Machine Learning Problem Set Report

Rishabh Kaushick
NU ID: 002808996
College of Engineering
Northeastern University
Toronto, ON
kaushick.r@northeastern.edu

Assignment 4

Datasets

In this assignment, we are working on the datasets provided in the problem set. This includes the data of wireless accounts in an organization. We would be focusing on the wireless account terminations – which is otherwise known as churning. The data for this project is found in 3 csv files as mentioned in the below table.

Dataset	Name	Dataset Characteristics	Attribute Characteristics	Number of Instances	Number of Attributes
1	Master Data with Target – Churn	Multivariate	Real	9590	9
2	Demographic Data	Multivariate	Real	9590	93
3	Billing Data	Multivariate	Real	9590	18

Table 1: Dataset Details

Exploratory Data Analysis

Before analyzing the data, let us look at the columns in each dataset.

```
master_data.info()
[22] ✓ 0.0s
...  <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
```

Figure 1: Master Dataset with the Target variable.

```
demographic_data.info()
```

```
[6]  ✓ 0.0s
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9590 entries, 0 to 9589
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)
```

Figure 2: The demographic data containing 93 columns.

```
billing_data.info()
```

```
[25]  ✓ 0.0s
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9590 entries, 0 to 9589
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)
memory usage: 1.3 MB
```

Figure 3: Billing csv dataset.

Figures 1 to 3 show the details of each of the csv datasets. In all the three datasets, the names of the columns are abbreviated based on the domain, and it is difficult to interpret their meaning.

Therefore firstly, I would suggest to my colleague that it would be better to create a data dictionary at the time of modelling. This way any employee like a data scientist or data engineer will be able to figure out the names of each column and understand a brief description about the same. This will be very helpful for a better analysis.

From figure 2, we can see that in the demographic csv dataset there are a total of 93 columns. If we train the machine learning algorithms with so many features, the model will become extremely complex and may not accurately make a good prediction. Therefore here, we need to perform dimensionality reduction here.

From Figure 2 and Figure 3, we can see that many of the column names are similar with just a slight change. For example, from figure 3 notice the following column:

1. write off ind:
   - write_off_ind_**n_ind**_current_month
   - write_off_ind_**y_ind**_current_month
2. payment_method_cd
   - payment_method_cd_**c_ind**_current_month
   - payment_method_cd_**ca_ind**_current_month
   - payment_method_cd_**cc_ind**_current_month
   - payment_method_cd_**d_ind**_current_month
   - payment_method_cd_**dd_ind**_current_month
   - payment_method_cd_**r_ind**_current_month
3. kb_payment_method_cd
   - kb_payment_method_cd_**c_ind**_current_month
   - kb_payment_method_cd_**d_ind**_current_month
   - kb_payment_method_cd_**r_ind**_current_month

We can see that the text in bold is the one which is changing. We can take an educated guess and say that **ind** stands for indicator. For example perhaps the 'payment_method_cd_**c_ind**_current_month' refer to those which have a 'c' indicator, whereas the 'payment_method_cd_**ca_ind**_current_month' refer to those which have a 'ca' indicator. Since we can see that the values of all these columns are either 0 or 1, it looks like it is one hot encoded version of a previous column.

Figure 1 above and Figure 4 below show that the column 'mvsf_br_amt_current_month' has 9 records with null data.

```
# this shows the records which are having a null value in one of the columns.
master_data[master_data.isna().any(axis=1)]
```

[37] ✓ 0.0s

	Customer_ID	mrc_current_month	mvsf_br_amt_current_month	mvsf_mrc_current_month	num_subs_current_month
549	549	35.0	NaN	35.0	1
1115	1115	185.0	NaN	185.0	3
1566	1566	35.0	NaN	35.0	1
1705	1705	35.0	NaN	35.0	1
3596	3596	35.0	NaN	35.0	1
4070	4070	35.0	NaN	35.0	1
4163	4163	75.0	NaN	75.0	1
5668	5668	35.0	NaN	35.0	1
6714	6714	35.0	NaN	35.0	1

Figure 4: Column containing null records.

Additionally I would like to ask my colleague more details about the rows which have null records. While modelling the data it would be better to understand the reason behind the null columns. This will also help us understand whether we must drop these rows, or what would the best way to fill those values.

Preparing The Data For The Models

Combining The Different CSVs into 1 Data Frame

We cannot train the model on three different data-frames, therefore first let us merge them into one data-frame object using the Customer ID column as the primary key:

```
# Using the merge function to combine the master data and demographic data on primary key - Customer ID column using an outer join.
master_data = master_data.merge(demographic_data, on=['Customer_ID'], how='outer')
```

[39] ✓ 0.0s Python

```
# Similarly merging master data and billing data
master_data = master_data.merge(billing_data, on=['Customer_ID'], how='outer')
```

[40] ✓ 0.0s Python

Figure 5: Merging data frames into one.

As a result of the merging, now the main data frame object contains a total of 119 columns:

```
master_data.info()
```
[41] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9590 entries, 0 to 9589
Columns: 118 entries, Customer_ID to kb_auto_payment_method_cd_r_ind_current_month
dtypes: float64(5), int64(113)
memory usage: 8.6 MB
```

Figure 6: Master data frame object.

Handling Null Values

The column '*mvsf_br_amt_current_month'* contains null values. Let us understand the type of data present in this column:

```
# first understanding the column containing null values
master_data['mvsf_br_amt_current_month'].describe()
```
[44] ✓ 0.0s

```
count     9581.000000
mean       242.222068
std        305.696945
min      -1317.090000
25%        100.000000
50%        168.000000
75%        276.260000
max      10000.000000
Name: mvsf_br_amt_current_month, dtype: float64
```

Figure 7

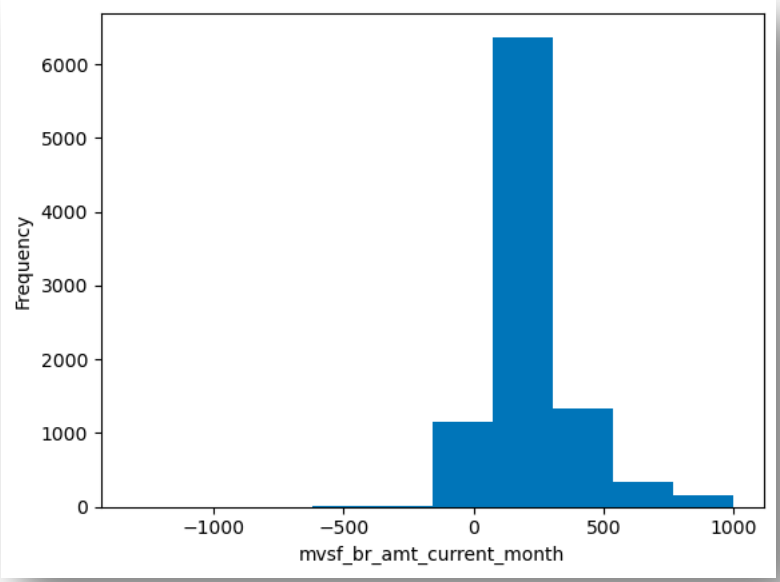


Figure 8

The data looks like it is mostly positive with a mean of 242. Let us fill the null values with the mean.

```
# filling the null rows with the mean value
master_data['mvsf_br_amt_current_month'] = master_data['mvsf_br_amt_current_month'].fillna(master_data['mvsf_br_amt_current_month'].mean())
```
[35] ✓ 0.0s

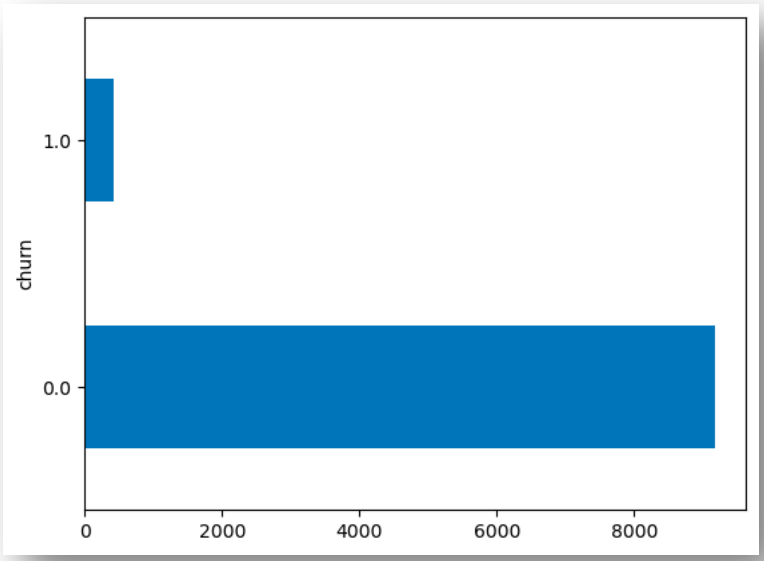
```
master_data['mvsf_br_amt_current_month'].isna().sum()
```
[37] ✓ 0.0s

```
0
```

Figure 9: Filling null values with the mean.

Exploring The Target Variable – Churn

Let us visualize the type of values present in the churn target column.




```
master_data['churn'].value_counts()
```

[38] ✓ 0.0s

```
churn
0.0    9174
1.0     416
Name: count, dtype: int64
```

Figure 10: Churn data distribution.

We can see that there are more than 9000 records for which the churn is 0, however, there are only a little over 400 records in which the churn is 1. Therefore, in this situation the model, will learn a lot of data when churn is equal to 0, however, it will not have enough data to learn when churn is 1.

I would suggest to my colleague that while data collection and data modelling, we should try to collect enough data points for all categories in the target column.

Dimensionality Reduction Using PCA

In this assignment, I have decided to use Principal Component Analysis (PCA) technique to reduce the dimensionality of the features. As compared to ICA, PCA does a better job to find the un-correlated items, thereby helping with feature reduction. ICA is better to filter out different independent components from a mixture of data sources, but that is not what we need to do for this situation.

In the data frame, we have 119 features, i.e. 119 dimensions. We must now decide what should be the dimensionality of the resulting data frame. We do not want to keep too less dimensions so that the features are lost, and at the same time we do not want to keep too many features as that will make the model more complex. Let us try to use 2, 5 and 10 components for this assignment, and when we train the supervised learning model later, we will be able to find out which works the best.

The figure below shows the plots of points between the first and second component for PCA implemented with number of components = 2. The blue circles denote those points which have churn = 0, and the orange triangle denote those which have churn = 1.

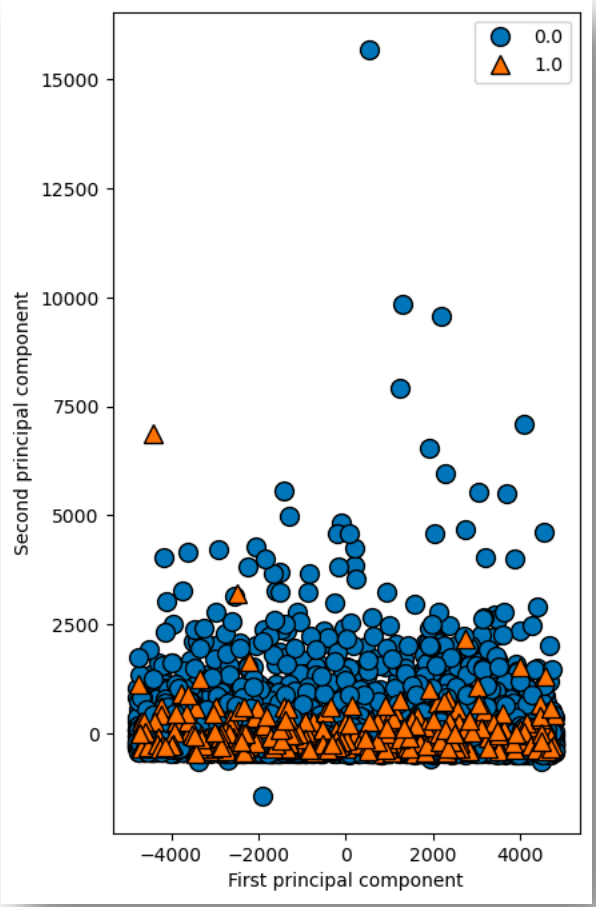


Figure 11: Plot of PCA with n components = 2.

The above figure shows that most of the data is very close to each other on the bottom of the graph. It does not seem to create any meaningful clusters. It is most likely possible that we need a higher dimension for there to be a clear separation of clusters.

Machine Learning MVP

Decision to Choose Models

Let us use a few different machine learning models for benchmarking. And we can further build on top of those. I have decided to choose the following models:

1. Random Forest
2. Artificial Neural Networks

There are a few reasons behind choosing the Random Forest model:

1. We can visualize the random forest's decision trees prepared by the system and understand the reason behind why the machine is making a specific prediction.
2. Random Forests computes multiple decision trees and finds the optimum ones by using average or majority ranking, thereby usually making a better model as compared to one decision trees.
3. Random Forests can help us understand the feature importance map. This will be very useful to understand which features are contributing to the churn.

Additionally, we will use Artificial Neural Networks so that we can compare between a black box approach together with the Random Forest Model.

Random Forest Without PCA

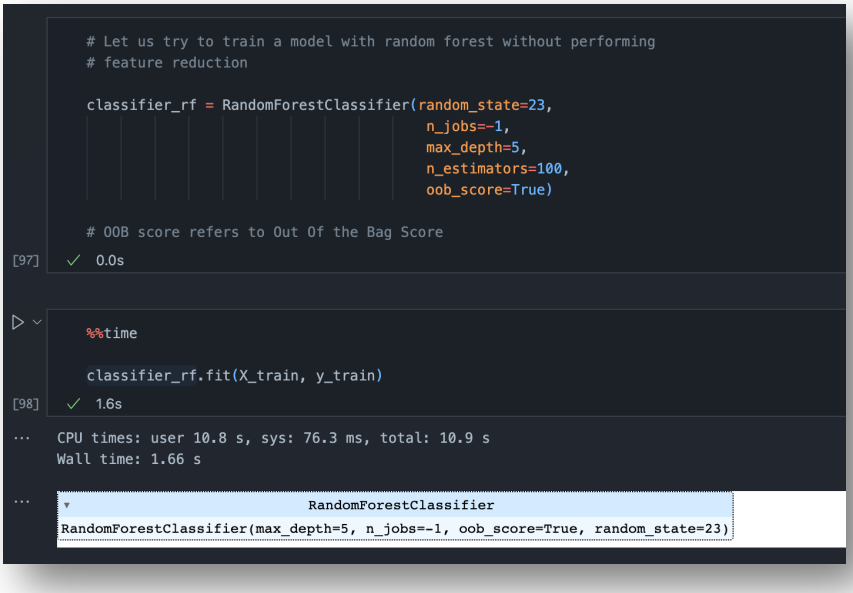


Figure 12: Random Forest Classifier without PCA.

Surprisingly, we can see that the training and the testing accuracy of this model is around 95%.

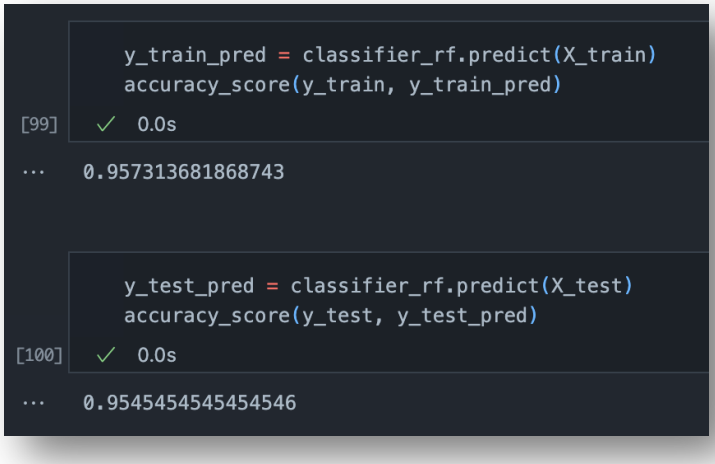


Figure 13: Train & Test Accuracy of the Random Forest model.

Random Forest Model + PCA (n=2 & n=5)

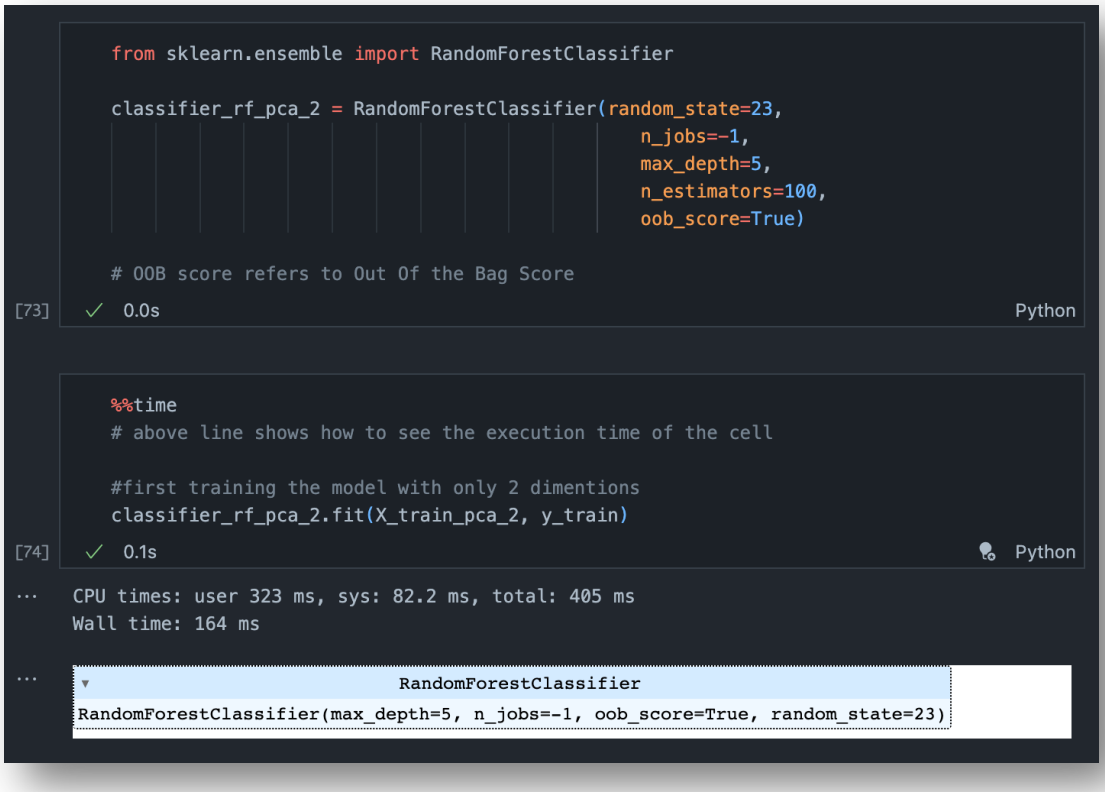


Figure 14: Random Forest Classifier on PCA (n=2).

Additionally, I trained a Random Forest model on the low dimension data obtained after the PCA operation where n = 2 and n = 5. As seen in the figures below, the training and testing accuracy is around than 95% for both these models.

Figure 15: Train & Test Accuracy of the Random Forest + PCA (n=2) model.

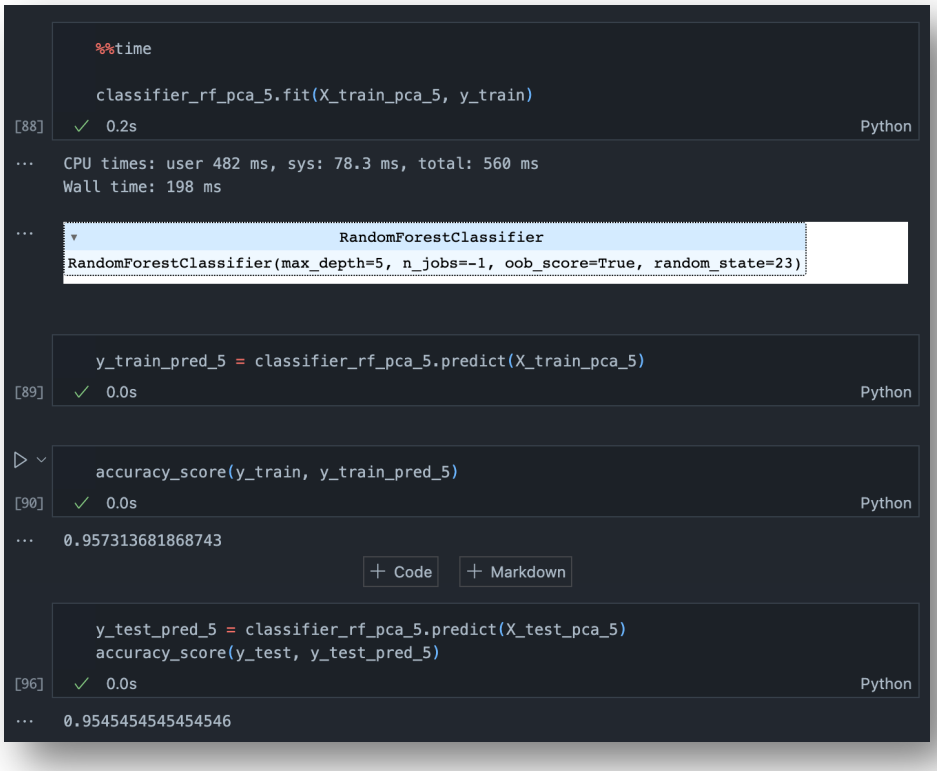


Figure 16: Train & Test Accuracy of the Random Forest + PCA (n=5) model.

Features Driving the Churn

Random Forest has a way to find out which are the features which are important. Let us try to find the feature importance of all the above three random forest models.

```
# Visualizing the above data in a better way:
important_features_rf = pd.DataFrame({
    "Feature Name": X_train.columns,
    "Importance Score": classifier_rf.feature_importances_
})

important_features_rf.sort_values(by="Importance Score", ascending=False)
```

[110] ✓ 0.0s

	Feature Name	Importance Score
3	mvsf_mrc_current_month	0.133893
7	rev_current_month	0.123576
1	mrc_current_month	0.115669
2	mvsf_br_amt_current_month	0.105420
0	Customer_ID	0.104347
...	...	...
67	pymt_mthd_cd_cc_ind_current_month	0.000000
66	cbu_cust_prov_state_cd_yt_ind_current_month	0.000000
65	cbu_cust_prov_state_cd_sk_ind_current_month	0.000000
63	cbu_cust_prov_state_cd_pe_ind_current_month	0.000000
58	cbu_cust_prov_state_cd_nl_ind_current_month	0.000000

117 rows × 2 columns

Figure 17: Feature Importance of the Random Forest model.

This is a really good way for us to find out which out of the 119 features have a significant impact on the random forest model. From the above figure we can see the columns which have the highest importance score. It can be noticed however, that the Customer ID should not be a part of the model since the value of this column does not have any significance.

Furthermore, the importance score can be very useful to remove columns which have 0 significance to the model. We can see all the bottom 5 rows have 0 significance to the model.

We have also computed the significance score for the random forest models which had been trained on the data from the PCA algorithm as shown below.

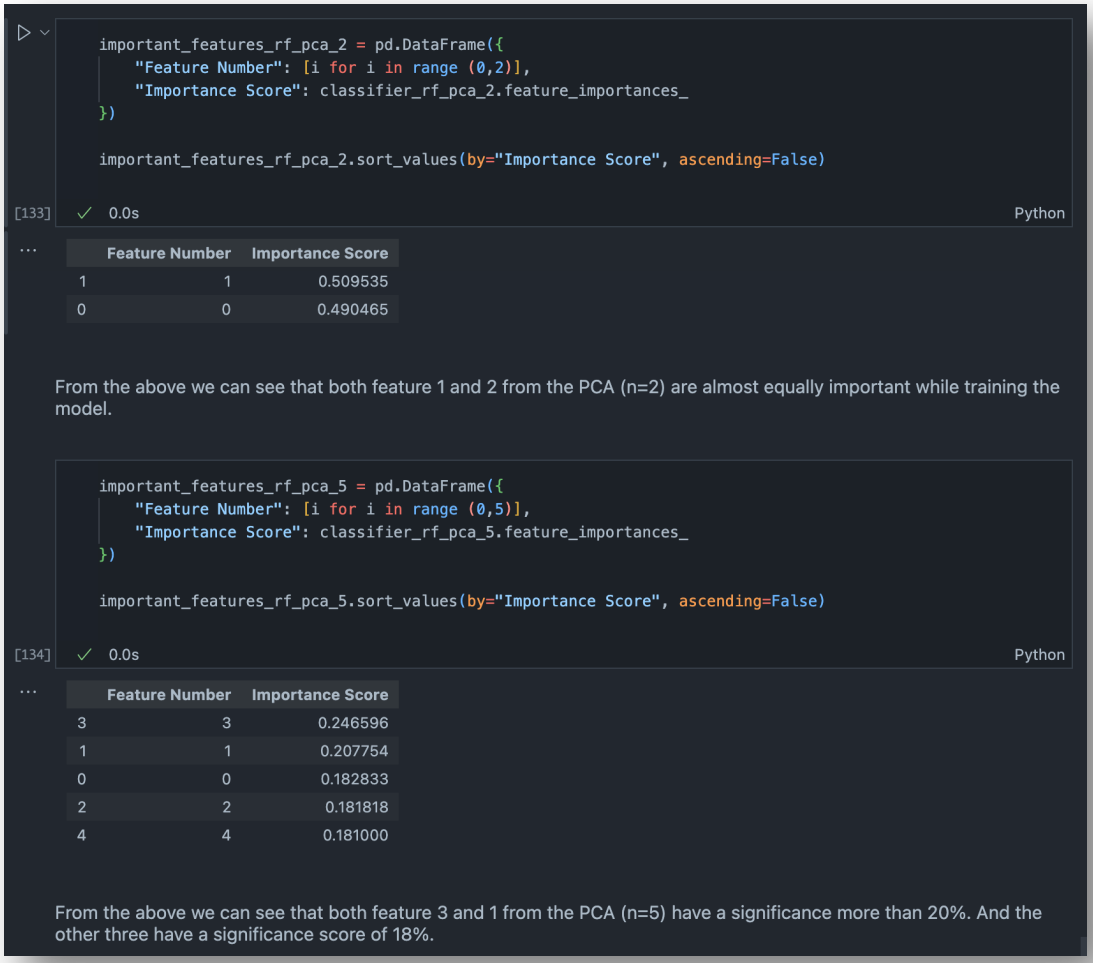


Figure 18: Feature Importance of the Random Forest + PCA models.

Conclusion

For the problem set, we have analyzed the data provided by our colleague and provided some feedback regarding the data modelling and collection. We suggested them to:

1. Create a data dictionary.
2. Enquire about the details of the null rows in the data-frame so that we can best handle the null rows.
3. To collect enough data points for all categories in the target column during data collection and data modelling.

Furthermore, I applied PCA for dimensionality reduction from 119 features down to 2 features & again down to 5 features. I created 3 Random Forest models trained on:

- All Data (119 features)
- Data after PCA (n=2)
- Data after PCA (n=5)

We could see that the random forest model had a very similar accuracy of 95% in both training and testing for all the three models. The best model was the Random Forest + PCA (n=2) with training accuracy of 95.745% and testing accuracy of 95.455%.

In this problem set, we also found which of the features played a major role in determining the churn using Random Forest's feature importance map.

References

1. *What is a data dictionary?*. What Is a Data Dictionary? | UC Merced Library. (n.d.). https://library.ucmerced.edu/data-dictionaries
2. Eddie_4072. (2023, July 16). *How to handle missing data in python? [explained in 5 easy steps]*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/dealing-with-missing-values-in-python-a-complete-guide/
3. Mehta, S. (2022, May 14). *Independent Component Analysis vs Principal Component Analysis*. Analytics India Magazine. https://analyticsindiamag.com/independent-component-analysis-vs-principal-component-analysis/
4. R, S. E. (2023, October 26). *Understand random forest algorithms with examples (updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
5. Kreiger, J. (2021, September 10). *Evaluating a random forest model*. Medium. https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56