ANALYTICS VIDYA JOB-A-THON

PREDICTION MODEL

Submitted by

Vijjeswarapu Surya Teja

Problem Statement:

We are provided with the customer leads data of last year containing both direct and indirect leads. Each customer lead provides information about their activity on the platform, signup information and campaign information. Based on his/her past activity on the platform, we need to build the predictive model to classify if the user would buy the product in the next 3 months or not.

Business Benefits:

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

Data Dictionary:

We are provided with 3 files - train.csv, test.csv and sample_submission.csv and train.csv contains the leads information. And also, the target variable indicating if the user will buy the product

Variable	Description
id	Unique identifier of a lead
created_at	Date of lead dropped
signup_date	Sign up date of the user on the website
campaign_var (1 and 2)	campaign information of the lead
products_purchased	No. of past products purchased at the time of dropping the lead
user_activity_var (1 to 12)	Derived activities of the user on the website
buy	0 or 1 indicating if the user will buy the product in next 3 months or not

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

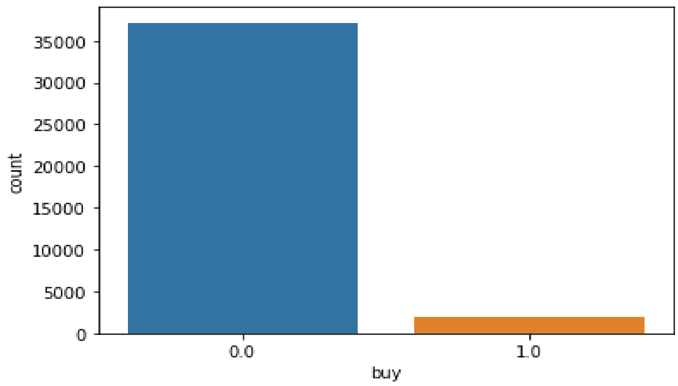
Libraries Used:

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

1) Data Visualisation

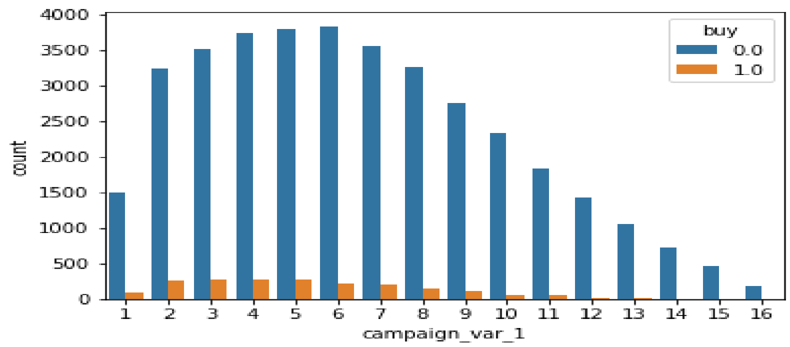
Since this is a very small dataset with only few features there isn't much in terms of plots and most of the data is pretty straightforward

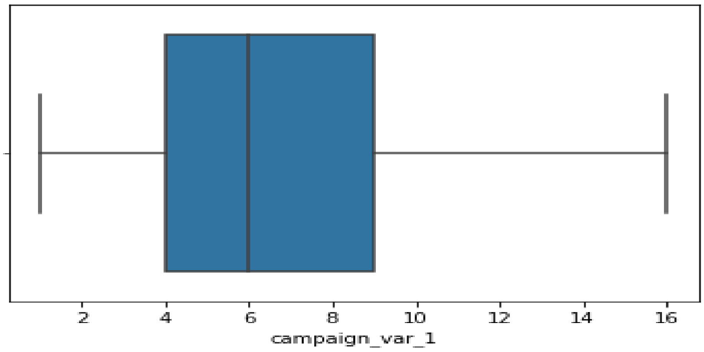
'buy' column: The plot represents an imbalanced dataset. Oversampling can help to predict more potential leads but the idea here is to manage sales bandwidth efficiently so oversampling the data might not be good as the sales team would be calling lot of undesired calls



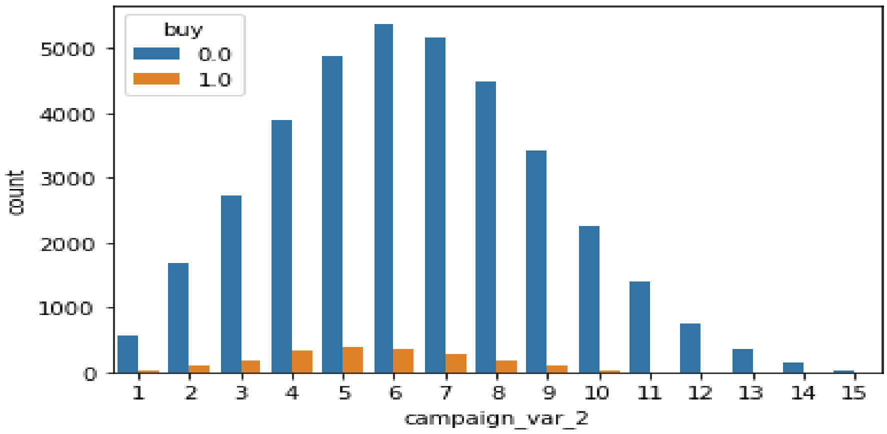
'campaign_var_1' and 'campaign_var_2':

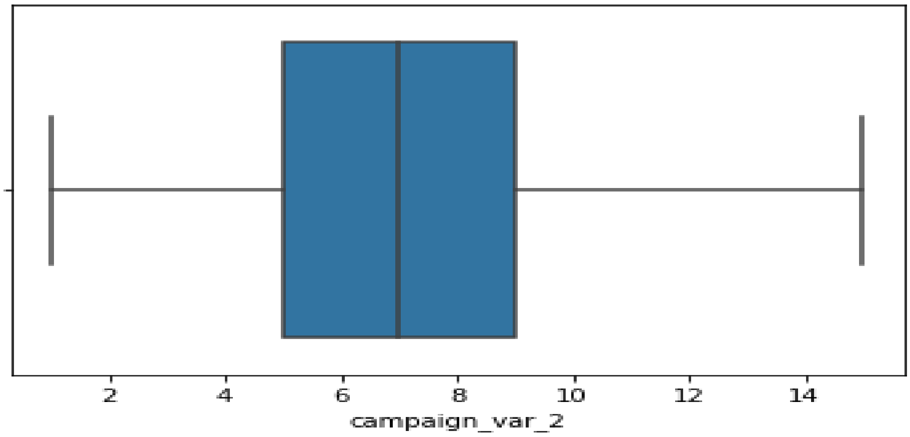
For campaign_var_1 buys almost equal distributed among 3,4,5 campaigns and no outliers in the data



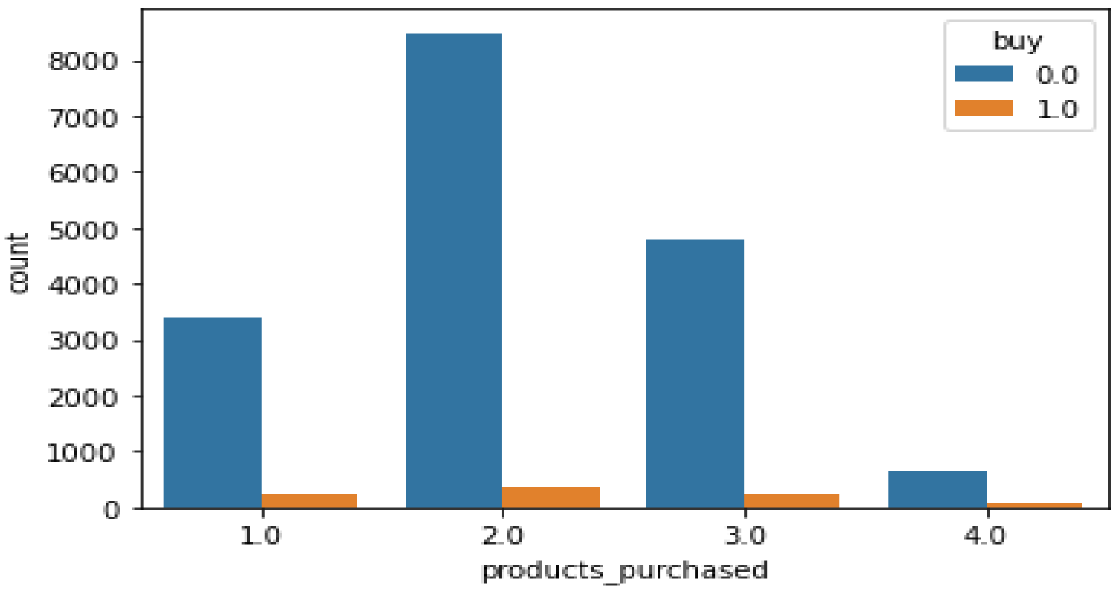


For campaign_var_2 buys almost equal distributed among 4,5,6 campaigns and no outliers in the data





‘products_purchased’ column: 'products_purchased' doesn't give much info as there were more than 50% missing values in the data



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 19 columns and with two date columns and remaining columns are numerical. There are no categorical datatypes. 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 only 2 columns with missing values and no outliers in the dataset

The 'products_purchased' column has 55.4% missing values so it was decided to drop the column as it may lead to erroneous model (The products_purchased column was imputed with zeros and mean values but still didn't benefit the model so it was dropped)

The 'signup_date' column has 41.5 % missing values. This is a very important feature as it describes user activity and interest so it was decided to impute the missing values. Since 40% of data is missing imputing with either mean or mode may hurt the model badly so I have decided to impute with another column which is 'created_at'. The 'created_at' column is when the lead was created date so it could better describe the user interest and better suits compared to either mean or mode values

Creating New Features:

Different combinations have been made to create new features and since they are not benefitting the model and having high multicollinearity they were dropped and only few new features as below have been used

- Combined all the useractivity varibales into one single column as 'total_activity' column
- Combined all campaign variables into one single column as 'total_campaigns' column
- Created three separate columns from 'signup_date' date column as 'signup_day', 'signup_month', 'signup_year'
- 'signup_date' is converted into one useractivity varibale with missing values replaced with zeros

Dropping features:

Obvious columns like 'id' column have been dropped earlier. The features which have high multi collinearity (more than 0.85) have been dropped. The date columns have been dropped as new additional features were created from them. 'products_purchased' column was dropped due to having more than 50% missing values

Correlation Matrix

corr_matrix - DataFrame

Index	:ampaign_var_1	:ampaign_var_2	:r_activity_va	:r_activity_va	:r_activity_va	:r_activity_va	:r_activity_va	:r_activity_va	:r_activity_va	:r_activity_va	:r_activity_va	r_activity_var	r_activity_var	r_activity_var	buy	signup_day	ignup_montl	signup_year	:r_activity_va	tal_campaigı	total_activity
campaign_var_1	1.000000	0.575101	0.020371	-0.037443	-0.010616	-0.034883	-0.024671	0.030934	-0.072626	-0.018211	-0.058469	-0.008395	-0.013608	-0.007810	-0.087202	0.001964	-0.012685	0.083854	-0.155888	0.917114	-0.108349
campaign_var_2	0.575101	1.000000	-0.039925	-0.045376	-0.007585	-0.045739	-0.033279	0.068066	-0.047743	-0.042167	-0.065208	-0.010986	-0.002274	-0.009580	-0.080064	0.005235	-0.020733	0.140935	-0.183422	0.853541	-0.126989
user_activity_var_1	0.020371	-0.039925	1.000000	0.017601	-0.023360	0.018763	-0.011201	-0.192975	-0.084755	-0.047212	0.034029	0.009850	-0.059969	0.003581	0.044811	-0.005832	0.002859	0.001467	-0.022702	-0.006480	0.272727
user_activity_var_2	-0.037443	-0.045376	0.017601	1.000000	-0.000680	0.127653	0.055613	-0.001538	-0.020844	-0.032335	0.137292	0.038266	0.087277	0.020758	0.354627	-0.000951	-0.000221	-0.048975	0.062567	-0.045957	0.163876
user_activity_var_3	-0.010616	-0.007585	-0.023360	-0.000680	1.000000	0.004403	0.022082	-0.053253	-0.007387	-0.021408	0.000885	0.014060	-0.029252	0.009858	0.005174	-0.005387	-0.001098	-0.007763	0.022377	-0.010457	0.232749
user_activity_var_4	-0.034883	-0.045739	0.018763	0.127653	0.004403	1.000000	0.063659	-0.018826	-0.004149	-0.037123	0.199956	0.038779	0.070903	0.032281	0.394706	-0.000281	0.005777	-0.069519	0.074122	-0.044503	0.188159
user_activity_var_5	-0.024671	-0.033279	-0.011201	0.055613	0.022082	0.063659	1.000000	-0.079282	-0.010042	-0.032518	0.074787	0.020800	0.024516	0.022356	0.164972	-0.001455	-0.003638	-0.039784	0.034846	-0.031928	0.313276
user_activity_var_6	0.030934	0.068066	-0.192975	-0.001538	-0.053253	-0.018826	-0.079282	1.000000	-0.167992	-0.060762	-0.015576	0.001631	-0.078368	0.000535	-0.010951	0.006291	-0.009501	0.005162	0.036197	0.052868	0.268465
user_activity_var_7	-0.072626	-0.047743	-0.084755	-0.020844	-0.007387	-0.004149	-0.010042	-0.167992	1.000000	-0.038432	-0.010835	0.003963	-0.060978	0.005201	-0.028428	-0.000706	-0.003123	-0.003366	0.042693	-0.069518	0.263123
user_activity_var_8	-0.018211	-0.042167	-0.047212	-0.032335	-0.021408	-0.037123	-0.032518	-0.060762	-0.038432	1.000000	-0.039844	-0.008283	-0.039283	-0.002724	-0.097355	-0.007034	0.005513	0.044930	-0.120139	-0.032145	0.173057
user_activity_var_9	-0.058469	-0.065208	0.034029	0.137292	0.000885	0.199956	0.074787	-0.015576	-0.010835	-0.039844	1.000000	0.038102	0.128124	0.031698	0.463947	-0.003112	0.001754	-0.078333	0.085601	-0.069012	0.225443
user_activity_var_10	-0.008395	-0.010986	0.009850	0.038266	0.014060	0.038779	0.020800	0.001631	0.003963	-0.008283	0.038102	1.000000	0.020856	-0.000405	0.084423	0.000100	0.001128	-0.017773	0.015645	-0.010699	0.056541
user_activity_var_11	-0.013608	-0.002274	-0.059969	0.087277	-0.029252	0.070903	0.024516	-0.078368	-0.060978	-0.039283	0.128124	0.020856	1.000000	0.018506	0.267995	0.004933	0.001384	-0.022170	0.061651	-0.009775	0.344016
user_activity_var_12	-0.007810	-0.009580	0.003581	0.020758	0.009858	0.032281	0.022356	0.000535	0.005201	-0.002724	0.031698	-0.000405	0.018506	1.000000	0.067967	-0.004669	0.001264	-0.023939	0.009569	-0.009642	0.051778
buy	-0.087202	-0.080064	0.044811	0.354627	0.005174	0.394706	0.164972	-0.010951	-0.028428	-0.097355	0.463947	0.084423	0.267995	0.067967	1.000000	-0.012054	-0.046612	-0.151513	0.177854	-0.094955	0.320703
signup_day	0.001964	0.005235	-0.005832	-0.000951	-0.005387	-0.000281	-0.001455	0.006291	-0.000706	-0.007034	-0.003112	0.000100	0.004933	-0.004669	-0.012054	1.000000	0.015576	-0.019801	0.000147	0.003802	-0.002552
signup_month	-0.012685	-0.020733	0.002859	-0.000221	-0.001098	0.005777	-0.003638	-0.009501	-0.003123	0.005513	0.001754	0.001128	0.001384	0.001264	-0.046612	0.015576	1.000000	-0.344649	0.019015	-0.018182	0.005317
signup_year	0.083854	0.140935	0.001467	-0.048975	-0.007763	-0.069519	-0.039784	0.005162	-0.003366	0.044930	-0.078333	-0.017773	-0.022170	-0.023939	-0.151513	-0.019801	-0.344649	1.000000	-0.471780	0.122078	-0.234527
user_activity_var_0	-0.155888	-0.183422	-0.022702	0.062567	0.022377	0.074122	0.034846	0.036197	0.042693	-0.120139	0.085601	0.015645	0.061651	0.009569	0.177854	0.000147	0.019015	-0.471780	1.000000	-0.188658	0.491383
total_campaigns	0.917114	0.853541	-0.006480	-0.045957	-0.010457	-0.044503	-0.031928	0.052868	-0.069518	-0.032145	-0.069012	-0.010699	-0.009775	-0.009642	-0.094955	0.003802	-0.018182	0.122078	-0.188658	1.000000	-0.130884
total_activity	-0.108349	-0.126989	0.272727	0.163876	0.232749	0.188159	0.313276	0.268465	0.263123	0.173057	0.225443	0.056541	0.344016	0.051778	0.320703	-0.002552	0.005317	-0.234527	0.491383	-0.130884	1.000000

3) Data Pre-Processing

The dataset was divided into train and test sets. The data was scaled using MinMaxScaler

The features were tested again with statistical methods to identify whether they are describing correctly or not

Chisquare method was adopted to test whether the features correctly describe. At 95% confidence the features with p values less than 0.05 were not selected for the final model

The final model contains 15 columns

4) Model Selection

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

- RandomForest
- XGBoost
- LightGBM
- CatBoost

CatBoostClassifier was later dropped as it works better with categorical varibales and since our data doesn't contain any categorical variables

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

XGBoost Classifier gave better results with the following parameters

- max_depth: 6
- learning_rate: 0.01
- colsample_bytree: 0.08
- n_estimators: 4000
- subsample: 1
- gamma: 0

5) **Results Interpretation:**

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

Results:

- Accuracy_score: 0.97
- F1_score: 0.71

CONCLUSION

The model gave a score of 0.76 in Analytics Vidya leaderboard and was in 6^{th} position on public data at the time of submitting this report