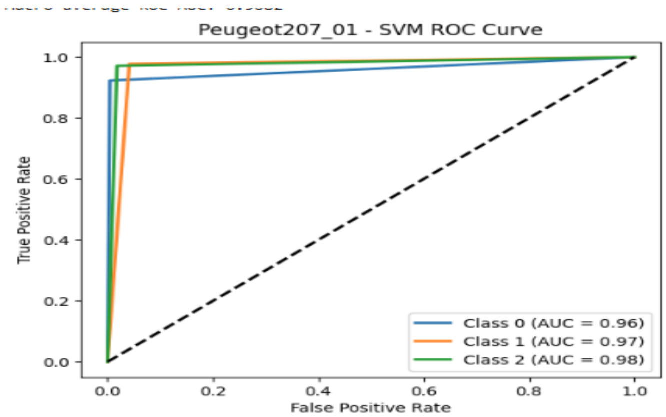

Figure 4. ROC curve figure comparing SVM and Logistic Regression for Peugeot_207_01 dataset

5 Conclusions

SVM consistently outperforms Logistic Regression, particularly for minority congestion classes. Logistic Regression shows low Log-Loss values, indicating strong probability calibration, with the best performance observed in the OpelCorsa_02 dataset. Although SVM exhibits higher Hinge Loss values overall, the small differences between its training and test losses demonstrate strong generalisation capability. Across all datasets, the similarity between train and test cost function values suggests minimal overfitting and robust model behaviour.
Dataset Highlights: Peugeot207_01 shows the highest Logistic Regression Log-Loss (0.647), whereas OpelCorsa_02 exhibits the lowest (0.298), indicating the most accurate probability estimations among all datasets.

Full code and workflow are available on GitHub:
Placeholder link → https://github.com/your-repo-here

6 References

1. Abdeslam, D., El-Masri, W. & Abousleiman, R. (2022) In-Vehicle Data for Predicting Road Conditions and Driving Style Using Machine Learning. ResearchGate. Available https://www.researchgate.net/publication/363342764_In-Vehicle_Data_for_Predicting_Road_Conditions_and_Driving_Style_Using_Machine_Learning (Accessed: 6th Nov, 2025).
2. Bergenhem, C., Knaeps, S., Leven, M., Erraeissi, A., Suykens, J. & De Schutter, B. (2024) Driver–Vehicle–Environment Modelling Using OBD-II and Smartphone Sensor Fusion. Applied Sciences, 14(9) p. 3905. Available at: https://www.mdpi.com/2076-3417/14/9/3905 (Accessed: 6th Nov, 2025)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, pp. 321–357.
4. Udacity. (n.d.). Predictive Analytics for Business. Available at: https://www.udacity.com/course/predictive-analytics-for-business--nd008 (Accessed: 6th Nov, 2025).
5. Scikit-learn. (2024). User Guide: Data Preprocessing. Available at: https://scikit-learn.org/stable/modules/preprocessing.html (Accessed: 6th Nov, 2025).
6. Nama, A. (2023) Understanding cost functions in machine learning: types and applications. Medium. Available at: https://medium.com/@anishnama20/understanding-cost-functions-in-machine-learning-types-and-applications-cd7dCc4b47d (Accessed: 15 November 2025).
7. Scikit-learn (2025) Model selection and Evaluation: Available at: https://scikit-learn.org/stable/model_selection.html (Accessed: 15 November 2025).