Machine Learning COMP09012

Group Assignment 2021

Group A: Amy Reidy, Andre McQuaid, David Kelly and Ruairí McConville

Keywords: Classification, Algorithm, Data, Support Vector Machine, Random Forest.

1 Introduction

Machine learning (ML) algorithms use a variety of mathematical techniques to analyse input features and predict outputs. One emerging application of ML is the automatic detection of road surface defects using data gathered from devices such as GPS sensors, in-vehicle cameras, and smartphone accelerometers (Nomura & Shiraishi, 2015). The development of these kinds of monitoring systems has many benefits, including improving road safety and increasing the viability of autonomous vehicles. The goal of this project was to experiment with using multi-classification algorithms to predict three different types of road surface conditions. The two types of classifiers that were chosen for this task are Support Vector Machine (SVM) and Random Forest (RF). After the models were trained with a range of features such as vehicle speed, vertical acceleration and engine coolant temperature, their performances were compared to determine which was the most successful at predicting the target classes.

2 Data and Features

The datasets used in this project were obtained from Kaggle.com, and this data was originally used as part of a case study for a paper proposing a framework for semantic-enhanced data mining on sensor streams (Ruta et al., 2018). The case study refers to a prototypical system for monitoring road and traffic conditions, and one potential use of this system would be to improve the functionality of navigation systems with real-time driver assistance.

The four datasets were generated from two journeys in an Opel Corsa 1.3 HDi and two journeys in a Peugeot 207 1.4 HDi. The numeric data was collected through On-Board Diagnostics (OBD-II) ports and smartphone micro-devices, and each dataset contains 14 numeric features: **altitude change** (calculated over 10 seconds), **instantaneous speed**, **average speed** and **speed variance** in the last 60 seconds, **speed variation** for every second of detection, **longitudinal acceleration** and **vertical acceleration** which were measured by a smartphone accelerometer, **engine load**, **engine coolant temperature**, **manifold air pressure** (MAP, a parameter the internal combustion engine uses to compute the optimal air/fuel ratio), **mass air flow** (MAF) rate, **intake air temperature** (IAT) at the engine entrance, **engine revolutions per minute,** and **average fuel consumption**.

There are also 3 categorical features in each dataset: **road surface** ('Smooth Condition', 'Uneven Condition', or 'Full of Holes Condition'), **traffic** ('Low Congestion', 'Normal Congestion', or 'High Congestion'), and **driving style** ('Even Pace' or 'Aggressive'). This data was gathered by the drivers manually labelling the journey records with the different classes of each category. The road surface category was selected as the target class for this project. This category was chosen as it is more correlated to the numeric features than driving style, and there is a slightly better balance of its class labels compared to traffic, plus the minority class for road surface seems to be more correlated to other features (making it easier to predict) compared to the smallest class for traffic.

The four individual datasets were concatenated to create one dataset to use to train and test the models, but prior to this, they were compared to each other to explore similarities and differences between the journeys. Among the discrepancies, there was noticeable variance in the road surface conditions between the datasets. While the Opel trips were driven on mostly smooth roads, the two Peugeot trips were on considerably worse road surfaces, especially the second Peugeot trip where only 6% of the roads were in a smooth condition. This led to an imbalance

of the target classes when the datasets were combined, and of the total 23,775 target values, 60% were ‘Smooth Condition’, 26% were ‘Uneven Condition’ and only 13% were ‘Full of Holes Condition’.

3 Date Preprocessing

3.1 Splitting the Data

Before carrying out any preprocessing steps, the dataset was split into a training subset to be used to train the models and a testing subset to evaluate the models. As there was an imbalance of target class labels, the data was divided using a stratified train-test split. This type of split ensured that the train and test subsets had the same proportion of the three different class labels.

3.2 Missing Values

Of the almost 24,000 rows of data, less than 20 rows were missing values. It is not clear if these values were missing at random, however with such a small portion of the data missing, it was unlikely that this would significantly affect the models. And so rather than choose a more complicated computational method, the null values were simply filled in with the median value of the column. The median value was chosen rather than the mean as many of the data columns had skewed distributions.

3.3 Transforming and Scaling

Most of the numeric features had moderately to highly skewed distributions, so to give them more Gaussian-like distributions, the data was transformed using the Yeo Johnson method of scikit-learn’s PowerTransformer function. This power transform can be used on both positive and negative values, and after comparing it to other types of transforms (such as square root and log transforms), it was clearly the most successful at reducing the skewness of all columns.

The numeric features were measured in different units and had large differences in their ranges, so it was also necessary to transform the data into comparable scales, as algorithms like SVM perform better when the data is scaled. This is because the distance between data points affects how SVM chooses a decision boundary. Thus, the numeric data was standardized so that each variable had a mean of 0 and a standard deviation of 1.

3.4 One Hot Encoding

The categorical columns (traffic and driving style) were converted into numerical variables through One Hot Encoding. This function represents categorical variables as binary vectors, and so it transforms the data into a suitable input format for the SVM model.

3.5 Feature Selection

Training a model with irrelevant or redundant features can lead to overfitting, while selecting the most relevant features can increase the accuracy of models whilst also reducing computational time. So, to determine which features would be most useful for predicting the road surface condition, a correlation matrix was used to compare how correlated each feature was with each of the target labels. Additionally, a Random Forest classifier was used to rank the features by their importance to this model. Based on these results, the following features were chosen to train the models with: 'VehicleSpeedInstantaneous', 'VehicleSpeedAverage', ‘'VehicleSpeedVariance', 'LongitudinalAcceleration', 'EngineCoolantTemperature', 'ManifoldAbsolutePressure', 'EngineRPM', 'IntakeAirTemperature', 'VerticalAcceleration', 'FuelConsumptionAverage' and 'Traffic'.

4 Classifier Training

4.1 Support Vector Machine Classifier

The first multi-classification algorithm selected was a Support Vector Machine (SVM) model, which was implemented with scikit-learn's Support Vector Classification (SVC) class. SVM creates a decision boundary that separates different classes and finds the maximum marginal hyperplane (MMH) between them. As it is not possible to create multiclass SVMs natively, the 'one-versus-rest' classification method was used to split the dataset into multiple binary classification problems. This method distinguishes between one class label and all the others, and the class prediction with the highest probability wins.

GridSearchCV was used to train the SVM model by fitting it with four different types of kernels (linear, radial basis function, sigmoid and polynomial) and a range of hyperparameters: different values for C (the penalty parameter of the error term) for all kernels and different values for gamma (the distance a single training example exerts influence) for the radial basis function. While searching for the best parameters, GridSearchCV used 5-fold cross-validation to assess and compare each combination's out-of-sample accuracy.

The most effective combination for classifying the target data was a polynomial kernel of degree 3 with C = 1000 and gamma = 'scale', and the SVM model built with these optimal parameters achieved an overall accuracy score of 0.987.

4.2 Random Forest Classifier

The second algorithm selected to predict road surface was a Random Forest (RF) classifier. An initial analysis was performed using a single decision tree classifier to gain familiarity. This classifier, with an initial *max_depth* parameter set to 4, produced an accuracy score of 0.887. Its tree structure was converted to visual representation using the GraphViz library to aid in interpretability. An analysis of the optimal tree depth was undertaken to determine the optimal *max_depth* parameter to use. This ended up being 19, with an improved accuracy score of 0.989.

At this point, two RF classifiers were created using the two distinct *criterion* types, namely "gini" (impurity) and "entropy" (information gain) which are used to measure the quality of the tree splits. It was determined that the "gini" criterion has slightly better accuracy. An analysis of the number of estimators (trees in the forest) also demonstrated that higher accuracy could be achieved using a greater number of estimators at the expense of performance.

Finally, GridSearchCV was employed again to determine the best parameters for the model; this time using 3-fold cross-validation. This resulted in the following set {'bootstrap': True, 'max_depth': 80, 'min_samples_leaf': 3, 'min_samples_split': 8, 'n_estimators': 1000}, along with an overall accuracy of 0.992.

5 Results Comparison

After the models were fitted with the training data, confusion matrices, classification reports (see Table 1) and plots of AUC-ROC curves for each model were utilised to evaluate and compare their performance at predicting the target labels for the test set. Based on scores for precision, recall, F1-score, accuracy, and AUC (the area under the 'Receiver Operator Characteristic' curve), it was determined that the performance of the RF model was slightly better than that of SVM. Although both models achieved quite high scores, it is noteworthy that RF outperformed SVM across precision, recall and F1-score for the prediction of the 'Full of Holes Condition' road surface label, which had the smallest number of occurrences in the imbalanced dataset. However, the differences in overall

performance are not very significant, so it is likely that other model characteristics would need to be considered to select one over the other, such time and space complexity.

	Support Vector Machine			Random Forest			
Condition	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**No. of Class Occurrences**
'Full of Holes'	0.96	0.98	0.97	0.99	0.99	0.99	650
'Smooth'	0.99	0.99	0.99	0.99	1.00	0.99	2847
'Uneven'	0.99	0.98	0.99	0.99	0.99	0.99	1258
Macro Avg.	0.98	0.99	0.98	0.99	0.99	0.99	4755
Weighted Avg.	0.99	0.99	0.99	0.99	0.99	0.99	4755
Accuracy			**0.987**			**0.992**	4755

Table 1: Classification Report Comparison.

A random forest classifier was also used in the case study for which this data was originally collected; however, it only scored 0.879 for both its accuracy and F1-score. The best performing algorithm in that study was a J48 decision tree which scored 0.883 on accuracy and an F1-score of 0.884. Thus, both models created for this project outperformed these algorithms. This could be partly due to the traffic category being included as an input feature (for the case study, it was one of the target classes), and that study including two other datasets from another type of car. These differences may explain some of the variance in the results from this project and the original study.

6 Conclusion

To determine the optimal model for each classifier, the GridSearchCV function was used to perform exhaustive searches over their important parameters, while also using cross-validation to evaluate each combination's performance on unseen data. Although this is quite a time-consuming process, creating models using the best estimators from these searches has demonstrated that both classifier types can be trained to produce highly accurate results. And despite the imbalance of target classes, the models were very successful at predicting each of the different types of road surface conditions. However, to achieve an even higher F1-score for the smallest class in future experiments, there are a range of options which could be explored, such as integrating the models to produce a hybrid classifier to leverage the strengths of each model (for example, in the work of Du et al., 2012) or using an algorithm like the SMOTE technique to generate synthetic samples of the minority class.

7 References

[Du et al., 2012] Du, P., Xia, J., Chanussot, J. and He, X. (2012) *Hyperspectral remote sensing image classification based on the integration of support vector machine and random forest.* 2012 IEEE International Geoscience and Remote Sensing Symposium (pp. 174-177). IEEE.

[Nomura & Shiraishi, 2015] Nomura, T., & Shiraishi, Y. (2015). *A method for estimating road surface conditions with a smartphone.* International Journal of Informatics Society, 7(1), 29-36.

[Ruta et al., 2018] Ruta, M., Scioscia, F., Loseto, G., Pinto, A. and Di Sciascio, E. (2018). *Machine Learning in the Internet of Things: A Semantic-enhanced Approach.* Semantic Web Journal, Volume 10, Number 1, page 183--204.