Machine Learning Task 1 Report

Name: Han Tang
Student No.: D16129273
Programme Code: DT228A DA

1 abstract

This report is meant to illustrate the process of building a predictive machine learning model for the task1 of the Machine Learning assignment.

2 Choose Competition

The Kaggle Competition I choose is ‘ Stay Alert! The Ford Challenge ’ (https://www.kaggle.com/c/stayalert). The objective is to design a classifier that will detect whether the driver is alert or not alert, employing data that are acquired while driving.

2.1 Dataset

There are 604,329 instances of data in the training dataset and 120,840 instances of data in the test dataset. The data for this challenge shows the results of a number of "trials", each one representing about 2 minutes of sequential data that are recorded every 100 ms during a driving session on the road or in a driving simulator. The trials are samples from some 100 drivers of both genders, and of different ages and ethnic backgrounds. The files are structured as follows:

The training data was broken into 500 trials, each trial consisted of a sequence of approximately 1200 measurements spaced by 0.1 seconds. Each measurement consisted of 30 features; these features were presented in three sets: physiological (P1...P8), environmental (E1...E11) and vehicular (V1...V11). Each feature was presented as a real number. For each measurement we were also told whether the driver was alert or not at that time (a boolean label called IsAlert). No more information on the features was available.

3 Existed Work

As a Kaggle competition ended 4 years ago, many models are built to solve this problem. I researched methods in the discussion sections of the competition and other reports or papers attempt to solve this problem In order to summarize existed work and formulate a plan in order to build an outperformed machine learning predictive model.

Similar machine learning techniques are applied to this dataset. The techniques most participants used limited to Nave Bayes, Logistic Regression, Support Vector Machine, Neural Network, and Random Forest. But the performances of their models are totally different, as they preprocessed the original data in different ways, especially in their feature engineering.

Thus I will mainly focus on the feature engineering methods applied by the participants, instead of how they choose parameters of algorithms in the summary part.

3.1 Review of existed work

The highest score (AUC = 0.861151) was reached by a logistic regression model. As the dataset consists of sequential data recorded every 100 ms for 2 minutes in each trial, the participant partition the data by trials (TrialID) rather than randomly partition. The Means and Standard Deviations of each trial were computed as new features (include the target feature IsAlert). Afterwards, feature selection based on diagnostics of the logistic regression was conducted and three strong features were chosen for modelling (sdE5, V11, and E9). However, this model applies future observation (The mean and standard deviation can only be calculated when a trial is finished), thus inapplicable for real-life situations. A running Mean and Standard deviation were applied to training instead and the AUC has dropped slightly, from 0.861151 to 0.849245).

Another participant focuses on the instances at the initial moment the driver lost alertness, the dataset is reduced significantly in this way and he highlighted the factors change significantly between status change for feature selection. E4, E5, E6, E7, E8, E9, E10, P6, V4, V6, V10, and V11 are selected for building a Neural Network (the architecture is not mentioned). This model reaches an AUC of 0.84953.

Participant Tariq also attempts to aggregate data from each trial and calculate means and standard deviations as additional features. After tossing up correlated features and other feature engineering, a logistic regression model trained from feature selected data reaches an AUC of 0.80779.

Participant Louis Fourrier generates around 600 new features to the dataset (The inverse, the square, and the cube of each features, all the combinations of 2 columns, time interval variables). He reaches the his highest AUC by applying forward search to select predictive features. A Nave Bayes model trained by these selected features reach an AUC of 0.844.

Participant Junpei Komigama trained a epsilon-SVR, RBF kernel model with parameters c = 2, g = 1/30, and p = 0.1, which reaches an AUC of 0.839.

Participant Jaysen Gillespie applies a random forest with 199 trees and min node size of 25, the correlated features are tossed out beforehand. This predictive model reaches an AUC of 0.81410.

3.2 Summary of existed work

An important feature for this dataset is that it contains sequential data. For each trial, the dataset records data every 100ms. Thus all the participants shuffle the dataset by trials for the purpose of preserving this sequential feature.

Aggregating data within a trial to generate means and standard deviations as new features for modelling is proofed as a useful method of data preprocessing. Another useful method of data preprocessing is to choose the instances close to the moment the driver lost alertness, which reduce time to train the models significantly.

Multiple methods of feature selection are applied, the mean/standard deviation of existed features, inverse, the square, the cube, and a combination of 2 columns are viewed as potentially useful new features. Correlated, remain constant features are always tossed out.

As for the choice of predictive machine learning algorithms, there is no valid proof that one algorithm outperforms all the others in this specific situation. Generally, Nave Bayes, Logistic Regression, Random Forest, Support Vector Machine, and Neural Network all reach a good performance in this case.

4 Model Building Plan

Even though many existed models have already had a decent performance, It's still possible to improve the model. A plan for building a new predictive model is outlined in this section.

4.1 Gap Identification

The predictive model with the highest AUC value is trained from 20% of the training dataset. What's more, the means and standard deviations of each trial are future observation features. Those make this predictive model inapplicable to a real-life situation. An AUC value of 0.861151 also means there are still rooms for improvement.

Another noticeable point within most of the existed work is that most of the models are evaluated by either AUC score or classification accuracy. For this specific situation, it's obviously more important to identify those not alert instances as driving while not alert can be deadly. Failing to identify 'not alert' can lead to worse consequences compare to failing to identify 'alert'. Thus true negative rate (TN / (TN + FP)) can also be a valuable measure of evaluation as it shows the percentage of 'not alert' instances successfully identified.

Furthermore, As all the models' classification accuracies are above 50%, which makes building an ensemble model to reach a better performance possible as if the recalls and the specificities of all the models can reach above 50% at the same time for all the models.

4.2 Model Building Plan

Firstly, those existed models with good performance will be reproduced, includes the way they preprocess the data and the parameters they choose to build predictive models.

Secondly, a local evaluation will be conducted on these models. The recalls and specificity will be used for evaluation, apart from classification accuracy and AUC score. Then to group those models with recall and specificity both higher than 50% to build an ensemble model, aims to reach a better performance than all the existed models.

5 Solution Development

Python is used as the developing environment for this project. Scikit-learn is the machine learning tool applied.

Missing data is identified as 0 throughout all the dataset.

5.1 The First Model

The first predictive model is built by the data preprocessing method of Mick Wegner, whose model reaches an AUC of 0.84953, at the fourth position in the leaderboard.

He was concerned that using the entire data set would create too much noise and lead to inaccuracies in the model. The final goal of the system is to detect the change in the driver from alert to not alert so that the car can self-correct or alert the driver. So he decided to just focus on the data at the initial moment when the driver lost alertness.

According to this, I subset the dataset to the moment when the driver lost alertness. The rows with the feature 'IsAlert' == 0 and the last rows with the feature 'IsAlert' == 0 are chosen, along with 5 rows before and after each (100ms of time between each observation, 5 rows before means focus on the data recorded 0.5s before and after the driver lost alertness).

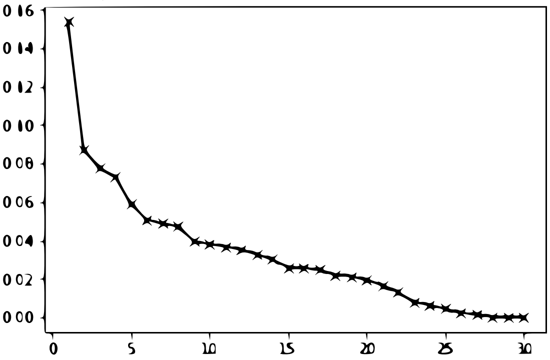
After subsetting, 37421 instances without duplication are chosen to build the predictive model.

5.1.1 Feature Engineering

There are 30 features included in the dataset, thus filter those features with higher impact could not only save computational resources but also potentially improve the performance of the predictive model. Principle Component Analysis (PCA) is applied as the feature engineering technique in this case.

For PCA, the dataset is standardized firstly, then the fraction of variances of each feature is calculated to identify those features have higher impact on the result.

図 1: Scree Plot of 30 features



We can see that the first 14 attributes contribute 80.95% of the total variance, the number of features selected for modelling is decreased from 30 to 14 in this way.

5.1.2 Modelling

As the size of subset is relatively smaller, stratified 10 fold cross-validation is applied as data evaluation method to make full use of the dataset. Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine, and Neural Network models are built from this dataset.

Gussian Nave Bayes model performs an validation accuracy of 61.74%. Logistic Regression with optimization algorithm of 'liblinear' reaches an validation accuracy of 64.6%. Multiple models in Support Vector Machine family, include Linear-SVC, Nu-SVC, C-SVC are applied as well. Their validation accuracies varied from 65% to 78%

A Neural Network with 5 neurons in the first hidden layer and 2 neurons with the second hidden layers reaches a validation accuracy of 65.01%.

I tried to use neural networks with different architectures, another neural network with 5 hidden layers and 14, 14, 12, 10, 5 neurons in each layer. The activation function is also changed from RELU to logistic regression. Unfortunately the performance of the new neural network does not change mach.

Model	Accuracy	recall	Specificity	AUC sc
Logistic Regression	64.60%	94.32%	25.25%	0.5978
Nave Bayes	61.74%	89.47%	25.02%	0.5724
RandomForest	93.54%	97%	90.08%	0.9354
Linear-SVC	64.55%	94.85%	24.44%	0.5958
Nu-SVC	78.63%	92.36%	60.45%	0.7640
C-SVC	67.55%	98.13%	27.06%	0.6260
Neural Network1	65.01%	94.31%	26.21%	0.6026
Neural Network2	65.96%	97.02%	24.83%	0.6092

Performances of algorithm on PCA dataset

It can be found from the performance diagram that all the models perform pretty well on predicting those drivers 'in alert'. However, most models cannot reach a decent result when it comes to identifying drivers not in alert, which is more important in this specific situation.

On the other hand, the RandomForest model outperform all other models, especially when it comes to specificity, which makes it a part of our final ensemble model. The Nu-SVC model reaches a specificity of more than 50% as well, which means it can also be part of an ensemble model.

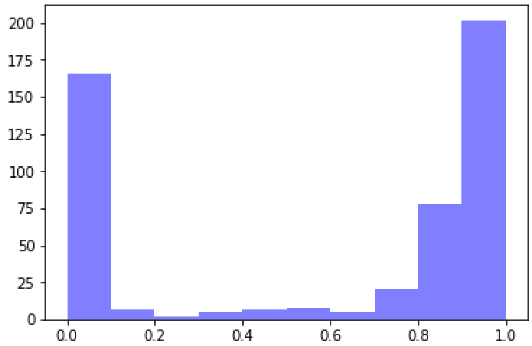
5.2 The Second Model

Unfortunately, we did not get a good predictive machine learning model by the first data preprocessing method (apart from the RandomForest with 50 trees model). I decided to conduct an exploratory analysis on the dataset in order to provide a guidance of data preprocessing.

5.2.1 Explarotary Analysis and feature engineering on the dataset

I calculate the average IsAlert value per trial and plot the result on a histogram.

図 2: Histogram of mean alertness per trial



It is found that for most drivers, they either stay alert or not alert throughout the 1200ms trial. Thus the characteristic of each driver, recorded in the mean and standard deviation of each attribute, can be helpful for predictive analysis.

On the other hand, it is impossible to get the mean and standard deviation of a trial at the beginning of each trial, which makes using stable means and standard deviations of each feature unpractical in real-life situation. Moreover, using stable means and standard deviations cannot record the change of the driver's behavior within a trial, which may be constantly changing overtime.

For these reasons, I decided to use rolling means and standard deviations of each features as new features instead of simply using stable means and standard deviations in order to make full use of the sequential feature.

The rolling window is set to 5, as for every 5 instances (500ms), it calculates the mean and standard deviation for them, then the algorithm drop the first instance and add a new instance, etc.

5.2.2 Modeling

Similarly, I applied algorithms mentioned above to this preprocessed dataset. As the size of the dataset is big enough, I use 80%-20% to train-test split the dataset instead of cross-validation.

I firstly tried RandomForest algorithm, the one performs the best in the last feature selected dataset, to see if there's any improvement compare to the other feature selecting method. The RandomForest has 50 trees, the parameters are the same as the one applied before. It reaches a decent performance on the validation dataset, with a validation accuracy of 98.91%. Algorithms of the Support Vector Machine family all fail to converge within a specific period of time. A neural network with four hidden layers, each layer has 90, 70, 50, 30 neurons respectively also applied, reaches a validation accuracy of 80.76%. Furthermore, Nave Bayes and Logistic Regression have not improved much compare to the preview models. Generally, Neural Network and RandomForest performs better than other models in this situation, and RandomForest performs far better than Neural Networks.

Model	Accuracy	recall	Specificity	AUC sc
Logistic Regression	61.21%	75.21%	41.96%	0.5858
Nave Bayes	62.86%	45.28%	87.02%	0.6615
RandomForest	98.91%	98.58%	97.55%	0.9873
Neural Network	80.76%	96.40%	59.26%	0.7783

Performances of algorithm on rolling mean std dataset

5.2.3 Comparison of models

Comparing the performances of models trained from data preprocessed by different methods, it is found that algorithms logistic regression, Support Vector Machine, and nave bayes are not suitable for this problem. While Neural Network can reach a good performance in the dataset preprocessed by generating time sequential feature, it is not the model fits the dataset the best. The Random Forest Algorithm generates the best result on predictive analysis, either trained from data preprocessed by PCA or from data preprocessed by other feature engineering techniques. Another interesting finding is that most models perform better when it comes to predicting 'alert' drivers than to predicting 'notalert' drivers, apart from the Nave Bayes model. Considering two values are basically equally distributed (alert: 349785, notalert: 254544), it's hard to say one label is over represented than the other, which makes the unbalanced predict result hard to explain.

As a result, I choose three models for local evaluation, which are two RandomForest models and a Neural Network Model.

6 Local Evaluation

This competition ended 8 years ago, and the solution dataset has already been publicated. I use the data 'solution.csv' to evaluate the final models.

6.1 Model 1

The first model is the RandomForest trained by the data with features selected from PCA.

	Predict = 0	Predict = 1
Actual = 0	22571	7343
Actual = 1	63616	27310

AUC = 0.52744

Though this model only reaches an accuracy of 41.28% on the test dataset, it identifies many not alerted drivers correctly. Overall, this model is not good enough, no matter evaluated by which method.

6.2 Model 2

The second model is the RandomForest trained from the data with added features of rolling mean and standard deviation.

	Predict = 0	Predict = 1
Actual = 0	16671	13243
Actual = 1	8679	82247

AUC = 0.7309

This model reaches a good performance, with classification accuracy of 81.86% on the test data. It has a good performance in predicting alert drivers, with recall = 90.45%, precision = 86.13% and F1-score = 88.24%.

However, for this specific situation. The model is expected to predict 'not alert' drivers precisely, specificity (The percentage of Actual = 0 is predicted correctly) should be the evaluation method we focus on for this reason. The specificity of this model only reaches 55.73%, which still has lots of room to improve.

Overall, the AUC value of this model is 0.7309, not as good as the work I referenced, but still an improvement.

6.3 Model 3

The third model is the Neural Network trained from the data with added features of rolling mean and standard deviation. It has four hidden layers with 90, 70, 50, 30 neurons in each layer.

I was meant to use the first layer to grab all the original features and the coming layers to process and predict the output, thus the number of neurons for the first layer is as many as the number of the features.

	Predict = 0	Predict = 1
Actual = 0	12886	17028
Actual = 1	361	90565

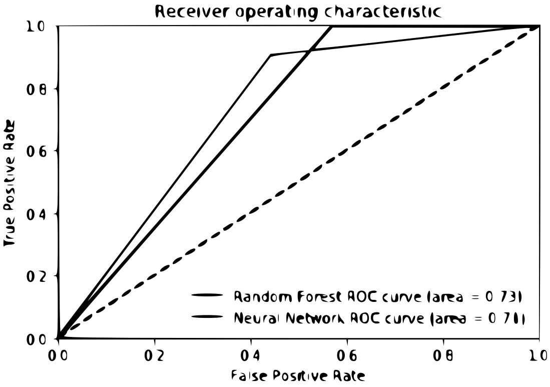
AUC = 0.71340

This model reaches a good performance as well, compares to the first model. It successively predict most of alert drivers (recall = 99.6%, precision = 84.17%, F1-score = 91.24%). However, the model fails to predict many not alert drivers correctly (specificity = 43.08%), which is the more important evaluation method for this predictive model.

6.4 Comparison of Model 2 and Model 3

Both two models perform better on predicting alert drivers than identifying not alert drivers as the true positive rate are both higher then their true negative rate in their confusion matrix , though the main goal of this predictive model is to predict not alert drivers.

図 3: ROC curve of two models



The curve reaches 100% true positive rate firstly is the neural network, the other curve is the random forest. It also can be found that the random forest model performs better than the neural network model. However, the neural network predicts most alert drivers correctly and when it predicts a driver as not alert, it's correct at the most of times. The random forest model reaches a significantly higher result on identifying not alert drivers from all the drivers than the neural network model, though when it predicts a driver as not alert, it gets 34.25% chance of being wrong.

The Random Forest model would be a better choice in this situation, but the architecture of the neural network model can be optimized to reach a higher performance.

7 Result Reflection and Comparison

7.1 Result Conclusion

This project was meant to build a supervised learning model to predict not alert drivers, the model with the best performance is achieved by Random Forest with 50 trees in it. It predicts 16671 of 29914 not alert drivers correctly in the test data. It reaches a classification accuracy of 81.86% and AUC value of 0.7309.

7.2 Result Comparison

When it compares to the results of those in the leaderboard, there are lots of participants' models reach a higher performance. The best model reaches an AUC value of 0.86115, though it applies means and standard deviations of each trial as new features. Almost 20 participants' models reach AUC scores over 0.8, which is significantly higher than mine. Refers that there is still large room to improve my model.

7.3 Discussion and Future work

The existed predictive model for this problem is far from perfection. There are a few perspective that can improve the performance of the model.

7.3.1 Data preprocessing method

The rolling means and the standard deviation is proved to be a good method to preserve the sequential attribute of the data. However rolling means for every 0.5s could be too short to grab the driving pattern of a driver. Expand the rolling window to produce rolling means and standard deviations in a longer period could be considered as a useful method to introduce the long-term driving patterns of drivers. Better performance is believed can be achieved by model learns not only from drivers' behaviours in a short time (0.5s) but in a long time as well.

7.3.2 Model Optimisation and selection

Though the neural network model fails to produce a better performance than the random forest model, it is still not convincing that random forest is always the best option for this problem. Neural network still shows great potential to produce good result. Further work could to optimize the architecture of neural networks.

8 Reference

参考文献

[1] J. Gillespie, "13th place options methods/tips from non-top 3 participants," 2011. [Online]. Available: https://www.kaggle.com/c/stayalert/discussion/328

[2] J. Komiyama, "Junpei komiyama on finishing 4th in the ford competition," 2011. [Online]. Available: http://blog.kaggle.com/2011/03/16/junpei-komiyama-on-finishing-4th-in-the-ford-competition/

[3] L. Fourrier, F. Gaie, and T. Rolf, "Junpei komiyama on finishing 4th in the ford competition," 2011. [Online]. Available: http://blog.kaggle.com/2011/03/16/junpei-komiyama-on-finishing-4th-in-the-ford-competition/

[4] M. Wagner, "Mick wagner on finishing second in the ford challenge," 2011. [Online]. Available: http://blog.kaggle.com/2011/04/20/mick-wagner-on-finishing-second-in-the-ford-challenge/

[5] inference, "First place in the " stay alert! " competition," 2011. [Online]. Available: http://blog.kaggle.com/wp-content/uploads/2011/03/ford.pdf

[6] T. bin Tariq and A. Chen, "Stay alert! the ford challenge," 2012. [Online]. Available: http://cs229.stanford.edu/proj2012/ChenTariq-StayAlert.pdf