Technical Report

This study was conducted to predict if a client will stop running advertising campaign(s) on certain social networks' platforms.

The Ads dataset contains 10 features (variables) including:

1. **CPL_wrt_BC** which is the change in cost per lead with respect to business category,
2. **client_state** which indicates the client's location,
3. **duration** which specifies how long the client has been running advertising campaigns in months,
4. **num_prods** that specifies how many distinct advertising products the client has bought,
5. **calls** that is the number of calls received,
6. **CPL_wrt_self** which is change in client's cost per lead in the past three months,
7. churn that is our target column (0=retention | 1=churn),
8. avg_budget which indicates the average monthly budget spent on advertising campaigns,
9. **BC** that is the client's business category, and
10. **Clicks** which shows the number of clicks received for a particular advertisement.

By performing preliminary Exploratory Data Analysis (EDA), this can be inferred that there are 10,000 observations, and 10 features in the dataset. In addition, there are two string categorical features (BC and Client states) in the dataset. In order to create a ML model, we should not have text in our data. Therefore, before creating a model, we need to make this data ready in order to convert these categorical string data into model-understandable numerical data by replacing the existing string data with the new encoded data (Label Encoding). For instance, since there are 50 states in the US, we can assign one numeric value to each unique state (in this case is between 0 and 49). However, the final model may develop a (wrong) correlation such as the state number increases the clicks increases, but this clearly may not be the case in some other data. Another possibility is to convert the categorical string data to binary. There are 20 distinct customer categories and 50 states for the categorical features, respectively. If we convert the categorical features to binary (OneHotEncode), we may end up adding ~70 features for these two columns only which is not rational. In addition, I would suggest to add population of each state in the dataset. if we could add the population of each state to the dataset, our model would be more accurate since the population of each state is different and can have more correlation with some other features.

By performing another preliminary exploratory data analysis, this can be inferred that there are about ~1000 missing values in one of the features (CPL_wrt_self), which is related to the change in client's cost per lead in the past three months. Since the percentage of the missing data is ~10% of the whole dataset, in this model all the observations are kept and an Imputer is used to fill the missing data with the "most-frequent" strategy.

In addition, by calculating IQR and Z-Score for all the features, I realized that there are outliers in some features of data. The data points which falls more than 1.5 times the interquartile range above the third quartile or below the first quartile. For instance, there are 60, 21, 65 data points are detected as outlier for CPL_wrt_self, CPL_wrt_BC, and avg_budget, respectively.

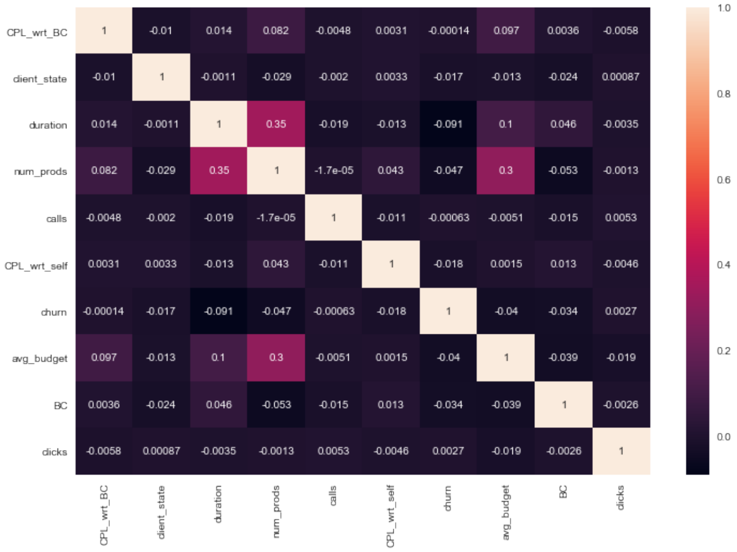
In addition, the scale of data in the dataset is not normalized and the data is not distributed normally among the observations. Therefore, in the model pipeline, I have normalized (which is also known as centering and scaling) each feature by computing the relevant statistics on the samples in the training set to be on a similar scale (all features are centered around zero and having variance one).

In this dataset, Chrun is considered as the target variable, and the other 9 features are independent variables. The value of Churn is not normally distributed and there is an imbalanced data for the target variable. The model uses Synthetic Minority Over-sampling technique to achieve an equal number of sample with the majority or minority class. I have used five different classifiers for the “Ads” dataset and since the performance of different ML algorithms depends on the size and structure of data, I came up with different results. However, Random Forest (RF) classifier tends to have better accuracy (88%) among all of them. Therefore, the finalized model (model_ML function) is created based on using RF classifier. The following table shows the results of different classifiers on the dataset (more details are added in the appendix).

Classifier Name	Accuracy	TP	TN	FN	FP
KNN	0.767	1491	43	364	102
Decision Tree	0.744	1324	164	243	269
GNB	0.4895	703	276	131	890
Random forest(inc. MSOTE)	0.88	1446	1376	243	135
SVM	0.7965	1593	0	407	0

By using the designated model (RF), 2,822 observations have been selected correctly on the test data (out of 3200 observations). In RF classifier, I used GridSearchCV to find out the best hyper parameters including the number of trees. However, the more trees you have, the better and reliable) estimates you get from out-of-bag predictions.

I have also tried to find out the correlation between different features in the dataset. The following figure illustrates the correlation among the features in the dataset. As can be inferred from the following figure, there is a 30% correlations between avg_budget and num_prods. num_prods and duration are highly correlated (35%). In addition, churn and duration have negative correlation.



As a summary, a machine learning model has been created based on RandomForest classifier on Ads dataset. The accuracy of the model is 88%. A pipeline is created to impute (the missing data), standardize the observations, balancing data based on the target variable, train, and test the classifier.

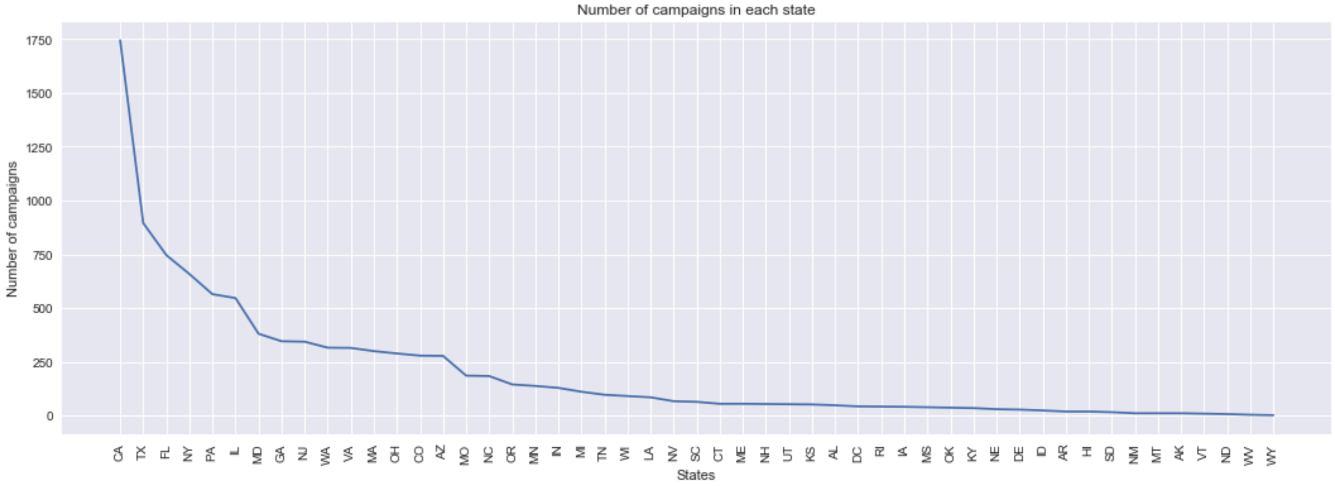
Non-Technical Report

This study was designed to predict if a client will stop running advertising campaign(s) on certain social networks' platforms. The Ads dataset contains 10 variables including:

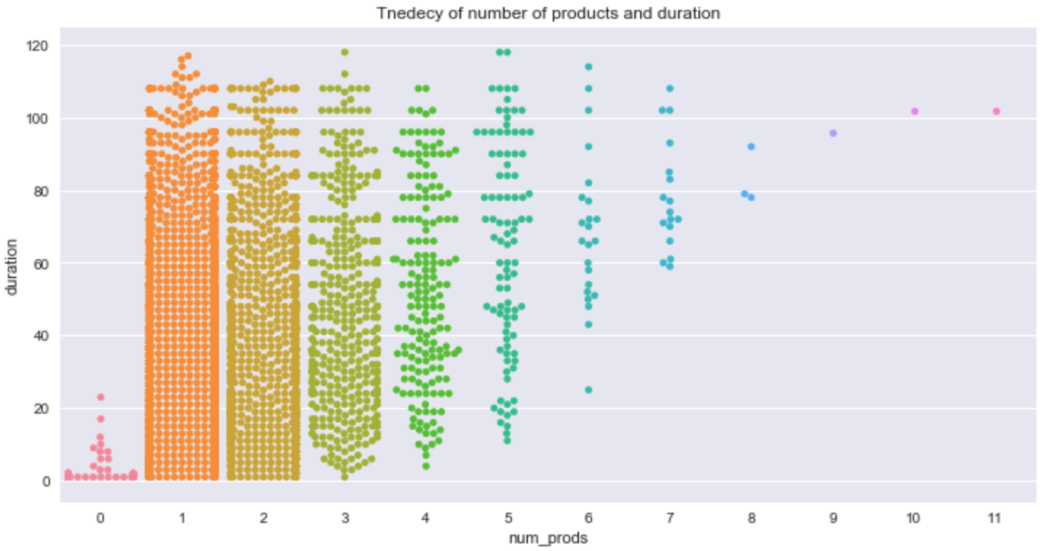
11. The change in cost per lead with respect to business category,
12. The client's location,
13. How long the client has been running advertising campaigns in months,
14. Number of distinct advertising products the client has bought,
15. Number of calls received,
16. A value for change in client's cost per lead in the past three months,
17. A value, churn that is our target column
18. The average monthly budget spent on advertising campaigns,
19. Client's business category, and
20. The number of clicks received for a particular advertisement.

I have used several exploratory data analysis (EDA) methods to detect missing values, and outliers in order to clean the dataset. After performing the preliminary EDA, I have created a machine learning (ML) model in order to predict the if a client will stop running advertising campaign or not. This model is crated based on 10000 observations from 50 states in 31 business categories in the united states. The model can then be used for predictions of future data.

In the dataset, there are 100,000 observations. 80% (80,000) of clients have not stopped their campaigns yet, and 20% (20,000) of clients have already stopped their campaigns. The model could be more accurate if we can add the similar number (balanced) of observations for those groups. A larger dataset might expose a different and perhaps more balanced perspective on the classes. In addition, the data is not normally distributed among different states. For instance, in California (CA), there are 1,745 observations, meanwhile in Wyoming (WY) there is only 1 observation as can be seen from the following figure.



The clients in the states which have more number of campaigns, have more products and those who have more products tend to continue the campaign more than those clients who have less number of products as can be seen from the following figure.



Based on the findings in this study, the following recommendations are made:

1. An ad campaign duration is highly dependent on number of products. The company should encourage the clients to invest and add more products, which makes them to have more engagements according to duration and number of clicks.

2. About 30% of data related to the business categories of "Home & Home Improvement". It seems that this category has the highest demand and more profitable. On the other hand, "Government & Politics" category has the lowest interest (0.0003%) among the different categories. Therefore, it worth investing on the Home and Home Improvement businesses which are more profitable.

3. Since each state has different population, it would be better to consider population of each state to have more accurate model.

4. The type of platform is not mentioned in the dataset (e.g. Facebook, twitter, Instagram, etc.). Advertising cannot focus on one type of platform and the model would be more accurate, if we could add the type of platform in the dataset to find out which platform is more popular among the customers or/and for specific business category.

5. According to business categories, some businesses may have a more immediate effect, therefore they may stop running the campaign as soon as they see the results, and some businesses may take time to see the effects. For example, online stores could have an immediately trackable success in advertising. because tracking is easy for an online store we will be able to see if advertising is working within the first few days. The type of business is not also mentioned in the dataset (Online shop or physical store).

As a summary, the created model is able to predict if the campaign is going to stop or not, based on the existing data in the dataset. The model is created based on a machine learning classifier according the dataset which includes 10000 observations and 10 features. The accuracy of the model is ~88%, which is the number of correct predictions made divided by the total number of predictions, multiplied by 100 to turn it into a percentage.

Appendix – Results of different classifiers on the Ads dataset (before over-sampling)

```
**************
 Cassifier :   KNN
 ************
ML model accuracy score: 0.767
[[1491  102]
 [ 364   43]]
              precision    recall  f1-score   support

           0       0.80      0.94      0.86      1593
           1       0.30      0.11      0.16       407

   micro avg       0.77      0.77      0.77      2000
   macro avg       0.55      0.52      0.51      2000
weighted avg       0.70      0.77      0.72      2000

Model Performance
Predictions: ['retention', 'retention', 'retention', 'retention', 'retention']

 **************
 Cassifier :   DTree
 ************
ML model accuracy score: 0.744
[[1324  269]
 [ 243  164]]
              precision    recall  f1-score   support

           0       0.84      0.83      0.84      1593
           1       0.38      0.40      0.39       407

   micro avg       0.74      0.74      0.74      2000
   macro avg       0.61      0.62      0.61      2000
weighted avg       0.75      0.74      0.75      2000

Model Performance
Predictions: ['retention', 'churn', 'retention', 'churn', 'retention']

 **************
 Cassifier :   GNB
 ************
ML model accuracy score: 0.4895
[[703 890]
 [131 276]]
              precision    recall  f1-score   support

           0       0.84      0.44      0.58      1593
           1       0.24      0.68      0.35       407

   micro avg       0.49      0.49      0.49      2000
   macro avg       0.54      0.56      0.47      2000
weighted avg       0.72      0.49      0.53      2000

Model Performance
Predictions: ['churn', 'churn', 'churn', 'churn', 'retention']
```

```
**************
 Cassifier :   DTree
 ************
ML model accuracy score: 0.744
[[1324  269]
 [ 243  164]]
              precision    recall  f1-score   support

           0       0.84      0.83      0.84      1593
           1       0.38      0.40      0.39       407

   micro avg       0.74      0.74      0.74      2000
   macro avg       0.61      0.62      0.61      2000
weighted avg       0.75      0.74      0.75      2000

Model Performance
Predictions: ['retention', 'churn', 'retention', 'churn', 'retention']


 **************
 Cassifier :   Rforest
 ************
ML model accuracy score: 0.8325
[[1558   35]
 [ 300  107]]
              precision    recall  f1-score   support

           0       0.84      0.98      0.90      1593
           1       0.75      0.26      0.39       407

   micro avg       0.83      0.83      0.83      2000
   macro avg       0.80      0.62      0.65      2000
weighted avg       0.82      0.83      0.80      2000

Model Performance
Predictions: ['retention', 'retention', 'retention', 'churn', 'retention']



 **************
 Cassifier :   SVM
 ************
ML model accuracy score: 0.7965
[[1593    0]
 [ 407    0]]
              precision    recall  f1-score   support

           0       0.80      1.00      0.89      1593
           1       0.00      0.00      0.00       407

   micro avg       0.80      0.80      0.80      2000
   macro avg       0.40      0.50      0.44      2000
weighted avg       0.63      0.80      0.71      2000
```