**Assignment Documentation**

**(Churn Prediction)**

1. **Solution approach and rationale** :

This is a problem statement that comes under the class of churn prediction problems. The basic approach of this problem is find the best of all the features available to us that are the main cause of the subscribers churning.

* Python and python based available libraries has been used as the language to build the whole approach.
* I’ve used various ways so as to analyze the data at hand, in order to get to our motive.
* I used multiple algorithms for extracting out the key features from all that were given.
* I also used correlation for finding collinear features in order separate or choose the relevant one.
* The results of all the algorithms applied can then be observed to find out the favorable and more reliable outcomes.

1. **Details:**
   1. **Data Preprocessing:**

* Firstly, we use .*info* and *.describe* methods from pandas to get a precise description of the data at hand.
* Using *.info* we get the number of non-null values in each of the rows and the data type it is of.
* Using .describe we get various parameters about the different columns with *ordinal data* in the dataset, like their count, the mean values, the standard deviation, the distribution of the data percentage-wise, etc. We can infer a good understanding of the distribution of the values from various features that are *ordinal* from this. The information of the numerical categorical values cannot be inferred from this.
* Then we checked the unique values in a particular column wherever required to get an idea of the various values and the variety of values in them.
* We also found the amount of null values present in some of the relevant features. We found that, END\_DATE has a lot of null values and hence it couldn’t be used for the classification.
* Also, the column ‘AGENT\_CODE’ which apparently relates to the various agents that lead a particular member. The number of unique values was 4317 against the total number of columns 10362. Hence, we can’t use this one too. It is a nominal data and as it is too large, we can’t even use it after encoding it. So, we drop this column too.
* The column ‘*MEMBER\_ANNUAL\_INCOME*’ had a few null values, we replaced them with the mean value of the column.
* The null values found in the columns, *MEMBER\_MARITAL\_STATUS, MEMBERSHIP\_STATUS, MEMBERSHIP\_PACKAGE, PAYMENT\_MODE, MEMBER\_GENDER* were replaced with value ‘*Other’*. And the null values in MEMBER\_*OCCUPATION*\_CD were replaced with numeric value of *7*.
* From the column ‘START\_DATE’, we extracted the month, as it might be a feature.
* We applied *hotcode encoding* on the categorical columns using *pandas.get\_dummies* and we get a new set of columns to work on.
  1. **The created model:**

Here we will be looking at the list of all the packages and algorithms that were used in this approach.

* Significant Packages:
  + Numpy
  + Pandas
  + Xgboost
  + Seaborn, matplotlib
  + Sklearn (pipeline, model\_selection, linear\_model, ensemble, feature\_selection, metrics)
* Algorithms:
  + LogisticRegression
  + ExtraTreesClassifier
  + RandomForestClassifier
  + SelectKBest
  + SelectFromModel
  + xgboost
* *Numpy* helps with handling the numerical needs, like handling none values, or implementing some other numerical functions.
* *Pandas* is one of the very best libraries that helps a lot with excel/csv sheets and hence with the “dataframes”.
* *Seaborn* and *Matplotlib* were used for plotting the required graphs and charts.
* *LogisticRegression, ExtraTreesClassifier, RandomForestClassifier, SelectKBest, SelectFromModel, xgboost* are the algorithms that I used for fitting the model and getting the best features that the model used for getting the best fit and hence classification.
* *Logistic Regression* also gave us the coefficients/parameters of the features that it learned. Referring to [Coefficients for Binary Logistic Regression](https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/how-to/binary-logistic-regression/interpret-the-results/all-statistics-and-graphs/coefficients/) we can interpret them to get an inference for us.
* Then we used *RandomForestClassifier, ExtraTreesClassifier, SelectKBest, xgboost*, to get the important features.

From these all information, that is, the coefficients and the crucial features from all different algorithms we can point out the specific reasons that classifies a user whether it will churn or not.

If you go through the code, you could see that a few features were among the best features from every algorithm, like, *MEMBER\_AGE\_AT\_ISSUE, MEMBER\_ANNUAL\_INCOME, ANNUAL\_FEES,* etc were among the crucial features effecting which class the user belongs to, if it will churn or not.

***xgb\_model* from the *xgBoost* also gives us the churn probability of each of the users based on the features. Using this we can even go in details and see what type of users are more probably going to churn. And we take appropriate measures for each of them and get their views about the membership.**

* 1. **Training and test results**

We will go through the various models used.

* The first model we used is *XGBClassifier*.
  + It gave us accuracy on training set of 75% and 70% on the test set.
  + It also gave us a plot of the important features.
  + With that, we also get the churn probabilities of each of the users which can be used for manual enquiry at later stage.
* The second model used was RandomForestClassifier.
  + We used it with *GridSearchCV* to get the best model using optimized features with hyperparameter tuning.
  + The accuracy with the best model was of 67%, similar as quite similar as earlier.
  + We also got a list of all the important features used by the model.
* The third model used was LogisticRegression.
  + It was also used with *GridSearchCV* to get hyperparameter tuning to get the best model.
  + We had the accuracy of 69.5%.
  + We got the coefficients for each of our features which can be interpreted for getting the important positive or negative features.

1. **Relevant code:**

The code was done in python, which can easily be converted to R as the approach remains the same and the libraries also have their alternatives in both the languages.

The code is attached with the mail.

1. **Other ideas/Future enhancements:**

Based on this analysis, we found out the following things:

* Features like Members annual income their age at issue, their term years, and membership Package Type B are the most important features as they were consistent throughout all the used model’s results. These features can be used to improve our policies and strengthen the customer relationship.
* We had exploratory plots which gives us an insight to the relation between a feature and the target variable. We could see that there is a lot of overlapping in the data, which means that the features can’t strongly classify the user’s belonging to a particular class.
* The churn probabilities we get from the xgboost classifier can be used as getting some insights like direct user reviews about their reason for staying or churning the membership. And accordingly, the package or the business plan can be enhanced.
* Definitely a lot things can be added to the analysis given more time and resources.
* We can try to get further more data to get better accuracy from the model.