Gesture-Based UI

Project 1: Classification Algorithms

Chayapol Hongsrimuang

March 2024

1 Introduction

In classification problems, there are numerous methods and algorithms that can be used to classify data based on the amount of features given to those methods. However, these methods often produce different results compared to each other as the level of complexity can differ from each other.

There are three main classification algorithms that are used (and covered in this module). Those being logistical regression, k-nearest neighbours (or kNN), and Support Vector Machine (or SVM).

Logistic regression (or logit model) is where the output is deemed as the probability score between 0 and (the amount of categories - 1). This algorithm makes use of the sigmoid function, where it produces a curve-like structure from 0 to (the amount of categories - 1). A decision boundary is then created for each probability threshold.

On the other hand, k-nearest neighbours (or kNN) algorithm is where the training data is first put into a scatter-like graph. Then, with the input, the distances between the input and nearest points are calculated. The top "k" nearest points are then used to classify what the input is. This is somewhat straight forward than logistic regression, as they do not require a concrete formula (like the sigmoid formula) to declare the model itself. Only the distance (typically Euclidean distance) is calculated.

And finally, the support vector machine is the combination of the two, where the data points are plotted together in an area, and the margins are declared as the decision boundaries. There are three main kernels that are used for this project, those being linear, radial basis function (RBF), and polynomial.

This report will be guiding through the steps of comparing the three algorithms mentioned above, as well as how they compare with each other using the sample data of a vehicle sensor.

2 Methodology

The first process of using the algorithms is to first edit and manipulate the data set that will be put into them. Firstly, the data is pre-processed to remove any unrelated columns that would not make sense for the data. All the data sources with the exact same columns are then combined into one huge group of data. These are then went through a removal of any rows that have an unknown value in them. While, there is another option of replacing those missing fields with median values, but they may correlate with each other in some way, and may not be suitable for a media replacement.

The data set is then split into two main categories, the inputs and the expected output. The inputs in this sense are numerical values in the data set. For the case of this data set, they are every column except for roadSurface, traffic, and drivingStyle. The expected output in this case would be

any of the three columns mentioned before. For this project, drivingStyle is used as the expected output column.

Downsampling is then done for the data set. This is where the data set are balanced (by removing excess data) according to the specified column that would be the output. This is to prevent overfitting and to prevent biases in favour of a category that has the most amount of data in them.

After which, the input values (X) are then normalised, to effectively be put in each algorithm. The expected output is also labelled to their numerical values, to be used in the classification tasks.

The data set is then split into training and testing data sets. This step is crucial for validating each model produced from algorithms with data that the model has not seen before during its training, and to evaluate the accuracy of said model.

The input values are then tested out with combinations of columns, to find the best scoring combination of columns based on the first algorithm of logistic regression. This is due to its simplistic implementation of the algorithm, which allows for fast execution with all the data set. Cross validation is also used in conjunction with finding the best combination of columns.

The column combination is then used to reduce the input columns of both training and testing set, further, allowing less overfitting of each model produced. Both data sets are then used for all algorithm models.

2.1 Algorithm-specific techniques

For k-nearest neighbours (kNN) algorithm, there are two approaches done when it comes to testing out the algorithm itself. First is the no k-Fold method, and second is with the k-Fold method. k-Fold is a cross validation technique that divides up the training data set further in k folds, with each different fold being the validation set of the training set for each execution. The number that is used in this case is 5 folds.

For support vector machine (SVM) algorithm, there are three different kernels to be tested on. Those being linear, radial basis function, and polynomial. (as mentioned earlier) The radial basis function will make use of "gammas", where it determines the distance influence of a single training sample. The gamma used in the project ranges from 0.001 to 1000. (and "scale") The polynomial kernel will make use of "degrees" as a way to produce more curve-type margins fitting into the model, in a similar sense to regression models. The degree used in the project ranges from 2 to 8.

All the kernels in SVM also make use of the "C", known as the regularisation parameter, with larger values of C resulting in harder margin. The C used in the project ranges from 0.0001 to 5.

3 Experiments and results

First, two execution tests were run for all algorithms to compare the effects of downsampling and without downsampling of data. (one without downsampling, and one with downsampling) The test were also run with all features of the numerical data set. The results are in Table 1.

These results, shown in Table 1, showed that downsampling can cause the algorithms to score lower than without downsampling. However, the biases without sampling can effect on future predictions of the models that have the expected output that is against the bias (i.e. belongs to the category that has a lower sample size than the others) This can be the effects of oversampling, if there's no downsampling of data, and the reason as to why data should be downsampled.

After the downsampling tests, five further execution tests are run and recorded. This time, each test will have its own best combination of columns to be used for the data set and includes downsampling of data. The execution test also includes the kNN model without any kFold mechanism,

Without downsampling vs. downsampling accuracy scores		
Algorithm	Without downsampling	Downsampling
Logistic Regression	0.8808	0.5775
kNN	0.9075	0.7952
SVM	0.9261	0.8422

Table 1: Comparison of accuracy score for downsampling data

to compare if kFold improves the accuracy score or not. The accuracy scores from all algorithms are recorded and calculated to find the average accuracy score of each algorithm. The results are shown in 2 and 1

Algorithms and Accuracy Scores				
Test	Logistic Regression	kNN (without kFold)	kNN (with kFold)	SVM
1	0.5188	0.7783	0.8030	8.0652
2	0.5920	0.7732	0.7823	0.7819
3	0.5957	0.7580	0.7668	0.7442
4	0.5239	0.7667	0.7598	0.7688
5	0.5196	0.75	0.7537	0.7529
Mean	0.55	0.7652	0.7731	0.7709

Table 2: Comparison of accuracy scores of different algorithms

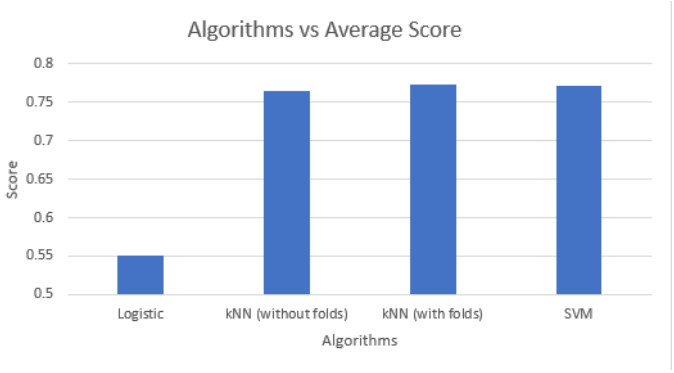

Figure 1: Bar chart from the algorithms

As seen in the results, the logistic regression tends to not be the best algorithm to be used for the prediction model, due to its low accuracy score. Meanwhile, kNN and SVM performed similarly with each other, with kFold kNN performing slightly better than without. However, there can be occasions where kFold may not be more accurate than without one.

For kNN, the best k number that produces the best "valid" testing score is usually around 1-3, with sometimes rarely being in the higher values. However, despite this, the training accuracy score is fairly high for that k value as well. This is indicated in one of the tests in Figure 2.

Figure 2: Graph from the fourth execution test for kNN (with kFold)

For SVM, the RBF kernel tends to perform better than the other kernels of linear and polynomial. The best "gamma" values used were either 1 or 10 for the kernel, while the "C" variable is between 1 and 5 (which is the maximum for these tests)

There were also an observation that was made regarding the execution of each test, where fewer columns can cause the runtime of the SVM algorithm to be significantly higher. This is especially true with higher degrees of polynomial kernel, with more curve fitting.

4 Conclusion

With classification problems, three dominant algorithms are used. Those being logistic regression, k-nearest neighbours, and source vector machine. From those algorithms, k-nearest neighbours and source vector machine tend to produce models that perform well in accuracy score, with k-nearest neighbours producing the best result.

For k-nearest neighbours, the first couple of k values (1-3) can produce the best valid accuracy scores for the data. For the source vector machine, the radial basis function tends to be the best kernel to be used for the model itself.

With the data set itself, there are various pre-processing techniques that have to be done, from normalisation to downsampling of data that would be used to train and validate the model. This is crucial to remove any chances for overfitting and to make the model more accurate at the same time.

There are also many more ways to approach this classification problem, including neural networks with probabilities for each category, and determine the best category based on those values. This approach will make use of other mathematical functions to help with the neural network itself. This includes Sigmoid, hyperbolic tangent functions, and many more. These networks may also produce better results than those algorithms as well, depending on the topography of the network itself.

Overall, k-nearest neighbours model is the best algorithms out of the three that is used for this assignment, with lower values contributing to more accuracy scores from the model.