**Hackathon Machine Learning**

**Description**

Personalisation of offers for the customer is paramount for the bank as there is only a limited number of communication the bank is allowed to send in a timeframe(generally a week) due to cost and regulatory aspects. Personalisation not only involves identifying the best offers to the customer but also the **best send time for those offers** so their likelihood to be read/opened is maximised.

In this competition, you are tasked with predicting the **optimal time slot** for sending marketing communications to a set of customers, in order to maximize their **engagement**. Engagement can be defined in terms of actions such as **opening** an email. The challenge is to predict when each customer is most likely to engage with the communications based on their past behaviour and historical interaction data. For simplicity, we restrict to single channel, i.e. **Email** and events for a communication can be send and open. Corresponding event for a communication can be identified by offer\_subid and batch\_id.

Customers have unique preferences for when they engage with marketing communications. Your goal is to predict the best time slot (e.g., a combination of time of day and day of the week) for each customer from a set of available time slots.

**Rules**

**Data**

**Communication History (6 months for 2 lakh customers) :**

It contains the interaction data of the customer and the offer. Below are the relevant columns in action history:

1. Customer\_Code : It’s the unique identifier for the customer
2. Offer\_id: Identifier for the offer
3. Offer\_subid : Identifier for the subid of an offer.(One offer\_id can have multiple offer\_subid)
4. Product category and subcategory of the offer : Each offer sent is for specific product category and subcategory
5. Batch\_id: Identifier for the batch. Helpful in tracking events on same communication.
6. Send\_timestamp: Date & time at which the email was sent
7. Open\_timestamp: Date & time at which the email was opened. Null implies the email was not opened

**Customer CDNA:**

It contains customer level features captured like Demographics etc. This data is generated at a regular frequency(generally weekly) which is captured in the **batch\_date** column. Make sure you join the cdna captured under latest date and less than the send event in action history.

Note: Not all features in cdna are important/relevant.

**Time Slots definition**: Time\_slot window is of 3 hours. Starting from 9:00 AM to 9:00 PM, that makes 4 slots on each day and overall 28 slots for the complete week.

* + **slot\_1** represents the first time slot (Monday 9:00 AM- 12:00 PM)
  + **slot\_2** represents the second time slot (Monday 12:00 PM – 3:00 PM)

**.**

**.**

**.**

* + **slot\_28** represents the 28th time slot (Sunday 6:00 PM – 9:00 PM)

The communication history table can be used for two purposes :

1. To create target variable i.e. to create slots for predicting if a customer is going to open the email in the same slot
2. To create independent variables i.e. what slots are customer opening in and that can be used to predict the target variable.

**Evaluation**

Evaluation will be based on the metric MAP.

**MAP** is a ranking metric that evaluates how well the predicted time slots are ordered, particularly focusing on the **precision of relevant time slots** at different ranks in the prediction list.

The time slots in which customers have received and opened the email, will be marked as relevant.

Time slots in which customers have received and not opened the email will be marked as not relevant. Time slots where no communication was sent for the customer, will be excluded from both the test and the submitted results.

The priority provided against the customer and the relevance generated will be used for calculating MAP.

**Evaluation data Format**

The test set will contain a different set of Customer\_codes, corresponding cDNA and Communication History for previous 3 months. The next 1 month data will be used for evaluating the MAP.

**Submission Format**

You must submit a csv file containing two columns. The first columns are named "**customer\_code**" and The second column is named “**predicted\_slots\_order”** which is a list of 28 elements with the best slot being the first element and second best slot being the 2nd element and so on.

|  |  |
| --- | --- |
| customer\_code | predicted\_slots\_order |
| 001 | [slot\_21, slot\_27, slot\_4, slot\_12, slot\_10,………] |
| 002 | [slot\_1, slot\_7, slot\_4, slot\_12, slot\_27, ……….] |
| ... | ... |

**Column description**

1. **Customer\_code**:

* **Type**: String
* A unique identifier for each customer in the dataset. This column will be used to match the predictions with the corresponding customers in the test set.

1. **predicted\_slots\_order**:
   * Type: list of strings
   * This is the predicted list of slots in the decreasing order of relevance. First being the best slot and last being the worst slot. **The length of this list should be 28 and no element should repeat**.