Comparative analysis of machine learning algorithms for predicting vehicle driving conditions

Introduction

This paper presents a comprehensive comparison of three distinct machine learning algorithms – Support Vector Machines (SVM), Logistic Regression and K-Nearest Neigbors (kNN) – to classify vehicle driving conditions. Utilizing a dataset derived from vehicle sensors, this study aims to identify the most effective algorithm for predicting various driving conditions categorized into Driving Style, Road Surface Condition and Traffic Congestion.

The dataset for this analysis is sourced from Kaggle [1], featuring sensor data from Opel Corsa and Peugeot 207 vehicles over multiple journeys. This dataset poses typical real-world data challenges, including missing values, variable scale, and class imbalances, necessitating thorough preprocessing and analysis techniques.

Methodology

Data Preprocessing

The raw data, collected via Kaggle, included sensor readings from two vehicle models, the Opel Corsa, and the Peugeot 207, over two journeys each. The merged dataset's initial review highlighted common difficulties associated with real-world data, such as missing values and numerical representation errors. Numeric columns with decimal commas were transformed to the required float format.
To address the issue of missing data, a median imputation technique was used, which is recommended since it is more robust than mean imputation, particularly in the presence of outliers [2]. Post imputation, the data was re-evaluated to ensure that it was comprehensive and ready for machine learning. Figure 1 illustrates the distribution of each sensor reading post-preprocessing, highlighting the variations in data spread and potential outliers.

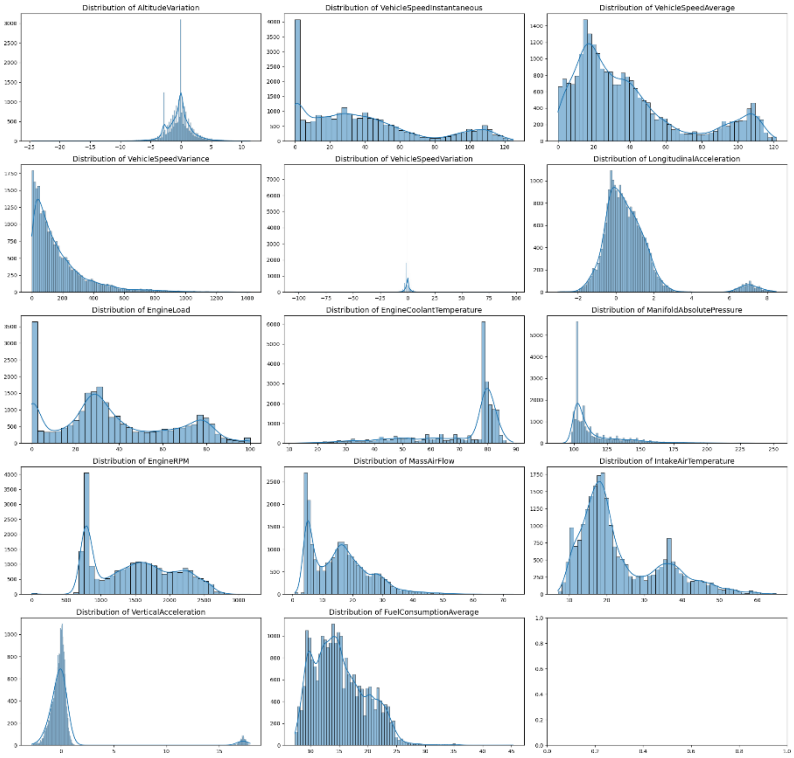

FIGURE 1

Feature Evaluation

Prior to feature selection, a visualisation of the distribution of each sensor reading was performed to examine the data's spread and skewness. The seaborn and matplotlib libraries were used to generate histograms, which revealed valuable information about the distribution of each feature [3].

The Random Forest method was used to assess feature relevance, ranking features according to their influence on classification outcomes. The literature validates the effectiveness of this strategy for feature selection. [4]. Figure 2 shows the feature importances as determined by the Random Forest classifier.

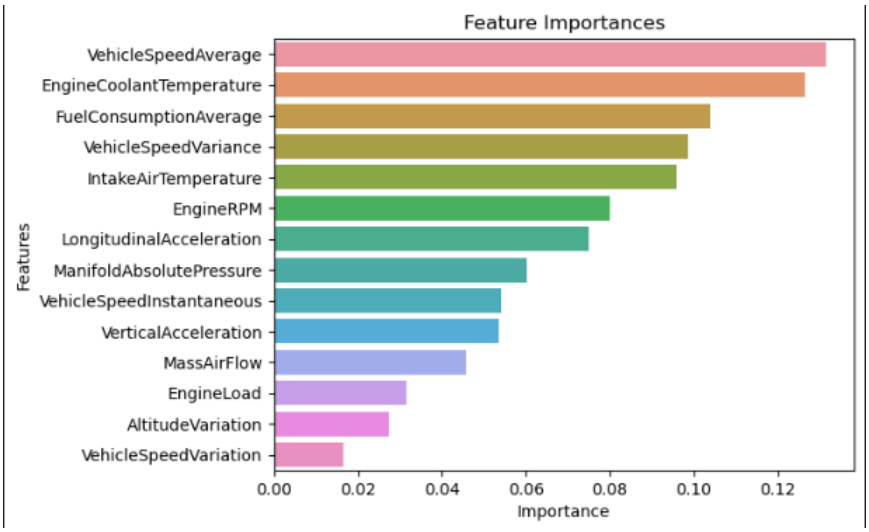

FIGURE 2

Dataset Balancing

The target variable's class distribution was significantly imbalanced, creating a potential bias in predictive models. To remedy this, the Synthetic Minority Over-sampling Technique (SMOTE) was used to generate synthetic samples from minority classes, resulting in a balanced dataset [5]. This approach is well-known for improving classifier performance in the presence of skewed datasets.

Hyperparameter Tuning

Hyperparameter tweaking was required for the k-Nearest Neighbours (kNN) classifier to determine the ideal number of neighbours for prediction. Although k=1 resulted in the maximum accuracy, it was prone to overfitting, as indicated by the disproportionate accuracy trade-off between classes. As a result, k=5 was chosen for its balance of accuracy and generalisation, as supported by the model complexity considerations stated by James et al [6].

Each preprocessing step was methodically documented with code comments to clarify the reasons behind decisions, assuring the study's transparency and reproducibility.

Experiments and Results

Classifier Selection and Configuration

Three classifiers were selected for the comparison analysis based on their popularity in the machine learning domain and their various operating principles, which provide a comprehensive perspective on the dataset's predictability:

1. **SVMs with Linear Kernel:** The SVM was chosen for its success in high-dimensional spaces and ability to separate linearly, and it was setup with a linear kernel to cater to the dataset's linearly separable features.
2. **Logistic regression:** A probabilistic classifier well-suited for binary classification tasks, was used to set a baseline for performance, leveraging its simple implementation and interpretability.

3. **K-Nearest Neighbors (kNN):** a simple and effective non-parametric classifier used for classification problems. The decision to choose k=5 rather than k=1, despite the latter offering higher accuracy, was motivated by a desire to avoid overfitting and ensure the model's generalisation to new, previously unseen data.

Experimental Setup

The dataset was divided into training and testing sets using a 70:30 ratio. Given the observed imbalance in the target variable's class distribution, SMOTE was used to the training data to generate new examples, thereby correcting the imbalance, and creating a more equitable learning environment for the classifiers.

GridSearchCV was used to tune hyperparameters, particularly for the kNN classifier, by iterating through a predefined range of neighbours to find the ideal value for k. The final set of k=5 neighbours established a compromise between accuracy and model complexity.

Results Analysis

The performance of each classifier was evaluated using cross-validation to ensure robustness. Metrics such as precision, recall, and F1-score were computed and reported for each class, providing a multidimensional view of each classifier's performance. Figure 3 presents a summary of the classification report for each classifier.

```
      Linear SVM Accuracy: 0.63
Classification Report for Linear SVM:
              precision    recall  f1-score   support

           0       0.37      0.80      0.51       906
           1       0.91      0.63      0.75      5330
           2       0.27      0.45      0.34       897

     Logistic Regression Accuracy: 0.64
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       0.40      0.77      0.53       906
           1       0.92      0.63      0.75      5330
           2       0.26      0.51      0.34       897

          kNN Accuracy: 0.97
 Classification Report for kNN:
              precision    recall  f1-score   support

           0       0.82      0.95      0.88       906
           1       0.99      0.93      0.96      5330
           2       0.77      0.94      0.85       897
```

FIGURE 3

• **Support Vector Machine (SVM)** demonstrated moderate precision and recall, indicating reasonable performance yet room for improvement, especially in distinguishing between some of the classes.

• **Logistic Regression** showed comparable precision and recall scores to SVM.

• **K-Nearest Neighbors (kNN)** exhibited superior performance across all metrics, particularly for class 1, where it achieved a near-perfect precision and recall. The trade-off in selecting k=5 was justified by the enhanced generalisation as evidenced by the consistency across classes.

Figure 4 depicts the Receiver Operating Characteristic (ROC) curves for each classifier, with the Area Under Curve (AUC) values indicating the classifiers' abilities to distinguish between classes. kNN's AUC values were exceptionally high, suggesting a strong classification ability.

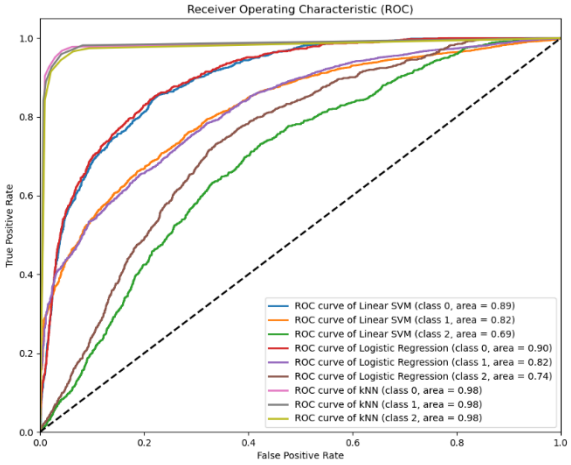

FIGURE 4

Conclusion

This work conducted a comparative comparison of three machine learning classifiers - SVM with a linear kernel, Logistic Regression, and kNN - to predict vehicle driving situations using sensor data. The kNN classifier, with k set to 5, proved to be the most effective model, having the highest precision, recall, and F1-scores across all classes. Its performance, particularly in terms of class balance and generalisation, emphasises the necessity of selecting model parameters that avoid overfitting and encourage robustness to fresh data.

The use of SMOTE to balance the dataset was critical in equalising the impact of each class on the classifier's learning process. GridSearchCV's hyperparameter tuning proved useful by systematically selecting a model configuration that improved classification performance.

This study's limitations include the number of classifiers and parameter ranges investigated. Future research could build on this foundation by experimenting with more advanced models, such as deep learning algorithms, which may better capture complicated patterns in high-dimensional data.

References

[1] M. Ruta, “Traffic, Driving Style and Road Surface Condition,” 2018. [Online]. Available: https://www.kaggle.com/datasets/gloseto/traffic-driving-style-road-surface-condition.

[2] P. D. Allison, Missing Data, SAGE Publications, Inc., 2011.

[3] M. L. Waskom, seaborn: statistical data visualization, Journal of Open Source Software, 2021.

[4] L. Breiman, “Random Forest,” *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[5] “Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall,” *Journal of Aritficial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[6] D. W. T. H. R. T. Gareth James, An Introduction to Statistical Learning, Springer New York, 2013.