

# Email Marketing Campaign Analysis

This repository contains code for analyzing email marketing campaign data and building a predictive model to optimize future email sends.

## Overview

The analysis aims to answer the following questions:

1. What percentage of users opened the email and clicked on the link within the email?
2. How can we build a model to maximize the probability of users clicking on links inside emails?
3. How much would this model improve click-through rates compared to random selection?
4. What patterns exist in how different user segments respond to the email campaign?

## Data Description

The analysis uses three CSV files:

1. **email\_table.csv**: Information about each sent email
  - email\_id: Unique identifier
  - email\_text: "short\_email" (2 paragraphs) or "long\_email" (4 paragraphs)
  - email\_version: "personalized" or "generic"
  - hour: Time when email was sent (0-23)
  - weekday: Day of the week when email was sent
  - user\_country: Recipient's country
  - user\_past\_purchases: Number of previous purchases by the recipient
2. **email\_opened\_table.csv**: Contains IDs of emails that were opened
  - email\_id: Identifiers of opened emails
3. **link\_clicked\_table.csv**: Contains IDs of emails where links were clicked
  - email\_id: Identifiers of emails with clicked links

## Requirements

- Python 3.x
- pandas
- numpy
- matplotlib
- seaborn
- scikit-learn

# Installation

```
pip install pandas seaborn scikit-learn numpy matplotlib
```

## Usage

1. Upload the three CSV files to your environment (Google Colab, local Python, etc.)
2. Run the script to perform the complete analysis:

```
Run email_marketing_analysis.py
```

```
#
```

Code Structure

The script performs the following analyses:

### 1. Data Loading and Preparation:

- Loads the three datasets
- Creates binary flags for opened and clicked emails
- Creates 2-hour time intervals and purchase categories

### 2. Campaign Performance Analysis:

- Calculates overall open rate, click-through rate, and click-to-open rate
- Visualizes performance across different segments

### 3. Predictive Modeling:

- Builds a Random Forest model to predict click probability
- Evaluates model performance
- Visualizes feature importance

### 4. CTR Improvement Estimation:

- Calculates potential CTR improvement using the model
- Visualizes lift over random selection

### 5. Segment Analysis:

- Analyzes performance by email type, personalization, time, day, country, and purchase history
- Creates visualizations for each segment

- Identifies interaction effects between different factors

## Visualizations

The code generates multiple visualizations:

- Feature importance for the predictive model
- CTR improvement curve
- Performance by email characteristics (length, personalization)
- Performance by timing (hour, day of week)
- Performance by user segments (country, purchase history)
- Heatmaps showing interaction effects

## Results

The executive summary document provides a detailed overview of the analysis findings, including:

- Campaign performance metrics
- Most effective email characteristics
- Predictive model performance
- Segment-specific insights
- Strategic recommendations

## Customization

To adapt this analysis to your own email campaign data:

1. Ensure your data follows a similar structure with email details, opens, and clicks
2. Adjust the feature engineering steps if your data includes different attributes
3. Modify the visualization parameters to match your branding
4. Update the binning for time intervals and purchase categories as needed