Comparative Analysis of Classification Algorithms on Driving Styles

Eoin Lawless

March 2024

Abstract

This study conducts a comparative analysis of Support Vector Machine (SVM), Logistic Regression, and k-Nearest Neighbors (k-NN) classification algorithms that are applied to a driving style dataset. It is aimed to research and evaluate the effectiveness of these algorithms in accurately predicting driving behaviors based on a vehicle sensors' dataset.

1 Introduction

This paper explores the application of machine learning algorithms to classify driving styles, aiming to See the results and comparison of the performance of SVM, Logistic Regression, and k-NN in predicting aggressive versus even-paced driving styles. This Dataset was acquired from the online website Kaggle. [1] In the dataset there were three categorical attributes, road surface, traffic and driving style.

2 Methodology

The research followed a structured approach, starting from Data visualization, Discrepancies/Cleaning data, to model training, evaluation, and comparison. The dataset, derived from vehicle sensor readings, includes attributes of driving dynamics e.g... speed, acceleration, and engine metrics. The visualization of the started of the first hint of problems with the code.

2.1 Discrepancies

Initial data obtained from the website were obscured and filled with errors. The data gathered from the Moodle page were fixed versions of the initial data from the website. This data still had an unlikable quality for a dataset used for training a model. The data based on the three category attributes were overly biased (i.e., some categories are overrepresented compared to others)

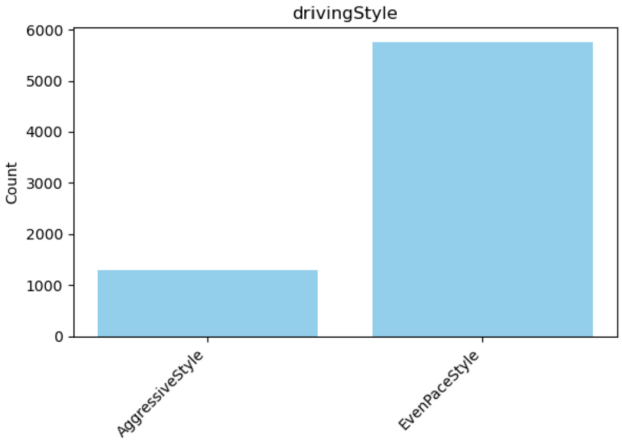

Figure 1: Driving style Imbalance example

2.2 Data Scaling and Balancing

To mitigate bias from variable scales, data features were standardized. The dataset was also balanced to ensure equal representation of aggressive and even-paced driving styles, this was done by employing

techniques such as Down-sampling to avoid model bias towards the more prevalent class.[2] Down-sampling involves randomly removing instances of the majority class to reduce its size to that of the minority classes. In this case even pace will be reduced to the Aggressive pace. This can improve the performance of your model by ensuring that it doesn't become biased towards the majority class. However, be aware that Down-sampling reduces the total amount of data available for training, which might not be ideal if your dataset is small. This is why the decision the driving style attribute was chosen, it delivered the largest sample of data even after Down-sampling at a rate of:

Table 1: Distribution of Driving Styles in the Dataset

Driving Style	Number of Instances
EvenPaceStyle	1287
AggressiveStyle	1287

the original data size of the two was:

Table 2: Original sample sizes:

Driving Style	Number of Instances
EvenPaceStyle	5751
AggressiveStyle	1287

3 Experiments and Results

3.1 Model Training

Each classifier was trained on the preprocessed dataset. SVM was explored with different kernels, Logistic Regression was applied with regularization to prevent overfitting, and k-NN was tested across a range of neighbor values to find the optimal setting.

3.2 Evaluation Metrics

Models were evaluated using accuracy, precision, recall, and F1-score metrics, with a focus on cross-validation scores to assess generalizability.

- **Accuracy**: The proportion of total correct predictions (both true positives and true negatives) out of all predictions. While it provides a quick overview of model performance, accuracy can be misleading in the presence of imbalanced classes, which was addressed through data balancing techniques in our dataset.
- **Precision**: The ratio of true positives to the sum of true positives and false positives. This metric is crucial in contexts where the cost of a false positive is high, indicating the reliability of positive classifications.
- **F1-Score**: The harmonic mean of precision and recall, providing a single metric to assess a balance between precision and recall. An ideal model aims for a high F1-score, indicating both high precision and high recall.

3.3 Results

- **SVM** [5] achieved the highest cross-validation accuracy with an average CV (Cross-validation) score of 0.7950, indicating its effectiveness in handling the non-linear characteristics of the dataset. This high accuracy suggests that the dataset's features have a significant margin of separation.

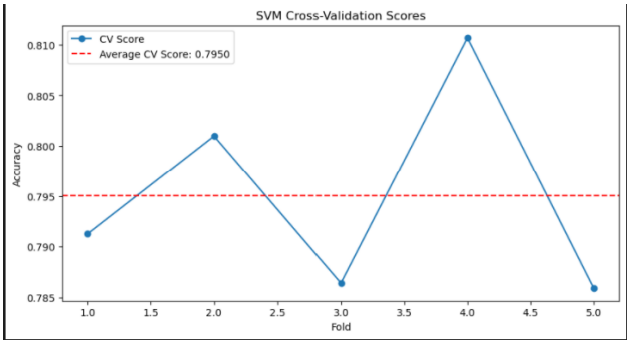

Figure 2: Graph of SVM

- **Logistic Regression** [3] is a straightforward and efficient algorithm for binary and multi-class

classification problems. It provided an average CV score of 0.6741, which is okay but not as high as the SVM. This outcome might be due to the linear nature of Logistic Regression, suggesting that the relationship between the features and the target variable might be more complex.

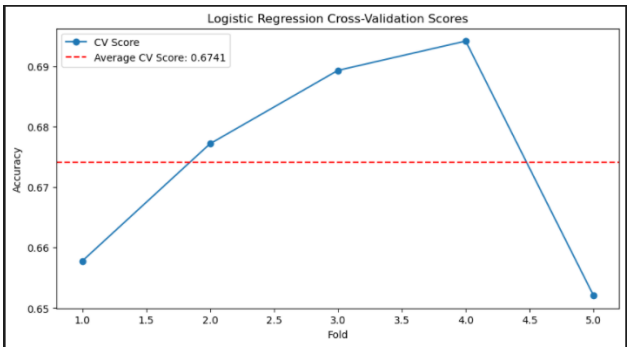

Figure 3: Graph of Logistic regression

- **k-NN** [4] is an instance-based learning algorithm, where the classification of a data is determined by the majority class among its k nearest neighbors. It showed its best performance at k=1 with an average score of 0.6912. The performance at such a low k suggests that the dataset contains instances that are closely grouped by driving style.

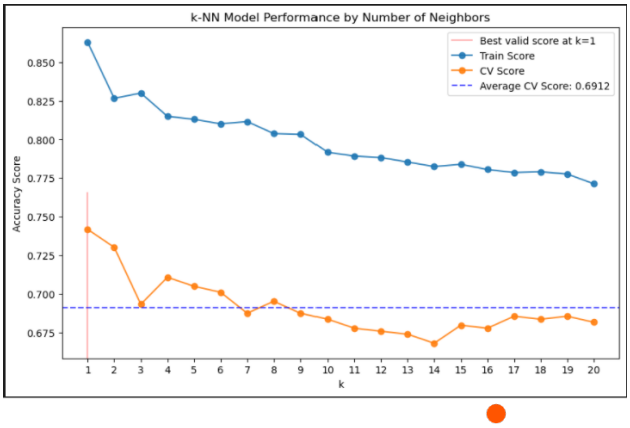

Figure 4: Graph of KNN

4 Discussion

The SVM model outperformed both Logistic Regression and k-NN in the experiments. The average cross-validation (CV) score of 0.7950 is quite good, considering the context of machine learning models, where often or not the scores are rarely perfect. Here's some good things about the score:

1. **Bench marking**: Without knowing the specific context or complexity of the data, a CV score close to 0.8 indicates that the model is performing significantly better than random guessing, which would yield a score around 0.5 for a binary classification task. It suggests that the model has learned patterns from the data.

2. **Generalization Ability**: The use of cross-validation helps ensure that the model's performance is not just a result of overfitting to a particular subset of the data but rather an indication of its ability to generalize well to unseen data

Another approach that could've been used to address the imbalance in the dataset, aside from down-sampling, involves techniques such as up-sampling the minority class, generating synthetic samples using methods like the Synthetic Minority Over-sampling Technique (SMOTE), or adjusting the class weights in the model training process. Up-sampling simply duplicates random samples from the minority class to match the number of instances in the majority class, thereby balancing the dataset. On the other hand, SMOTE generates synthetic examples rather than duplicating existing ones by interpolating between several minority class instances that lie close together.

The approach taken did overall affect the training scores significantly. When first experimenting with the code, the imbalance wasnt taken into consideration, this led to a larger dataset and far higher accuracy scores, but the graph where largely drastic in its outliers, which led to the discovery of the bias.

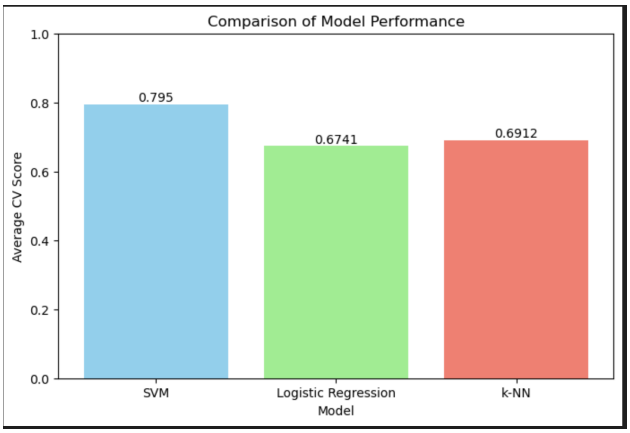

Figure 5: Comparison of the models

5 Conclusion

This study was all about comparing different machine learning methods to see how they perform. It was discovered that the Support Vector Machine (SVM) method was the standout, mainly because it's versatility and how it can handle a wide range of data types. While SVM took the top spot, the other methods that were tested, like Logistic Regression and k-nearest Neighbors (k-NN), were also useful. They gave good insights into how different approaches work with our dataset. One insight is that after downsampling, there wasnt a ton of data to work with. This limitation might have influenced how well each method performed. With more data, the results may have been much different or even clearer differences in how these methods handle the analysis.

References

[1] GIUSEPPE LOSETO. "Traffic, Driving Style and Road Surface Condition". In: (2019). URL: https://www.kaggle.com/datasets/gloseto/traffic-driving-style-road-surface-condition.

[2] Usman Malik. "Upsampling and Downsampling Imbalanced Data in Python". In: (). URL: https://wellsr.com/python/upsampling-and-downsampling-imbalanced-data-in-python/.

[3] "sklearn.linear$_m$odel.*LogisticRegression*". In: (). URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[4] "sklearn.neighbors.KNeighborsClassifier". In: (). URL: https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py.

[5] "sklearn.svm.SVC". In: (). URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.