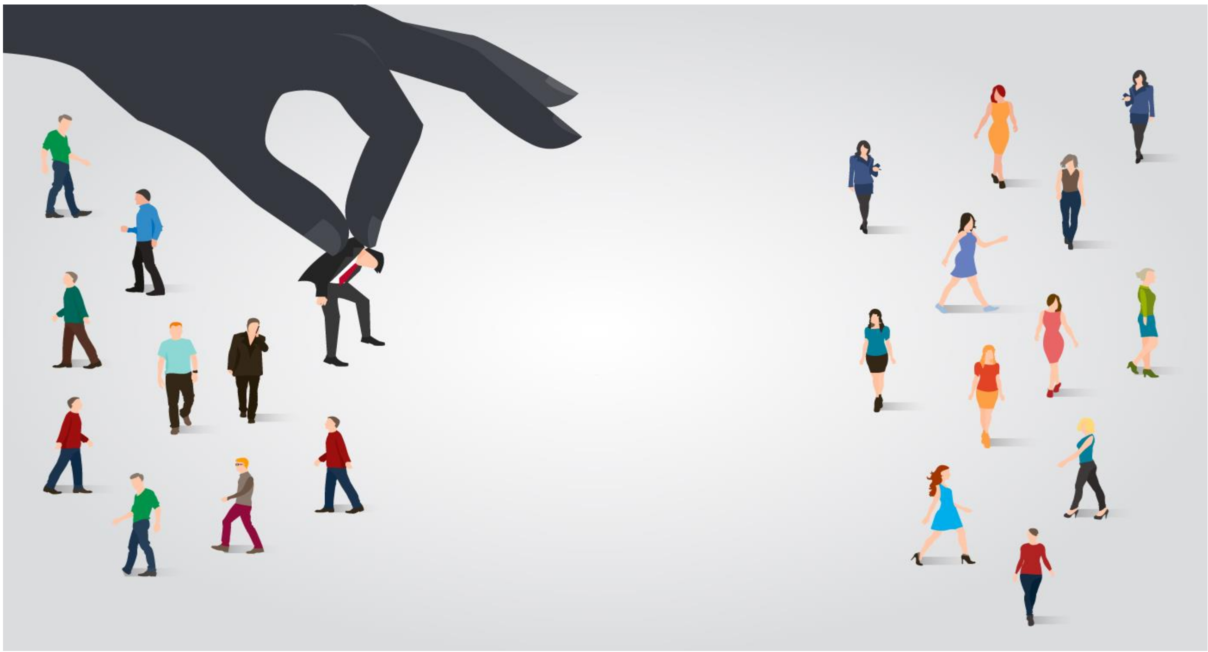
ANALYTICS VIDYA HACATHON

Regression Model

Submitted by

Vijjeswarapu Surya Teja



Problem Statement:

Most organizations today rely on email campaigns for effective communication with users. Email communication is one of the popular ways to pitch products to users and build trustworthy relationships with them. In this hackathon, we need to build a smart system to predict the CTR for email campaigns and therefore identify the critical factors that will help the marketing team to maximize the CTR.

Business Benefits:

CTR is a measure of success for email campaigns. The higher the click rate, the better your email marketing campaign is.

Data Dictionary:

We are provided with 3 files - train.csv, test.csv and sample_submission.csv.

Train and Test set contains different sets of email campaigns containing information about the email campaign. Train set includes the target variable *click_rate* and you need to predict the *click_rate* of an email campaign in the test set.

Variable	**Description**
campaign_id	Unique identifier of a campaign
sender	Sender of an e-mail
subject_len	No. of characters in a subject
body_len	No. of characters in an email body
mean_paragraph_len	Average no. of characters in paragraph of an email
day_of_week	Day on which email is sent
is_weekend	Boolean flag indicating if an email is sent on weekend or not
times_of_day	Times of day when email is sent: Morning, Noon, Evening
category	Category of the product an email is related to
product	Type of the product an email is related to
no_of_CTA	No. of Call To Actions in an email
mean_CTA_len	Average no. of characters in a CTA
is_image	No. of images in an email
is_personalised	Boolean flag indicating if an email is personalized to the user or not
is_quote	No. of quotes in an email
is_timer	Boolean flag indicating if an email contains a timer or not
is_emoticons	No. of emoticons in an email
is_discount	Boolean flag indicating if an email contains a discount or not
is_price	Boolean flag indicating if an email contains price or not
is_urgency	Boolean flag indicating if an email contains urgency or not
target_audience	Cluster label of the target audience
click_rate (Target Variable)	*Click rate of an email campaign*

CODE APPROACH:

The problem approach in code is divided into following parts

- Data Visualisation
- Data Cleaning and Feature Engineering
- Data Pre-Processing
- Model Selection
- Results Interpretation

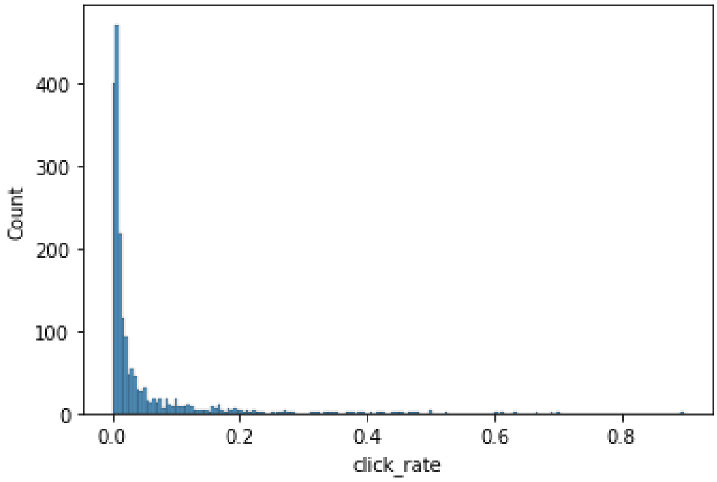
Libraries Used:

- Pandas
- Numpy
- Scikit learn
- matplotlib
- seaborn
- xgboost
- lightgbm
-

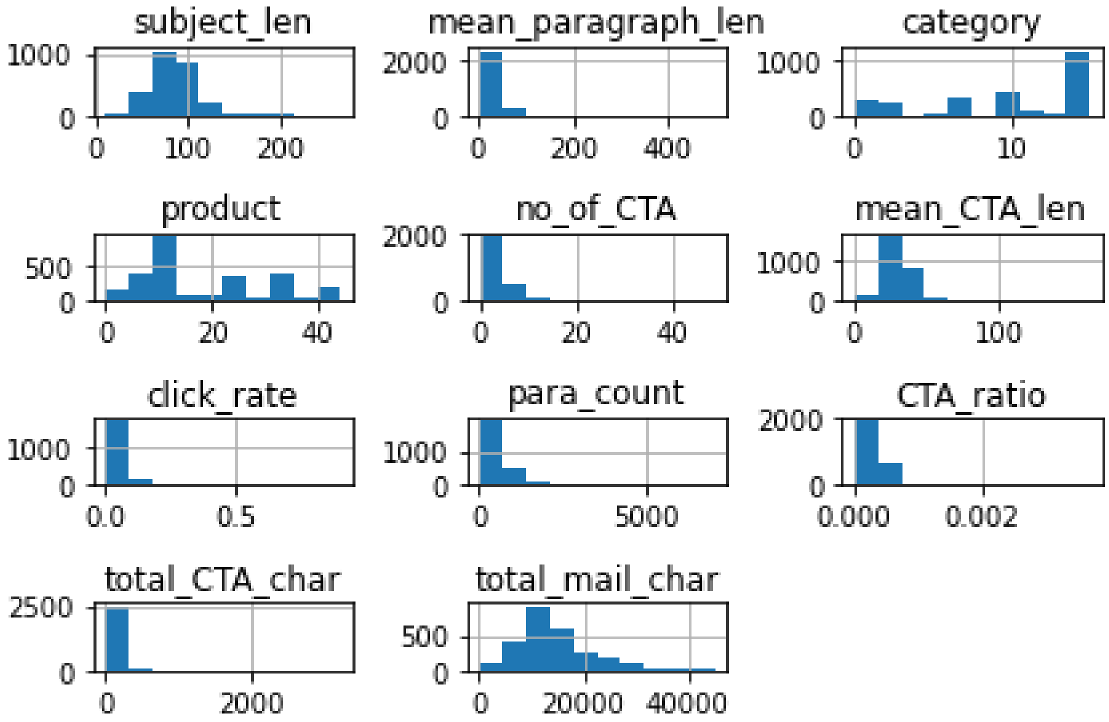
1) Data Visualisation

Since this is a very small dataset with only few features there isn't much in terms of plots and most of the data is pretty straightforward

'click_rate' column: The plot represents left skewed distribution. No transformations since I was using Catboost, it was robust for skewed data



All other varibales are more or less uniformly distributed and somewhere left skewed like the click_rate columns above



2) **Data Cleaning and Feature Engineering:**

Data Info:

After importing the libraries, the data is converted to a dataframe using Pandas for data cleaning and feature engineering. The train and test data sets are combined into a single dataset for uniformity while filling the missed values in data.

The data set contains 22 columns and with only times_of_day as the categorical column and and remaining are numerical columns. All the columns and data values are pretty straightforward and this being a very small dataset there aren’t many hidden features in the dataset

Handling missing values & Outliers:

There are no missing values in the dataset and to handle outliers in the dataset, in the final model selection catboost was chosen

Creating New Features:

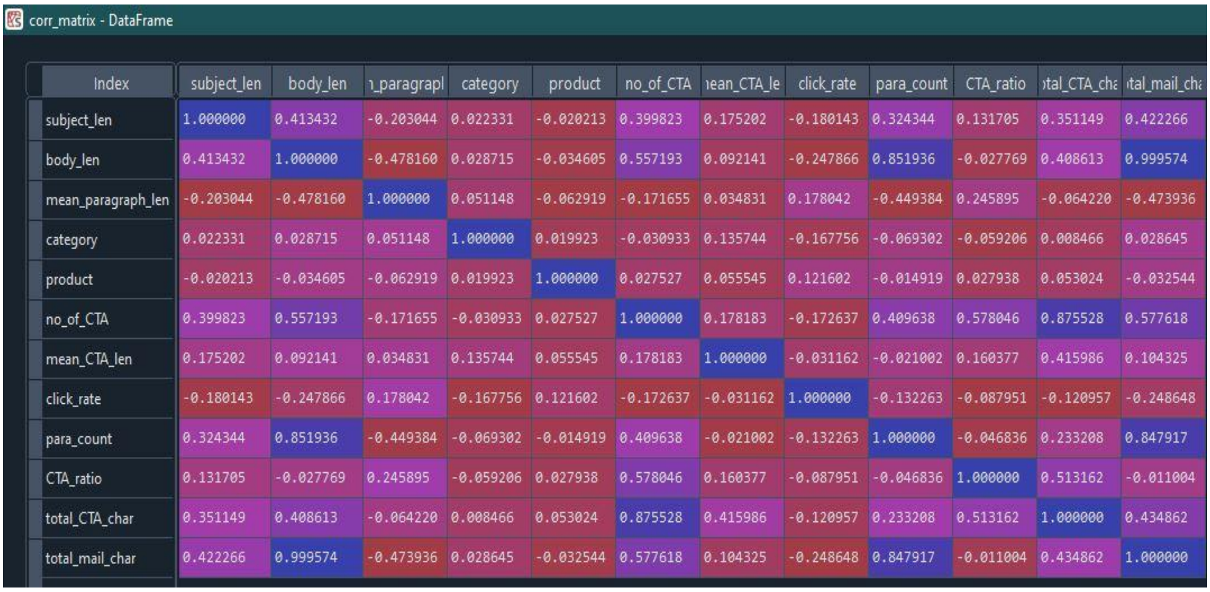
Different combinations have been made to create new features and features having high multicollinearity were dropped

- Total characters used in the entire mail
- CTA ratio based on available CTAs in body length
- Total CTA characters in the entire mail
- Total para's count in the mail

Dropping features:

Obvious columns like 'campaign_id' and 'is_timer' with no value chave been dropped The features which have high multi collinearity (more than 0.85) have been dropped.

Correlation Matrix



Index	subject_len	body_len	ı_paragrapl	category	product	no_of_CTA	ıean_CTA_le	click_rate	para_count	CTA_ratio	ıtal_CTA_chа	ıtal_mail_chа
subject_len	1.000000	0.413432	-0.203044	0.022331	-0.020213	0.399823	0.175202	-0.180143	0.324344	0.131705	0.351149	0.422266
body_len	0.413432	1.000000	-0.478160	0.028715	-0.034605	0.557193	0.092141	-0.247866	0.851936	-0.027769	0.408613	0.999574
mean_paragraph_len	-0.203044	-0.478160	1.000000	0.051148	-0.062919	-0.171655	0.034831	0.178042	-0.449384	0.245895	-0.064220	-0.473936
category	0.022331	0.028715	0.051148	1.000000	0.019923	-0.030933	0.135744	-0.167756	-0.069302	-0.059206	0.008466	0.028645
product	-0.020213	-0.034605	-0.062919	0.019923	1.000000	0.027527	0.055545	0.121602	-0.014919	0.027938	0.053024	-0.032544
no_of_CTA	0.399823	0.557193	-0.171655	-0.030933	0.027527	1.000000	0.178183	-0.172637	0.409638	0.578046	0.875528	0.577618
mean_CTA_len	0.175202	0.092141	0.034831	0.135744	0.055545	0.178183	1.000000	-0.031162	-0.021002	0.160377	0.415986	0.104325
click_rate	-0.180143	-0.247866	0.178042	-0.167756	0.121602	-0.172637	-0.031162	1.000000	-0.132263	-0.087951	-0.120957	-0.248648
para_count	0.324344	0.851936	-0.449384	-0.069302	-0.014919	0.409638	-0.021002	-0.132263	1.000000	-0.046836	0.233208	0.847917
CTA_ratio	0.131705	-0.027769	0.245895	-0.059206	0.027938	0.578046	0.160377	-0.087951	-0.046836	1.000000	0.513162	-0.011004
total_CTA_char	0.351149	0.408613	-0.064220	0.008466	0.053024	0.875528	0.415986	-0.120957	0.233208	0.513162	1.000000	0.434862
total_mail_char	0.422266	0.999574	-0.473936	0.028645	-0.032544	0.577618	0.104325	-0.248648	0.847917	-0.011004	0.434862	1.000000

3) Data Pre-Processing

The dataset was divided into train and test sets. No scaling was done as we are going to feed the categorical features directly into CatBoost Model

4) Model Selection

Since this is a Regression problem the following models were selected to test the results.

- XGBoost
- LightGBM
- CatBoost

GridSearchCV was used to find the better working model with hyper parameters among the above models

CatBoostRegressor gave better results with the following parameters

- depth: 4
- learning_rate: 0.1
- iterations:1000

5) Results Interpretation:

The CatBoost model with above parameters was used to on the 70% train data to predict the results for 30% of test data

Results:

- r2_score: 0.54

CONCLUSION

The model gave a score of 0.67 in Analytics Vidya leaderboard on Private data