Predict Wireless Churn

Assessment

Farid Ghorbani
College of Engineering
Northeastern University
Toronto, ON
Ghorbani.f@northeastern.edu

Abstract

The aim of this project is to predict if a wireless account will churn or not. The business would like you to train and assess machine learning models using the provided sample data. As a secondary objective, the business would also like to understand which features are driving churn.

1 Datasets

The data essential for this project is distributed across three CSV files. Let's check out what kind of information we have in each set of data:

Dataset: wls_churn_master_target_t1.csv

In the dataset wls_churn_master_target_t1.csv, we have 9590 records, each representing a wireless account. There are 9 different features include 'Customer_ID' and our target feature 'churn', but unfortunately, the names of these features are not very clear—they're abbreviated and difficult to understand at first glance. This makes it challenging to figure out what each feature represents.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9590 entries, 0 to 9589
Data columns (total 9 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Customer_ID                     9590 non-null   int64
 1   mrc_current_month               9590 non-null   float64
 2   mvsf_br_amt_current_month       9581 non-null   float64
 3   mvsf_mrc_current_month          9590 non-null   float64
 4   num_subs_current_month          9590 non-null   int64
 5   num_voice_subs_current_month    9590 non-null   int64
 6   num_nonvoice_subs_current_month 9590 non-null   int64
 7   rev_current_month               9590 non-null   float64
 8   churn                           9590 non-null   float64
dtypes: float64(5), int64(4)
memory usage: 674.4 KB
```

Dataset: wls_customer_demographics_t1.csv

In the demographic dataset, there are a lot of different pieces of information about each customer. but similar to the master dataset, the feature names are unclear. If we try to train our machine learning algorithms with all of these features, our model will become complex. This means we pick out the most important features—the ones that really matter for predicting whether a customer will churn or not. By doing this, we can simplify our model and hopefully make better predictions.

```
Data columns (total 93 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   Customer_ID                                     9590 non-null   int64
 1   n_distinct_srvc_prov_state_cd_current_month     9590 non-null   int64
 2   n_distinct_sls_indust_typ_txt_current_month     9590 non-null   int64
 3   n_distinct_clli_exch_cd_current_month           9590 non-null   int64
 4   n_distinct_billg_prov_state_cd_current_month    9590 non-null   int64
 5   n_distinct_pymt_mthd_cd_current_month           9590 non-null   int64
 6   n_distinct_rgnl_cust_prov_state_cd_current_month 9590 non-null  int64
 7   n_distinct_cbu_cust_prov_state_cd_current_month 9590 non-null   int64
 8   srvc_prov_state_cd_ab_ind_current_month         9590 non-null   int64
 9   srvc_prov_state_cd_bc_ind_current_month         9590 non-null   int64
 10  srvc_prov_state_cd_mb_ind_current_month         9590 non-null   int64
 11  srvc_prov_state_cd_nb_ind_current_month         9590 non-null   int64
 12  srvc_prov_state_cd_nl_ind_current_month         9590 non-null   int64
 13  srvc_prov_state_cd_ns_ind_current_month         9590 non-null   int64
 14  srvc_prov_state_cd_nt_ind_current_month         9590 non-null   int64
 15  srvc_prov_state_cd_nu_ind_current_month         9590 non-null   int64
 16  srvc_prov_state_cd_on_ind_current_month         9590 non-null   int64
 17  srvc_prov_state_cd_pe_ind_current_month         9590 non-null   int64
 18  srvc_prov_state_cd_qc_ind_current_month         9590 non-null   int64
 19  srvc_prov_state_cd_sk_ind_current_month         9590 non-null   int64
...
 91  pymt_mthd_cd_tesoothr_ind_current_month         9590 non-null   int64
 92  pymt_mthd_cd_tpsoothr_ind_current_month         9590 non-null   int64
dtypes: int64(93)
```

Dataset: wls_billing.csv

In the dataset wls_billing.csv, we have 17 features related to billing, all of which are represented by integer values are 0 or 1. These values likely indicate various aspects of customer billing behavior.

```
Data columns (total 18 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   Customer_ID                                     9590 non-null   int64
 1   write_off_ind_n_ind_current_month               9590 non-null   int64
 2   write_off_ind_y_ind_current_month               9590 non-null   int64
 3   payment_method_cd_c_ind_current_month           9590 non-null   int64
 4   payment_method_cd_ca_ind_current_month          9590 non-null   int64
 5   payment_method_cd_cc_ind_current_month          9590 non-null   int64
 6   payment_method_cd_d_ind_current_month           9590 non-null   int64
 7   payment_method_cd_dd_ind_current_month          9590 non-null   int64
 8   payment_method_cd_r_ind_current_month           9590 non-null   int64
 9   kb_payment_method_cd_c_ind_current_month        9590 non-null   int64
 10  kb_payment_method_cd_d_ind_current_month        9590 non-null   int64
 11  kb_payment_method_cd_r_ind_current_month        9590 non-null   int64
 12  auto_payment_method_cd_ca_ind_current_month     9590 non-null   int64
 13  auto_payment_method_cd_cc_ind_current_month     9590 non-null   int64
 14  auto_payment_method_cd_dd_ind_current_month     9590 non-null   int64
 15  kb_auto_payment_method_cd_c_ind_current_month   9590 non-null   int64
 16  kb_auto_payment_method_cd_d_ind_current_month   9590 non-null   int64
 17  kb_auto_payment_method_cd_r_ind_current_month   9590 non-null   int64
dtypes: int64(18)
```

We combined all three datasets into one dataframe based on the Customer_ID, which serves as a unique identifier across all datasets. we now have a single dataset containing a total of 119 columns and 9590 records.

1.1 Data Exploration

1.1.1 Fill Missing Values

After checking all columns, I found that the column 'mvsf_br_amt_current_month' contains null values. To address this, I filled the null values with the most frequently value in the column.

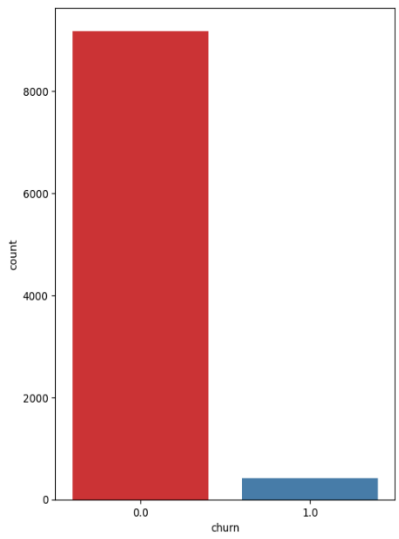
```
1  null_columns = churn_master_data.columns[churn_master_data.isnull().any()]
2  churn_master_data[null_columns].isnull().sum()
```

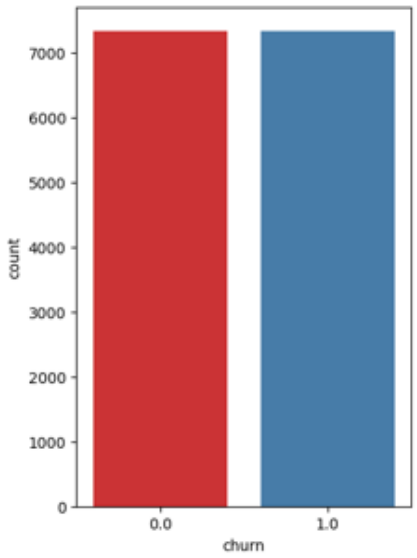
✓ 0.0s

```
mvsf_br_amt_current_month    9
dtype: int64
```

1.1.2 Is data balanced?

There are a lot more instances of Non-churn compared to churn in the dataset. This imbalance can pose a challenge when building predictive models, as the algorithms may become biased towards the majority class (Non-churn) and may not perform well in detecting the minority class (churn).



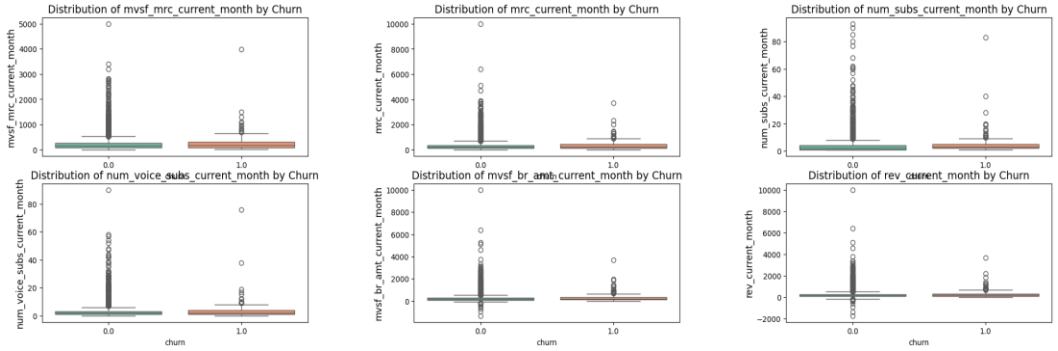


To address this issue, a technique called smoothing imbalanced oversampling was employed. This method generates additional samples for under-represented classes, thereby reducing bias and enhancing the robustness of the models.

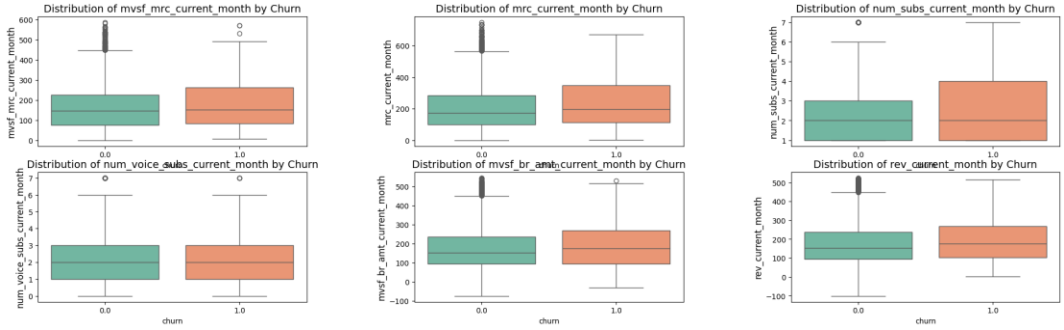
2 Models

Let's try out a few different machine learning models to see how they perform. I've picked three models: Random Forest, XGBoost, and KNN. After some tuning, all of them showed high accuracy scores around 96%. However, when we looked at the F1 score, which measures how well the model predicts both churn and non-churn cases, we found it to be 0.02. This suggests that the models are mainly predicting the majority class (non-churn) and not doing well on predicting churn. Also, there are many features.

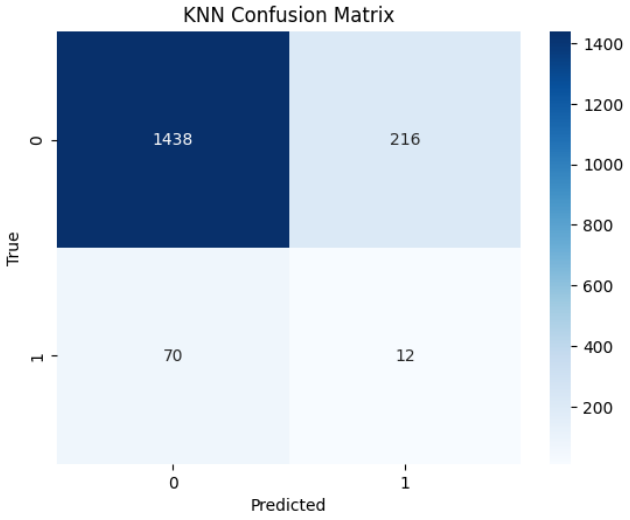
To address this, I tried to drop irrelevant features. I first conducted an analysis of feature correlations. Additionally, given the large number of features in the dataset, I attempted to enhance model performance by dropping irrelevant features. I began by listing the top 10 most positively and negatively correlated features with 'churn'.



Subsequently, I delved deeper into these features, analyzing their distributions using boxplots to identify potential extreme outliers. For some of these features, I removed extreme outliers specifically for churn transactions to mitigate their impact on model training.



Finally, I retained only these selected features and retrained the model to improve its predictive performance. As a result, the precision and recall score increased from 0.02 and 0.06 to 0.06 and 0.13 respectively. Notably, among the algorithms tested, the K-Nearest Neighbors (KNN) model showed slightly better performance compared to others.



3 Features Driving

I used Random Forest and XGBoost models to find the most important features and sorted them accordingly. In the XGBoost model, there are 6 features with 0 importance. However, the top 5 features are listed below.

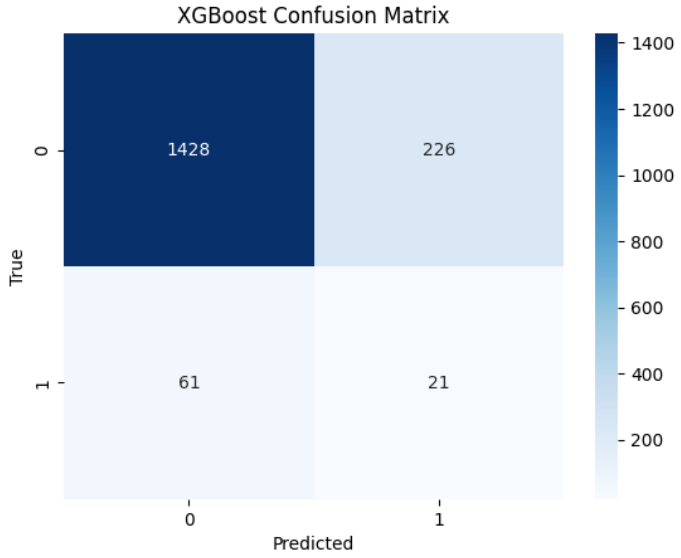
	Feature Name	Importance Score
106	payment_method_cd_r_ind_current_month	0.057821
0	mrc_current_month	0.050454
2	mvsf_mrc_current_month	0.050240
37	billg_prov_state_cd_qc_ind_current_month	0.049938
1	mvsf_br_amt_current_month	0.049847

Similarly, in the case of the Random Forest model, there are 4 features with 0 importance, and the top 5 features are also provided below.

	Feature Name	Importance Score
0	mrc_current_month	0.145794
1	mvsf_br_amt_current_month	0.134629
2	mvsf_mrc_current_month	0.129502
6	rev_current_month	0.126661
4	num_voice_subs_current_month	0.072724

4 Retrain Our Models

In further efforts to enhance model performance, I focused on removing irrelevant features more rigorously. For both the Random Forest and XGBoost models, we identified the top most important features and selected the union of the top 5 features from each. This refined approach yielded notable improvements, particularly in the XGBoost model. The recall score increased to approximately 0.26, with a precision score of 0.09. These results represent a significant improvement compared to the previous iterations of the model.



5 Conclusions

I suggested to my colleague that they create a data dictionary, which would clarify the meaning of each column in the dataset. This would make it easier for everyone involved to understand and work with the data effectively. Additionally, I recommended investigating the null rows in the dataframe to gain a better understanding of them.

I opted for XGBoost as the primary model for this dataset due to its built-in feature importance analysis feature, which facilitates the identification of the most influential features in predicting the target variable. Given the high number of features in the dataset, this capability proved to be particularly valuable in selecting the most relevant information for predicting churn.

While XGBoost excelled in handling the complexity of the dataset and provided insights into feature importance, the choice between algorithms also considered the nature of the dataset. KNN, known for its ability to perform well in datasets with imbalanced classes, such as ours where there are significantly more non-churn instances than churn ones, was also considered. KNN's methodology of looking at nearby data points to make predictions allows it to effectively identify churn instances even in the presence of imbalanced classes. However, after careful consideration, XGBoost was chosen as the preferred model for its robust feature selection capabilities and overall performance in this context.

Finally, based on the feature importance analysis conducted using Random Forest and XGBoost, we identified four features that stand out as particularly important:
'payment_method_cd_r_ind_current_month', 'mrc_current_month', 'mvsf_mrc_current_month', and 'mvsf_br_amt_current_month'