Problem Statement:

A D2C start-up develops products using cutting edge technologies like Web 3.0. Over the past few months, the company has started multiple marketing campaigns offline and digital both. As a result, the users have started showing interest in the product on the website. These users with intent to buy product(s) are generally known as leads (Potential Customers).

Leads are captured in 2 ways - Directly and Indirectly.

Direct leads are captured via forms embedded in the website while indirect leads are captured based on certain activity of a user on the platform such as time spent on the website, number of user sessions, etc.

Now, the marketing & sales team wants to identify the leads who are more likely to buy the product so that the sales team can manage their bandwidth efficiently by targeting these potential leads and increase the sales in a shorter span of time.

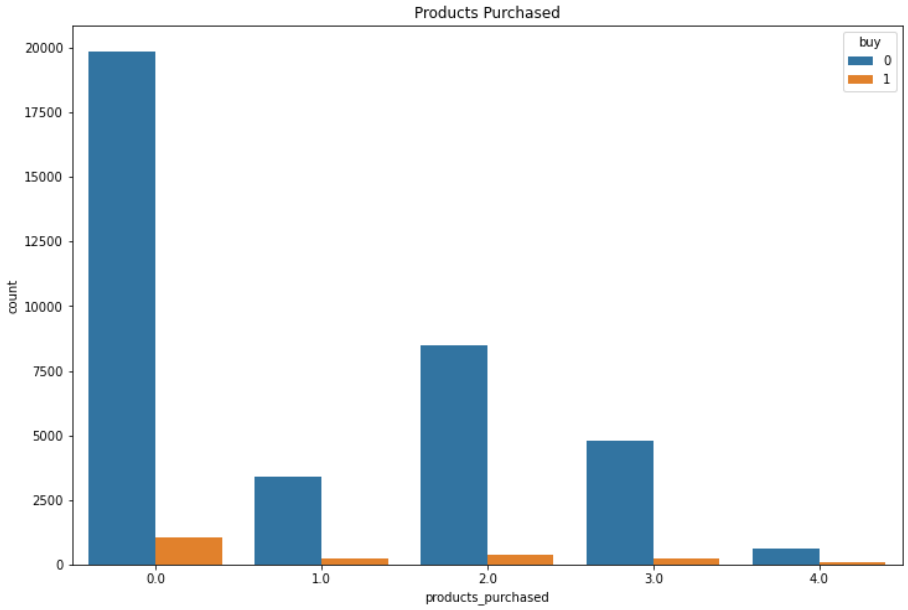
Now, as a data scientist, your task at hand is to predict the propensity to buy a product based on the user's past activities and user level information.

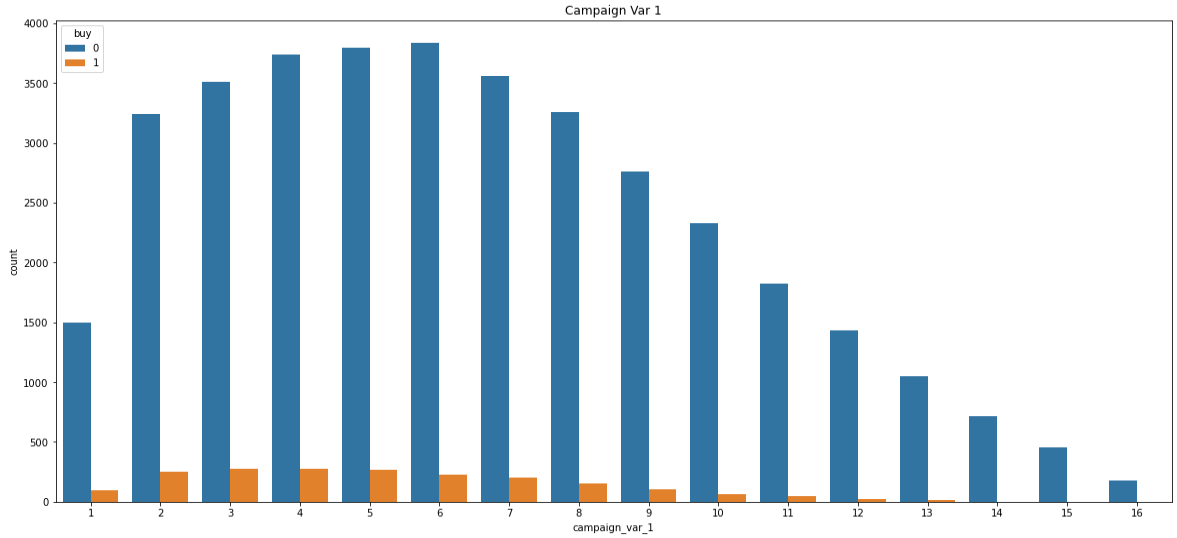
Approach:

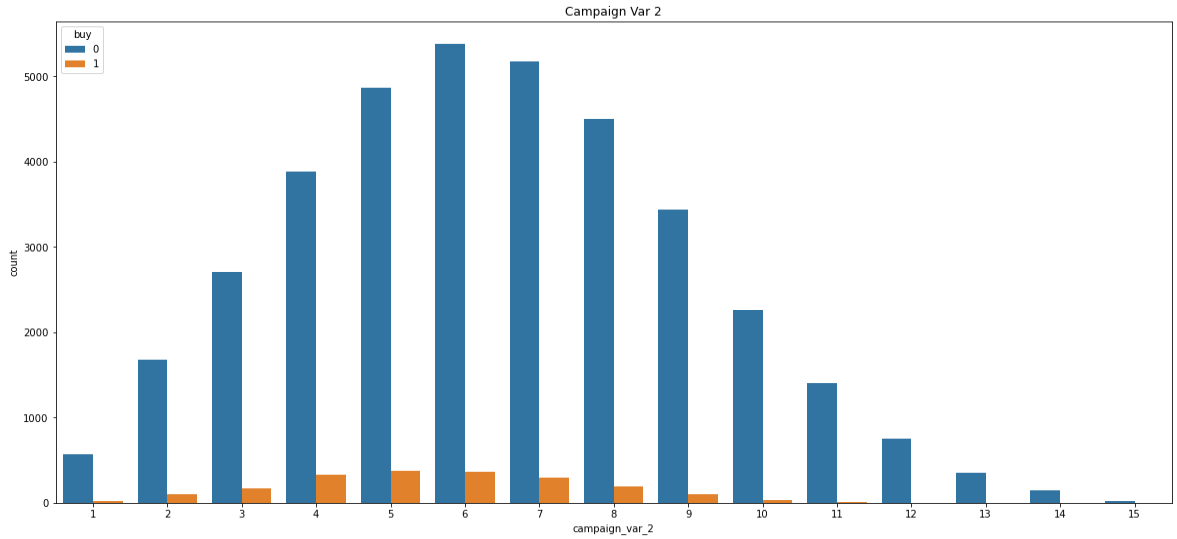
I have used the "**Logistic Regression**" method to build the prediction model for the above mentioned problem statement. I have used the "**scikit-learn**" library to import the "**Logistic Regression**" function and "**statsmodels**" library to import the "**variance_inflation_factor**" function to calculate the VIF.

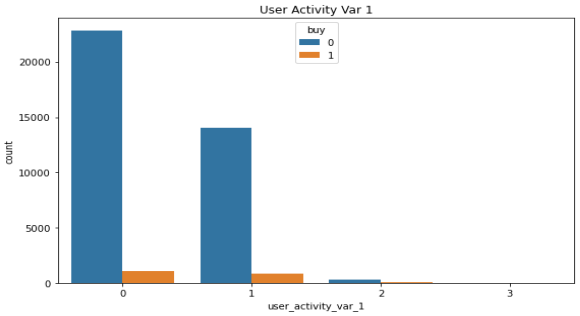
To begin with, I read the train dataset provided using the pandas and stored it as a pandas dataframe. Then, I did some basic checks on the dataframe like checking the shape of the dataframe, number of unique values in a column, missing values in the column, etc. Then, I did the data imputation for the missing values, created some new columns from the existing columns which were useful in model building and deleted few columns which were not relevant for model building.

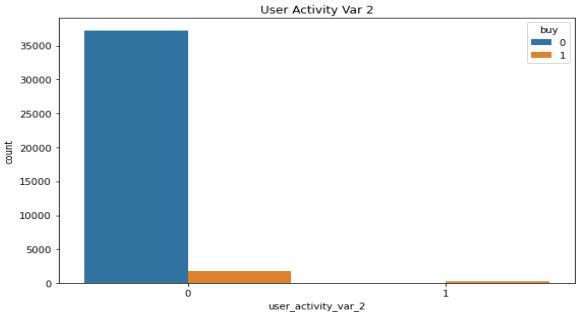
After that I did some basic EDA on the dataframe with respect to independent variables and dependent variable. Also, plotted a heat map for the correlation of the variables. Some of the graphs plotted during the EDA are shown below.

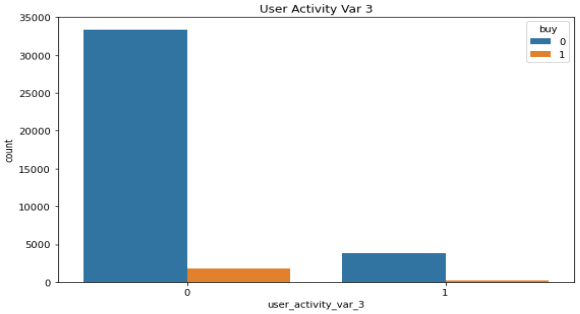

Products Purchased
buy
0
1
count
20000
17500
15000
12500
10000
7500
5000
2500
0
0.0
1.0
2.0
3.0
4.0
products_purchased

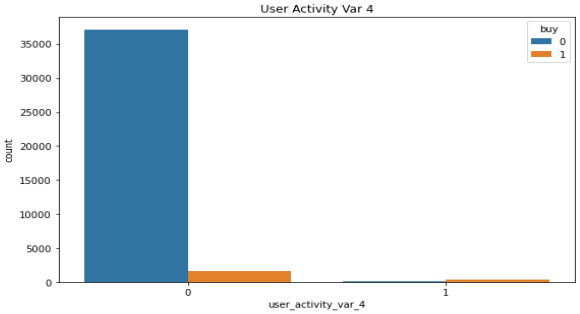

Campaign Var 1
buy
0
1
count
4000
3500
3000
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
campaign_var_1

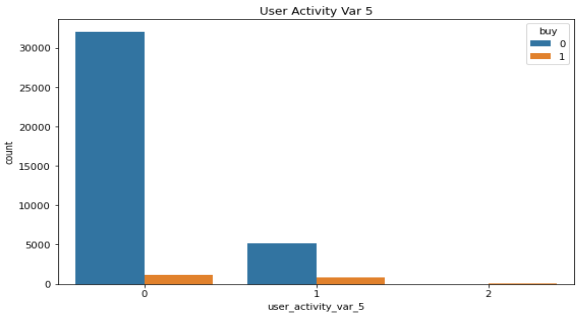

Campaign Var 2
buy
0
1
count
5000
4000
3000
2000
1000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
campaign_var_2

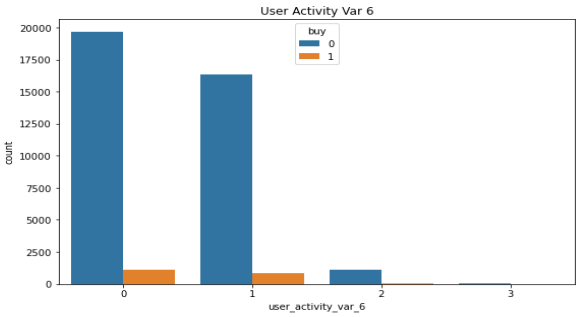

User Activity Var 1
buy
0
1
count
user_activity_var_1

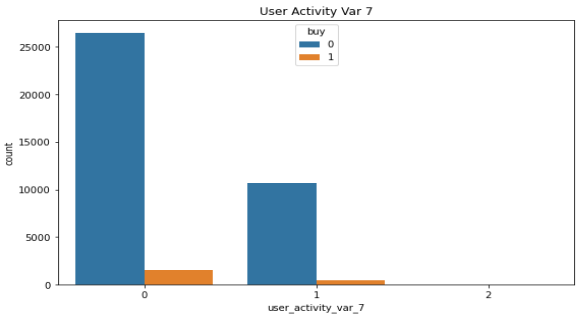

User Activity Var 2
buy
0
1
count
user_activity_var_2

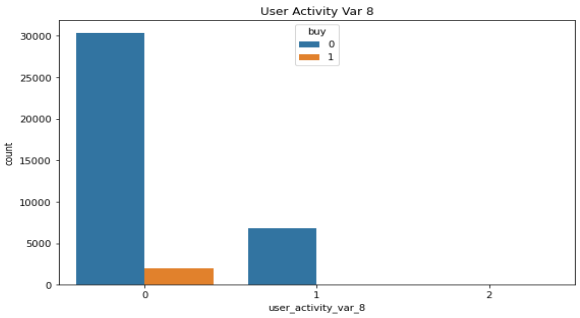

User Activity Var 3
buy
0
1
count
user_activity_var_3

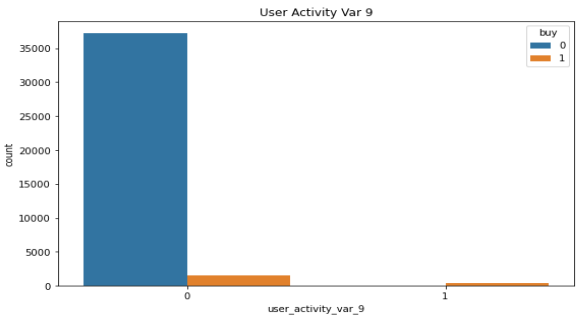

User Activity Var 4
buy
0
1
count
user_activity_var_4

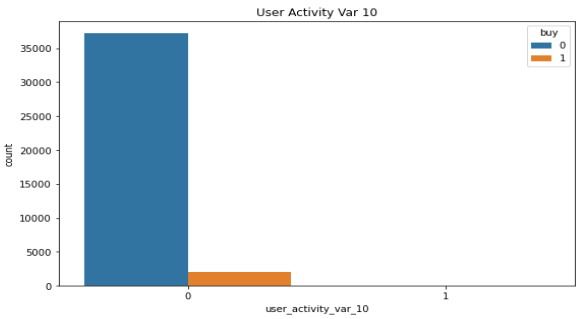

User Activity Var 5
buy
0
1
count
user_activity_var_5

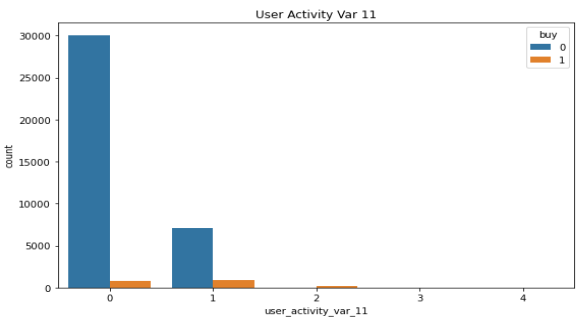

User Activity Var 6
buy
0
1
count
user_activity_var_6

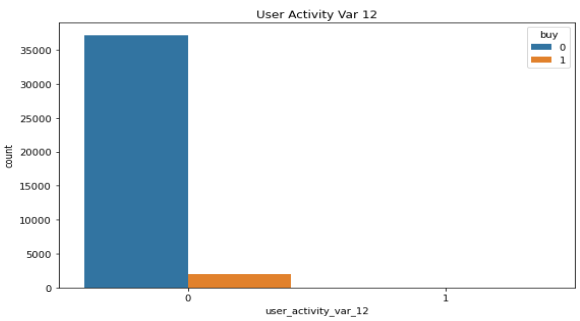

User Activity Var 7
buy
0
1
count
user_activity_var_7

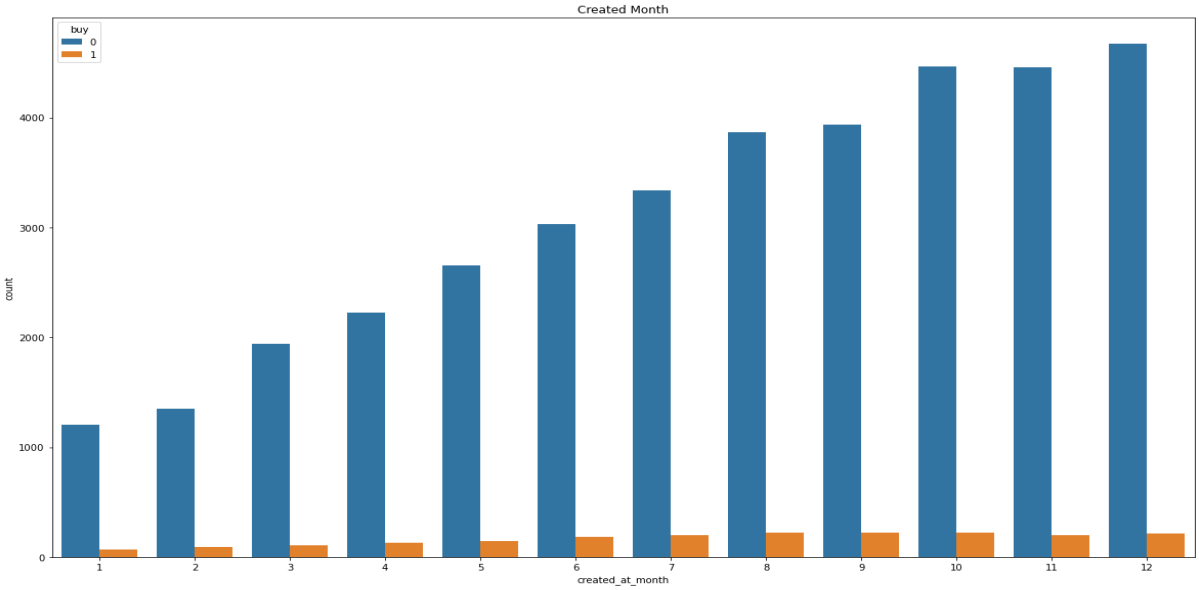

User Activity Var 8
buy
0
1
count
user_activity_var_8

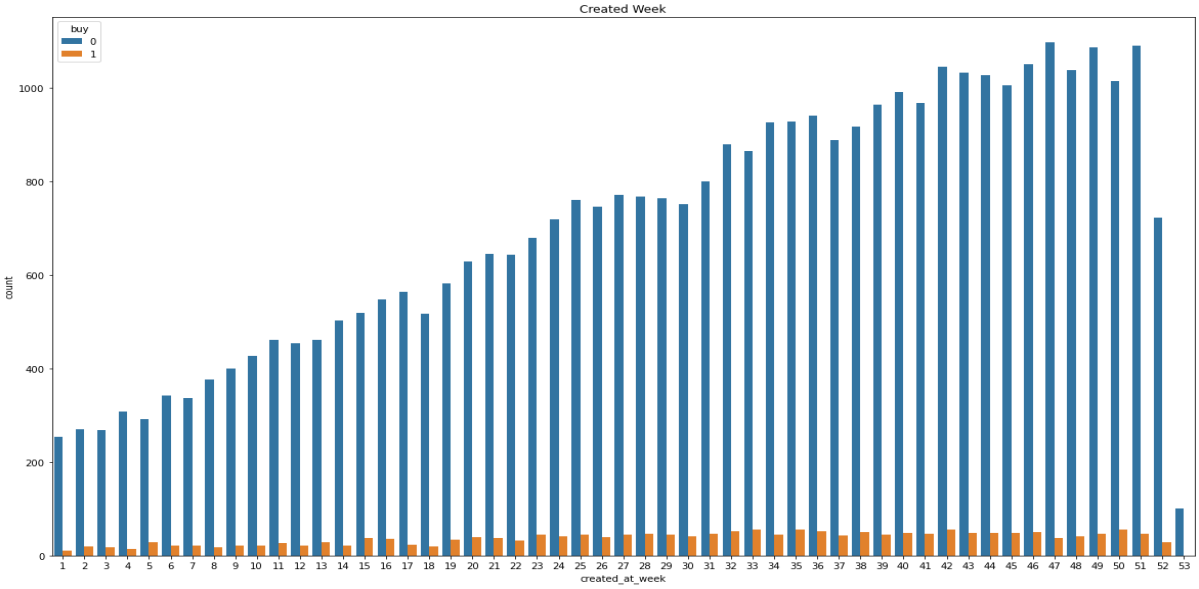

User Activity Var 9
buy
0
1
count
user_activity_var_9

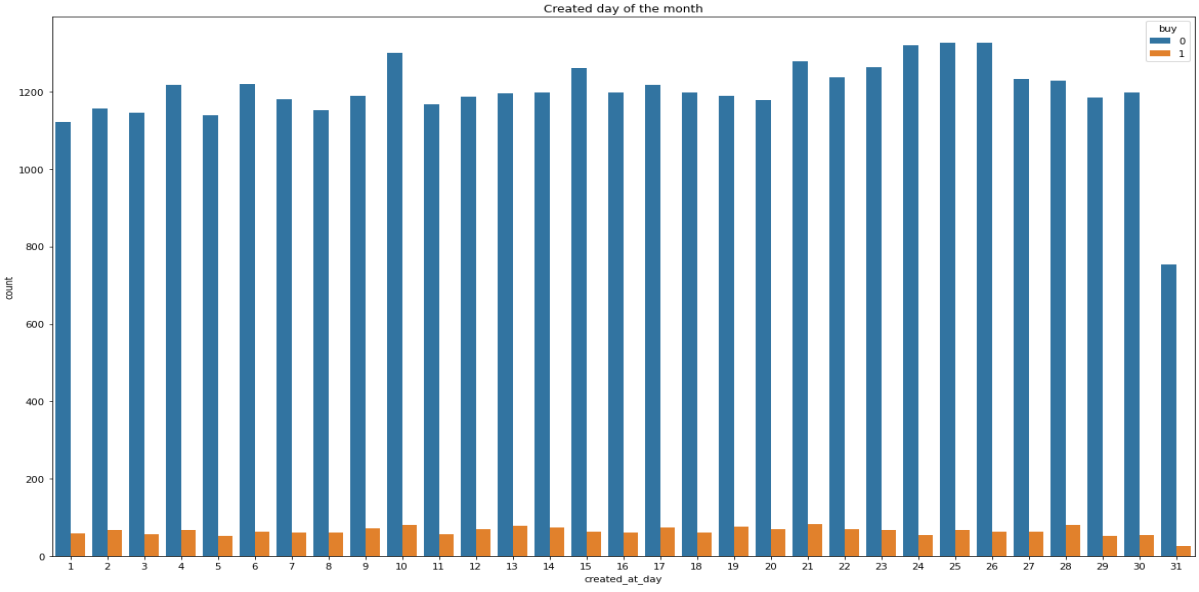

User Activity Var 10
buy
0
1
count
user_activity_var_10

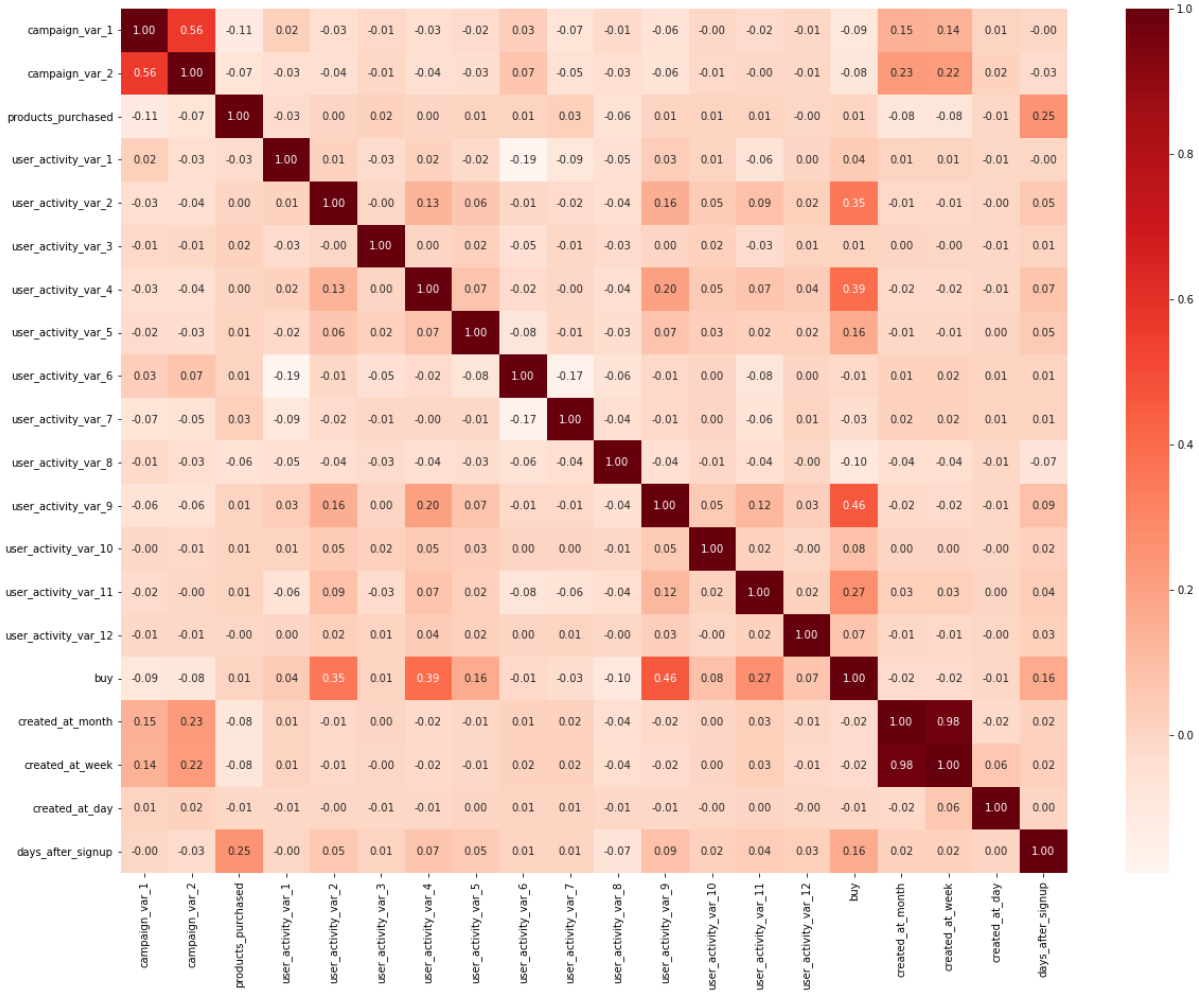

User Activity Var 11
buy
0
1
count
user_activity_var_11


User Activity Var 12
buy
0
1
count
user_activity_var_12


Created Month
buy
0
1
count
created_at_month


Created Week
buy
0
1
count
created_at_week


Created day of the month
buy
0
1
count
created_at_day



After the EDA, I imported the "**MinMaxScaler**" function from the "**scikit-learn**" library to scale all the variables in the range 0 to 1 so that all the variable values are in same scale. Then I started building the model using Logistic Regression. I went on to drop the variables (columns/features) one at a time as per the **p-values** obtained after the building the model and the **VIF** values. If the p-value if greater than **0.5** for any variable or VIF value is greater than **5** then, that variable can be dropped. Like that I went on to build the model and eliminate the variables one by one which were having p-value greater than 05 or VIF value greater than 5.

I got the final model after building 6 models. In this model, p-value for all the variables was less than 0.5 and VIF value less than 5. Below is the result from my final model.

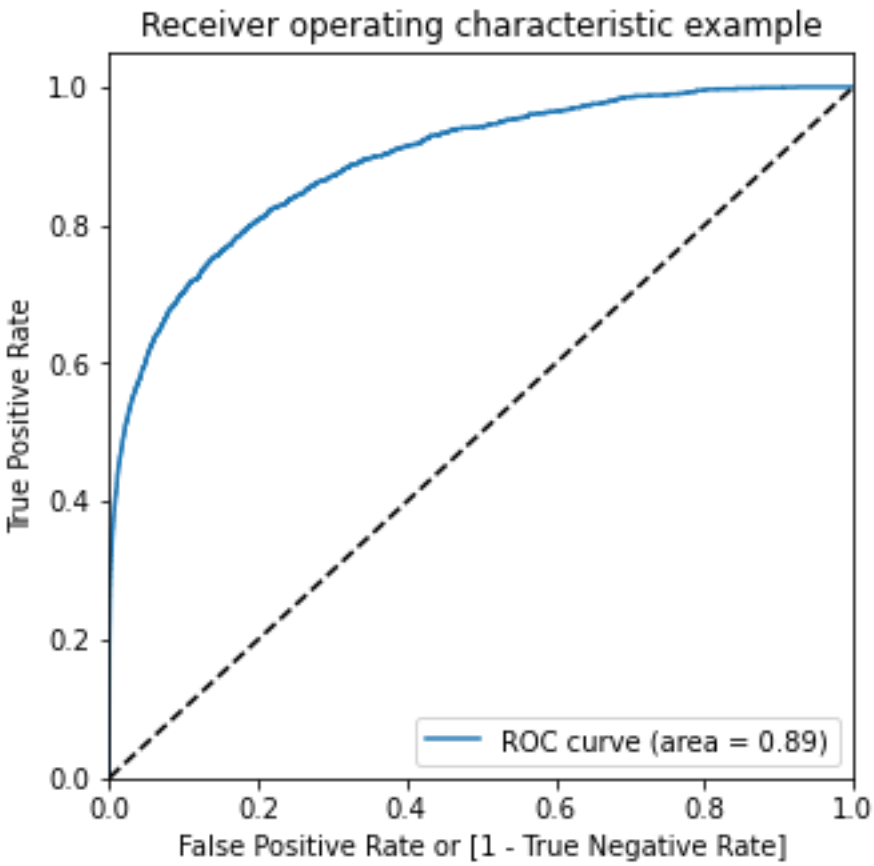
Generalized Linear Model Regression Results

Dep. Variable:	buy	**No. Observations:**	39161

Model:	GLM	**Df Residuals:**	39147
Model Family:	Binomial	**Df Model:**	13
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-4866.0
Date:	Mon, 06 Jun 2022	**Deviance:**	9732.1
Time:	09:49:15	**Pearson chi2:**	4.04e+04
No. Iterations:	9		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-4.5858	0.119	-38.449	0.000	-4.820	-4.352
campaign_var_1	-1.6638	0.164	-10.132	0.000	-1.986	-1.342
campaign_var_2	-1.0072	0.197	-5.114	0.000	-1.393	-0.621
products_purchased	-0.7794	0.099	-7.850	0.000	-0.974	-0.585
user_activity_var_1	1.9228	0.165	11.664	0.000	1.600	2.246
user_activity_var_4	5.1454	0.165	31.152	0.000	4.822	5.469
user_activity_var_5	2.8922	0.120	24.011	0.000	2.656	3.128
user_activity_var_6	0.9668	0.159	6.067	0.000	0.654	1.279
user_activity_var_7	-0.3536	0.135	-2.614	0.009	-0.619	-0.088
user_activity_var_8	-5.1244	0.418	-12.268	0.000	-5.943	-4.306
user_activity_var_11	8.6543	0.215	40.322	0.000	8.234	9.075
user_activity_var_12	3.6281	0.759	4.781	0.000	2.141	5.116
created_at_week	-0.3731	0.108	-3.443	0.001	-0.585	-0.161
days_after_signup	4.7341	0.193	24.505	0.000	4.355	5.113

After building the model, I made the predictions on the train dataset using the final model. The "**Accuracy**" of the model was found out to be **0.962**. The ROC curve was plotted on the "actual buy values" and the "predicted buy values". **Area under the ROC curve was 0.89** which is a very good value. Below is the ROC curve plotted.



Then, I made a table for **accuracy**, **sensitivity** and **specificity** to find out the correct cut-off value. Below is the table.

	probablity	accuracy	sensitivity	specificity
0.0	0.0	0.051020	1.000000	0.000000
0.1	0.1	0.907382	0.675175	0.919867
0.2	0.2	0.949286	0.537538	0.971423
0.3	0.3	0.959654	0.449449	0.987084
0.4	0.4	0.962207	0.381381	0.993434
0.5	0.5	0.962386	0.329830	0.996394
0.6	0.6	0.962054	0.290290	0.998170
0.7	0.7	0.960879	0.251752	0.999004
0.8	0.8	0.959041	0.208709	0.999381
0.9	0.9	0.956487	0.152152	0.999731

With the help of above table, I selected **0.4** as the **cut-off value** as it gives the perfect balance of all the three parameters i.e. accuracy, sensitivity and specificity.

At last, I read the test dataset using the pandas and stored it as pandas dataframe. Then I did the data cleaning and data transformation as I did for the train dataset. After that, I scaled all the columns using MinMaxScaler function and then predicted the target variable (buy) using the logistic regression model build earlier. Then, I stored the "id" and "buy" (predicted target variable) in a new dataframe and wrote the data into a CSV file. (Solution file)

Thank You!!