Project on Customer Churn Classification

Chen Wang

JAN 2020

1 Objective

This project focuses on predicting whether a customer will change telecommunications provider, something known as "churning". The benefits of customer churn prediction lie in several aspects as follow:

- Help organization to gain a better understanding of future expected revenue.
- Target individuals and prevent them from churn.
- Cost of customer acquisition is five times higher than that of customer retention.
- Help organization to identify and improve the poor area of customer services.

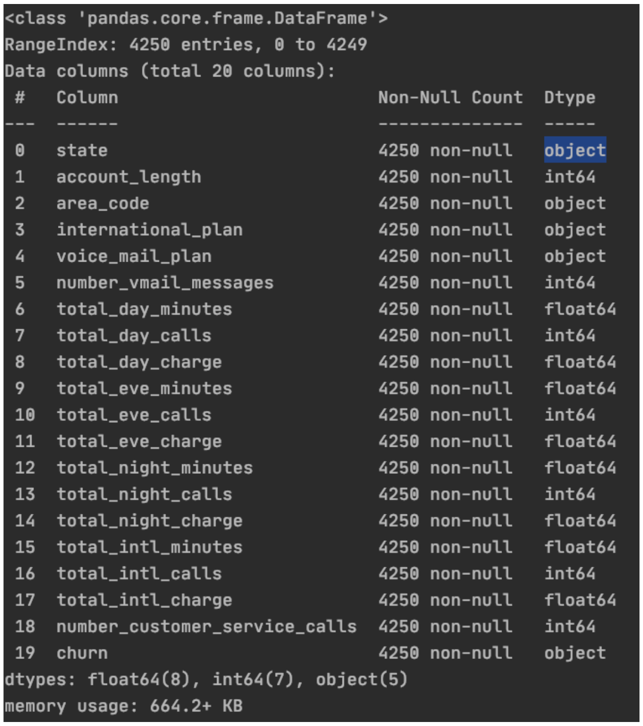
2 Data understanding

The training dataset is public available on Kaggle competition 2020 for customer churn prediction, and it contains 4250 samples. Each sample contains 19 features and 1 boolean variable "churn" which indicates the class of the sample. The 19 input features and 1 target variable are:

- "state", string. 2-letter code of the US state of customer residence
- "account_length", numerical. Number of months the customer has been with the current telco provider
- "area_code", string="area_code_AAA" where AAA = 3 digit area code.
- "international_plan", (yes/no). The customer has international plan.
- "voice_mail_plan", (yes/no). The customer has voice mail plan.
- "number_vmail_messages", numerical. Number of voice-mail messages.
- "total_day_minutes", numerical. Total minutes of day calls.
- "total_day_calls", numerical. Total number of day calls.

- ”total_day_charge”, numerical. Total charge of day calls.
- ”total_eve_minutes”, numerical. Total minutes of evening calls.
- ”total_eve_calls”, numerical. Total number of evening calls.
- ”total_eve_charge”, numerical. Total charge of evening calls.
- ”total_night_minutes”, numerical. Total minutes of night calls.
- ”total_night_calls”, numerical. Total number of night calls.
- ”total_night_charge”, numerical. Total charge of night calls.
- ”total_intl_minutes”, numerical. Total minutes of international calls.
- ”total_intl_calls”, numerical. Total number of international calls.
- ”total_intl_charge”, numerical. Total charge of international calls
- ”number_customer_service_calls”, numerical. Number of calls to customer service
- ”churn”, (yes/no). Customer churn - target variable.

Information of attributes is shown in Fig. 1



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4250 entries, 0 to 4249
Data columns (total 20 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   state                          4250 non-null   object
 1   account_length                 4250 non-null   int64
 2   area_code                      4250 non-null   object
 3   international_plan             4250 non-null   object
 4   voice_mail_plan                4250 non-null   object
 5   number_vmail_messages          4250 non-null   int64
 6   total_day_minutes              4250 non-null   float64
 7   total_day_calls                4250 non-null   int64
 8   total_day_charge               4250 non-null   float64
 9   total_eve_minutes              4250 non-null   float64
 10  total_eve_calls                4250 non-null   int64
 11  total_eve_charge               4250 non-null   float64
 12  total_night_minutes            4250 non-null   float64
 13  total_night_calls              4250 non-null   int64
 14  total_night_charge             4250 non-null   float64
 15  total_intl_minutes             4250 non-null   float64
 16  total_intl_calls               4250 non-null   int64
 17  total_intl_charge              4250 non-null   float64
 18  number_customer_service_calls  4250 non-null   int64
 19  churn                          4250 non-null   object
dtypes: float64(8), int64(7), object(5)
memory usage: 664.2+ KB
```

Figure 1: Information of attributes in the dataset

Correlation between each feature and target variable is shown in Fig. 2

To complete this analysis, I will do data cleaning and feature engineering, afterward, features will be used to train a suitable machine learning models with hydrometers turnings.

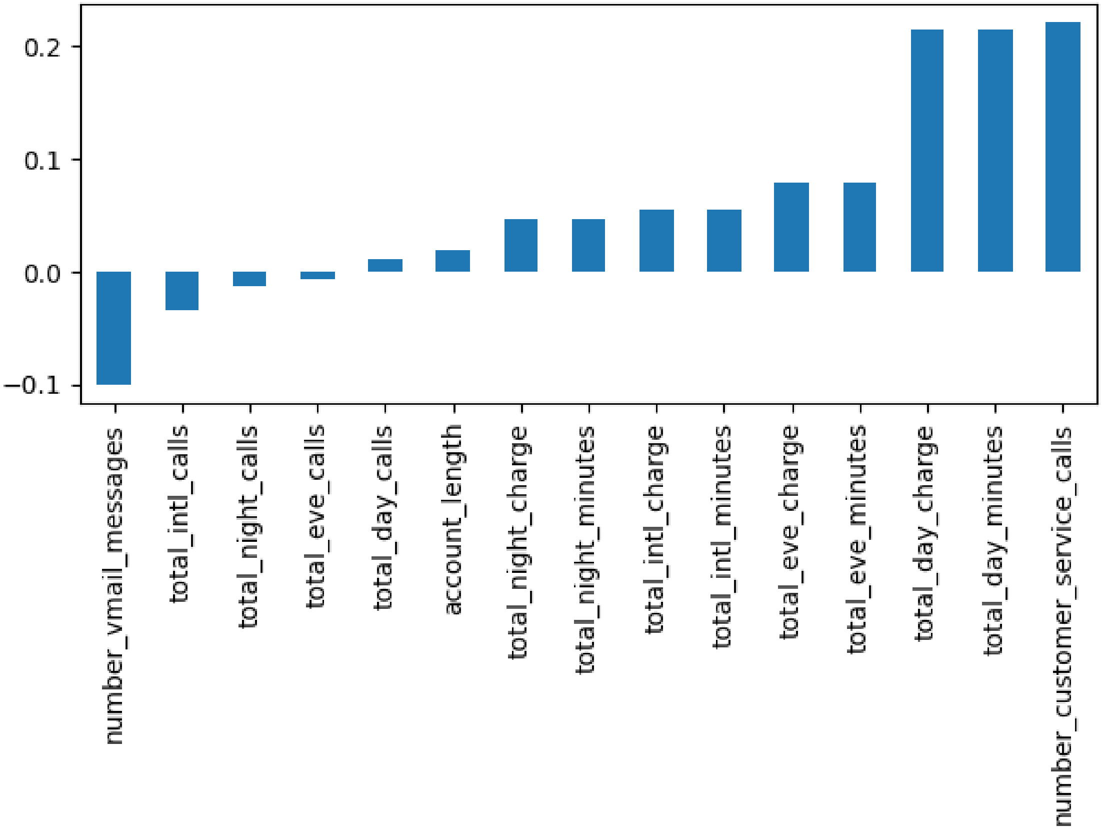


Figure 2: Correlation between each feature and target variable in the dataset

3 data cleaning and data exploration and feature engineering

3.1 data cleaning

- Split dataset into training and testing using stratified splitting
- Check missing data, inconsistent, typo, outliers
- Check the scale of both features and outputs, and apply standardScaler
- Check the skew of numerical features, and apply log1 transformation
- Binary and One Hot Encoding transformation for categorical attributes
- Check the imbalance or not, consider upsampling or down sampling

3.2 data exploration

Fig 3 and Fig 4 show the customer churn curve over months. In particular, customers with international plan are more likely to churn.

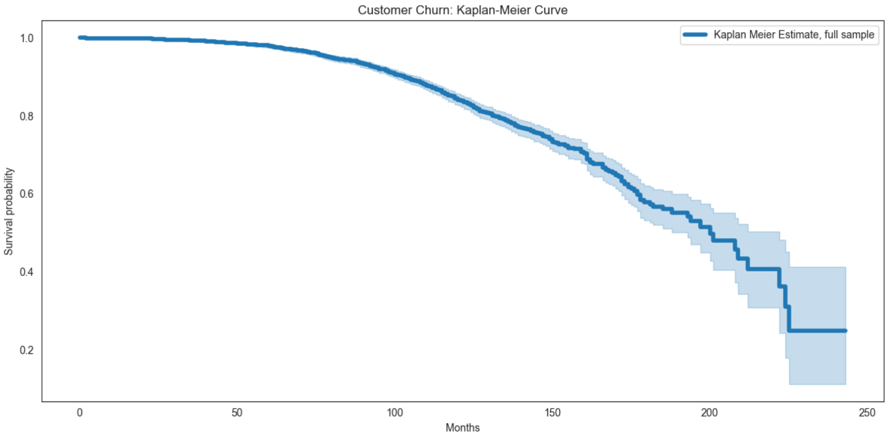


Figure 3: Customer Churn: Kaplan Meier Curve

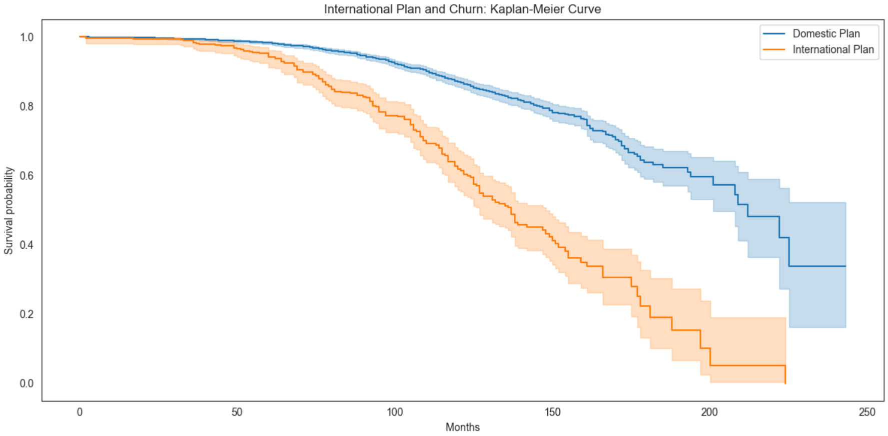


Figure 4: International Plan and Customer Churn: Kaplan Meier Curve

4 Variations of Classification Methods

In this section, I compare multiple deep learning techniques with a set of of traditional machine learning methods. In particular, 10-cross validation is used to get the mean of precision, recall, F_1 score and accuracy of the models as follows:

- Artificial Neural Network with one hidden layer associated with sigmoid, running with 1500 epoch with SGD optimizer (ANN-1LSig)
- Artificial Neural Network with one hidden layer associated with relu, running with 1500 epoch with SGD optimizer (ANN-1LRelu)
- Artificial Neural Network with two hidden layer associated with relu, running with 1500 epoch with SGD optimizer(ANN-2LRelu-1500)
- Artificial Neural Network with two hidden layer associated with relu, running with 3000 epoch with SGD optimizer (ANN-2LRelu-3000)
- Linear Regression(LR)

- K-Nearest Neighbour (KNN)
- Decision Tree (DT)
- Support Vector Machine (SVC)
- Random Forest (RF)
- Randomized Decision Tree (ET)
- Gradient Boosting ($GBoost$)
- Adam Boosting ($AdBoost$)

Table 1: Comparison of performance metrics on testing candidate algorithms

Methods	Precision	Recall	F_1 score	Accuracy
$ANN-1LSig$	0.850744	0.872941	0.842677	0.872941
$ANN-1LRelu$	0.918574	0.923137	0.916968	0.923137
$ANN-2LRelu-1500$	0.928660	0.931765	0.929397	0.931765
$ANN-2LRelu-3000$	0.918949	0.923137	0.919918	0.923137
LR	0.848686	0.872157	0.845092	0.872157
KNN	0.88032	0.88549	0.85605	0.88549
DT	0.941573	0.943529	0.940307	0.943529
SVC	0.923257	0.921569	0.909946	0.921569
RF	0.957765	0.956078	0.952706	0.956078
ET	0.910735	0.900392	0.875738	0.900392
$GBoost$	0.957092	0.956863	0.954196	0.956863
$AdBoost$	0.856489	0.87529	0.859283	0.875294

5 Models recommend

Based on the performance metrics of both ANN-based methods and traditional ML methods, apparently, the assembling learning algorithm, i.e., $GBoost$, achieves the consistently highest performance. All the performance metrics are conducted using 10-cross validation.

6 Key Findings and Insights

- Among all the ANN-based methods, $ANN-1LSig$ presents the worst performance in terms of the performance metrics. This is because the hidden layer with sigmoid activation function. Generally, it is caused by Gradient disappearance. Therefore, $ANN-1LRelu$ method has significantly improvement by replacing the sigmoid function in $ANN-1LSig$.
- Among all the ANN-based methods, $ANN-2LRelu-1500$ performs better than the other ANN-based methods. This performance is due to two reasons: $ANN-2LRelu-1500$ use 2

hidden layers, instead of one, increasing the model complexity to capture the difficult patter in our dataset; $ANN - 2LRelu - 1500$ use less epoch than $ANN - 2LRelu - 3000$, but it results in better performance. This is because the $ANN - 2LRelu - 3000$ leads to over-fitting issues when trained in 3000 epoch, see Fig 5 and 6 for details.

- Among all the machine learning methods, assembling learning based classifiers general outperform those techniques using one single classifier. In particulary, gradient boosting consistently and significantly achieves the highest performance, compared to other methods.

- Similar efforts are devoted to turning both ANN-based methods and classic ML methods. I believe ANN-based methods can reach the same performance as *GBoost*, but more time is required to tuning the parameters of ANNs. Therefore, *GBoost* presents a good practise considering both effectiveness and efficiency.

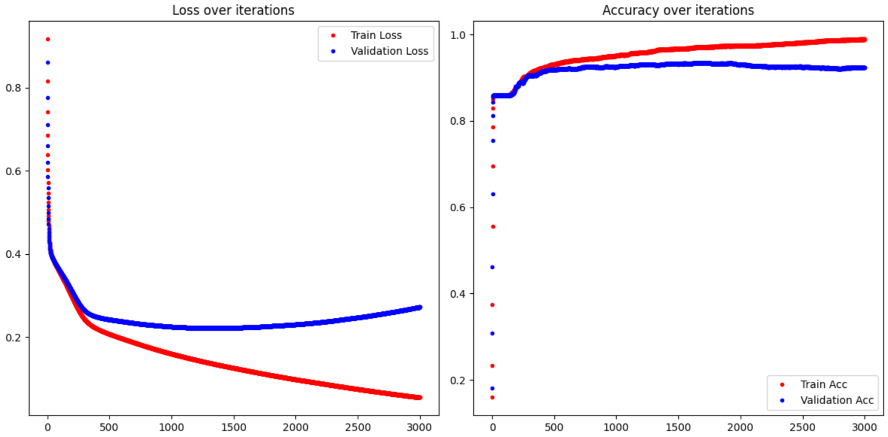


Figure 5: Over-fitting issues in $ANN - 2LRelu - 3000$ with 3000 epoch

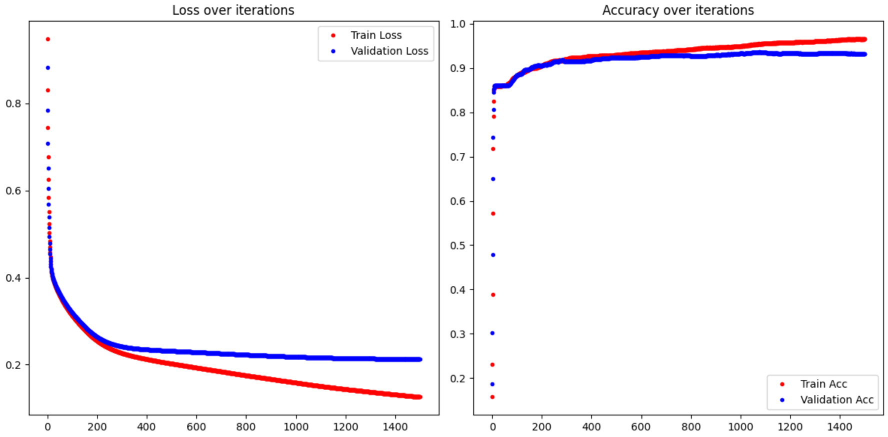


Figure 6: Avoid over-fitting by reduce epoch to 1500 in $ANN - 2LRelu - 1500$

7 Suggestions

- To increase quality measured by performance metric, one way to incorporate domain knowledge, i.e., to collect some other features related to customer life time value, such as recency, frequency, and monetary.
- The peformace of *GBoost* could be furthre improved by combining feature reduction techniques with *GBoost*. For example, I can use learned features via Autoencoder, and use that reduced features to train *GBoost*.