Machine Learning Techniques Final Project

Team Members：蕭啟湘 (B08902142) 侯宇謙(B08507012) 郭家瑜(R08922A14)

Abstract

In the project, we aimed to explore the model with best prediction to certain day's label. After removing some less related variables, resetting the adr, finding the relationship between the labels and adr, we've used 3 models: LR (L2-regularized logistic regression (LR), random forest (RF), and the neural network (NN) models, for training the dataset.

Platform used: python 3

Data cleaning

We cleaned the data and removed some features as Table 1 demonstrates.

Table 1. Removed Variables & Corresponding Reasons

Variables removed	Reason of Removal
ID, arrival_date_week_number	Little relation with adr
country, reserved_room_type, booking_changes, deposit_type, agent, company	May increase VC dimension too much as to decrease model performance
is_canceled,reservation_status,reservation_status_date	No corresponding data in the test set

Feature engineering

We transformed certain features as Table 2 demonstrates.

Table 2. Feature Transformation

Features	Transformations
is canceled, arrival_date_year, arrival_date_month, arrival_date_day_of_month	Stored in date format Ex. 2015, July, 1→ [2015, 7, 1]
hotel, meal, market_segment, distribution_channel, assigned_room_type, customer_type	Stored in hot encoding Ex. hotel = { "Resort Hotel" : [1,0] , "City Hotel" : [0,1]}
adr	If the item "is_cancel"=1, then its adr is reset to 0
Other data	The same as the original data

Data Processing

1. We added up all data that are in the same day because what we want to predict is the total adr in a single day instead of the individual adr for each customer. We add up all data by the following methods:

- total_daily_adr = sum(adr * (stays_in_weekend_nights + stays_in_week_nights))
- others data: directly added all the data in the same day

2. We explored the relationship between total_daily_adr and the label. After trying different methods including neural networks and regression, we found that the most accurate formula is: label = floor(total_daily_adr/1000). For example, when the total_daily_adr=15000, then the label is 1. Thus after predicting the total_daily_adr, we can get the value of the label by floor(total_daily_adr/1000)

L2-regularized logistic regression (LR)

1.Algorithm

Here we chose an approach of using the LIBLINEAR package developed by NTU introduced in the previous homework, solving the machine learning problem with L2-regularized logistic regression.

$$\mathbf{w}_\lambda = \arg\min_{\mathrm{w}} \frac{\lambda}{N}||\mathbf{w}|| + \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + \exp(-y_n \mathbf{w}^T x_n)\right)$$

2.Parameters

I chose the critical parameter $\mathrm{C} = 1/2\lambda$ by calling the "−C" function which calculates the best parameter for training. Other parameters are "−s 0" for L2-regularized logistic regression and "−e 0.001" for the tolerance of termination criterion.

(1) Adding up adr of the same day before training

Best parameter C calculated is Best C = 3.63798e-12

(2) Train each adr respectively and sum up according to each day

Best parameter C calculated is Best C = 0.00195312

3.Model training results

(1) Adding up adr of the same day before training ⇒MAE = 1.155844

(2) Train each adr respectively and sum up according to each day ⇒MAE = 1.480519

Random Forest (RF)

1. Algorithm
   Here the random forest regressor and its python package (sklearn.ensemble.RandomForestRegressor) were chosen. As taught in the course, random forest is the combination of decision tree and bagging, and has the advantages of less likelihood of overfitting.
2. Parameters
   - The parameters were based on the model of estimating the total adr in one day, instead of the model predicting each customer's adr (e-Appendix 1.)
   - Default parameters in the "RandomForestRegressor" package (e-Appendix 2)
   - There are 2 kinds of parameters adjustment in the model.

   (1) "Hand-crafted parameters": The parameters are modified manually.
   (1-1) The parameters adjusted: n_max_dept, n_estimator, criterion.
   (1-2) Data scaling was performed before model training.
   (2) "Tuning of hyper-parameters"(by using the package of random search cross validation in scikit-learn)
   (2-1) parameters of 3-fold cross validation
   (2-2) parameter of 5-fold cross validation:
3. Model training results
   (1) The best performance of the RF model and the related parameters were listed in Table 1. (For the performance & parameters of other models, please refer to e-Appendix 3.)
   (2)

Table 1. The best performance of the RF model and the related parameters.

Type		Parameters	Best MAE (public score)	Private score
Hand-craft parameters	No data scaling	n_estimators=20	0.789474	0.779221
Hyper-parameters	3-fold CV	bootstrap: False, max_depth: None, max_features: sqrt, min_samples_leaf: 2, min_samples_split: 5, n_estimators: 1400	0.894737	0.701299
	5-fold CV	bootstrap: False, max_depth: None, max_features: sqrt, min_samples_leaf: 1, min_samples_split': 2, n_estimators': 400	0.881579	0.701299

Abbreviation: CV = cross validation

(3) Important Variables in the best model (n_estimators=20) (Figure 1)

- Circled features in Figure 1 were described as follows sequentially: stays_in_weekend_nights, stays_in_week_nights, adult, children, babies, total_of_special_requests

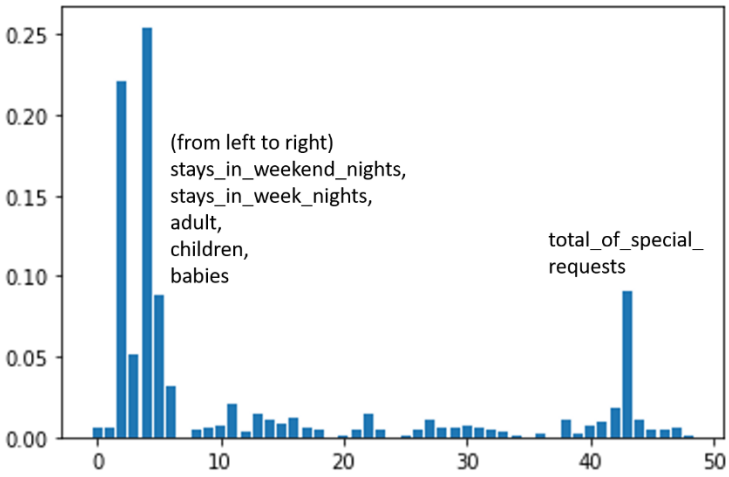

Figure 1. 'variable importance plot'(model: n_estimators=20, others as default parameters)

Neural Network (NN)

1. Algorithm

   Here we used keras package to implement the neuro network. Though the usage of keras package isn't taught in class, it's believed that the NN model should perform well.

2. Parameters

   1 input layer (numbers of nodes = data size) 1 output layer (1 node)
   2 hidden layers (each layer: 16 nodes)
   activation function : Rectified Linear Unit (ReLU)
   optimizer: Root-Mean-Square prop (RMSprop)
   loss function : mean square error (MSE)
   epoch = 10 batch_size = 1

3. Performance of the training models (Table 2)

Table 2. Best Performance of NN Model Via Different Approaches

Approaches	MAE (public score)	MAE (private score)
Approach 1: delete all cancelled data, training each adr respectively → adding up the data in the same day	1.578947	1.584416
Approach 2: preserved all cancelled data; switch the adr of the cancelled data to be 0 * predict every data's adr under the risk of being canceled	1.342105	1.376623
Approach 3: reserve all data, adding the data in the same day in advance→ predict the total_adr in one day	0.763158	0.714286

The Recommended Best Models

- Concerning the **MAE public score,** we would recommend to use Neuro Neural Network (best MAE: 0.76)> Random Forest (0.78) > L2-regularized logistic regression (1.48).
- Though the best performance of RF is similar to that of NN, the average MAE of RF (≈ 0.85), significantly poorer than the MAE of NN (≈ 0.78).

1. Pros of NN
   (1) High availability
      - There are many powerful, free, and easy packages of NN to be utilized on the Internet
   (2) Low threshold in application
      - The NN model could be built easily within 10 lines.
      - Even the builders have no idea about the principle behind NN, the NN model may still perform well.
   (3) Flexibility
      - All of the numerical data could be trained for classification or regression in NN.
   (4) Good model performance
      - The NN model is good at non-linear models unlike some traditional ML methods..
2. Cons of NN
   (1) Time-consuming
      - NN models in this project usually take ≥ 5 minutes to be performed. In contrast, the LR and RF model could be easily done in half minute.
   (2) Hardware requirement
      - To be more efficient in training NN models, high level hardware requirement such as GPU is often required.
   (3) Complexity of the parameters
      - There are too many parameters that should be adjusted, such as the numbers of layers, the number of nodes in each layers, and the activation function, etc.
   (4) Overfitting
      - NN model typically needs a lot of data. Without enough data, overfitting is common.
   (5) Weak interpretability
      - The NN model in this project couldn't provide meaningful interpretation. However, more and more NN for better explainability have been developed recently.

Comparison of the Models

Table 4. Comparison of the Models			
	LR	**RF**	**NN**
Efficiency	Fast < 0.5 min	Fast < 0.5 min * Slow in training hyper-parameters * Random search, 3 & 5-fold CV: 1-4 hours * Grid search 5-fold CV: > 4 hours	Slow > 5 mins
Scalability	Fine	Fine	Fined
Popularity	+++ LR is commonly used in the past, but we're uncertain about current status.	++++ RF is popular & commonly used in the traditional ML approaches.	+++++ NN is super popular in recent years due to outstanding model performance.
Inter-pretability	+++ Good interpretability with the available coefficient.	++++ Good interpretability with more apparent variables. Notably, correlation doesn't equal to causality. So the data should be interpreted cautiously.	+ NN, so called "black-box", has been criticized for low interpretability. Nowadays more and more theory and explainable NN, eg, adaptive explainable NN.
Abbreviations			

Q: How we balance the workload?
A: Each of us train one model respectively, LR-侯宇謙, RF-郭家瑜, NN-蕭啟湘. We discussed how to preprocess the data and write the reports together.

Acknowledgement
We sincerely thank professor Lin for providing the course and guidance, all the TAs for the support and assistance of answering lots of questions, and the classmates in the class for the inspiring and heated real-time discussion in class or the novel questions on NTUCool. All of the above mentioned have helped us in completing the project work :-)

Reference (e-Appendix)
https://drive.google.com/file/d/1OkeD8GGPB2bJ_wlEWDja8nd3Wdu63kVH/view?usp=sharing