**Project: Predicting Coupon Redemption**

Website: <https://www.kaggle.com/vasudeva009/predicting-coupon-redemption>

Github:

**Introduction**

This project is based on a dataset from Kaggle.com. It is about a ficticious XYZ Credit Card company that helps its merchants understand their data better and take key business decisions accurately by providing machine learning and analytics consulting. ABC is an established Brick & Mortar retailer that frequently conducts marketing campaigns for its diverse product range. As a merchant of XYZ, they have sought XYZ to assist them in their discount marketing process using the power of machine learning [1].

Data has been collected on some aspects related to campaigns, coupon redemption, demographics and customer transactions, in order to measure in a extend the effectiveness of marketing campaigns. The retailer wants to make informed decisions by predicting the customer behavior in coupon redemption and also build better campaigns. This document analyses the elements that, eventually affect customer behavior in coupon redemption.

**Dataset description**

The data available for this problem is from the retail that describe the following aspects of the problem:

User demographics details

Campaign and coupon details

Product details

Previous transactions

Every one of these aspects has a specific csv file that describes them, the files are train.csv, campaign\_data.csv, coupon-item\_mapping.csv, customer \_demographics.csv, item\_data.csv and test.csv.

**Exploratory Data Analysis**

The exploratory data analysis was performed by analyzing the context of the problem and the characteristics of the variables as observed in the directory of the project, in this case “exploration.ipynb” in the notebooks directory of the project.

The target feature **redemption\_status** is binary and unbalanced, that is why it have been balanced after the exploration process. Most of the numeric features are skewed. It had not been found correlated features besides from **campaign\_id** correlated to **start\_date** and **end\_date**, both deleted **campaign\_id** kept.

In the case of missing values in the features, it had been found some kind of patterns in some of them, these patterns look like that the customers did not want to give the information for unknown reasons. from these features, the ones that have percentages over 40% have been deleted, some others have around 30%, these features have been processed to deal with them.

Something very interesting is that the feature **campaign\_id** is kind of leaked because it had been detected that in the trainset we have in the values from 1 to 13, also 26, 27, 28, 29, but in the case of the test set from 17 to 25, and there does not exist 14, 15, 16, that is why it had not been used 26 to 29 in train because they represents campaigns in test, meaning that the sequence had been preserved.

In the case of feature engineering, there were the possibility to create some new features that helped the models, these are:

daysCampaign : end\_date - start\_date

month: month(date)

day: day(date)

price: selling\_price-other\_discount

In general, the resulting features had been standardized and in some cases it have been used one\_hot encoding to process them, in the specific case of the feature **category**, some special characters or spaces had been deleted in order to apply one-hot encoding.

**Performance Metric**

The performance metric is evaluated using the accuracy, the redemption\_status feature is imbalanced, that is why some preprocessing to balance this feature had been done.

**Modeling**

Stratified cross-validation had been used to have five folds in the dataset. the techniques logistic regression, Random Forest and XGBoost had been used to process the dataset. The results in the case of logistic regression were not so good, but in the case of Random Forest and XGBoost the results were very good.

The results of the mentioned techniques were stored in order to use blending, it had been calculated average and weighted average and ranked average with the probabilities obtained from the models. In this case there were and increment in the accuracy result.

With the probability results stored, it had been used stacking combining the model probabilities and using linear regression for the predictions which made the accuracy even better.

**Prediction Results**

With all the processes mentioned in the modeling point, the accuracy had been becoming better every time until we reach an accuracy of 0.97 using stacking wich is a very good result. Getting that high results it had not been found some clues that made us think that overfitting had take place.

**Bibliography**

[1] <https://www.kaggle.com/vasudeva009/predicting-coupon-redemption>