AV_job-a-thon-june-2022-Approach

Machine learning model to predict if the user would buy the product in the next 3 months or not based on the user's past activities and user-level information.

1.Exploratory Data Analysis

- Pandas, seaborn, matplotlib libraries are used in Exploratory data analysis.

2.Data Pre-Processing

- Missing value indicator created for the column **products purchased**
- Date feature engineering:
  - Convert to date-time format
  - Extract day from the date
  - Extract the day name from the date
  - Extract the day number from the date
  - Extract month number from the date
  - Extract the month name from the date
  - Extract the quarter of the year
  - Extract week of the year from date
  - Extract year
  - Extract the day of the month
  - Extract day of the year
  - Create weekday column
  - Create weekend column
  - Create month start
  - Create month end
  - Create quarter start
  - Create quarter end
  - Create year start
  - Create year end

- Products purchased column’s 20911 missing values are filled by using group by median value.
- Created user count feature by using group by row count.

- Created a group by numerical summary(min, max, median, mean ,count) feature for product purchased column

- Created a new feature that is the whether the lead is created after the signup. It has so many missing values due to the signup column doesn't have values for 15113 rows. So created a missing value indicator for this column.The missing values are filled with zero.

- Create a date difference between the lead created date and the signup date. It has so many missing values due to the signup column doesn't have values for 15113 rows. So created a missing value indicator for this column. The missing values are filled with zero.

3.Model

- After pre-processing finally 42 columns are selected for the classification model.
- Selected columns are,
    - 0_campaign_var_1
    - 1_campaign_var_2
    - 2_products_purchased
    - 3_user_activity_var_1
    - 4_user_activity_var_2
    - 5_user_activity_var_3
    - 6_user_activity_var_4
    - 7_user_activity_var_5
    - 8_user_activity_var_6
    - 9_user_activity_var_7
    - 10_user_activity_var_8
    - 11_user_activity_var_9

- 12_user_activity_var_10
- 13_user_activity_var_11
- 14_user_activity_var_12
- 15_products_purchased_null
- 16_day
- 17_day_number
- 18_month_number
- 19_year_quarter
- 20_week_of_year
- 21_year
- 22_dayofmonth
- 23_dayofyear
- 24_weekday
- 25_weekend
- 26_month_start
- 27_month_end
- 28_quarter_start
- 29_quarter_end
- 30_year_start
- 31_year_end
- 32_user_count
- 33_grp_mean
- 34_grp_median
- 35_grp_max
- 36_grp_min
- 37_grp_count
- 38_lead_after_signup
- 39_lead_after_signup_null
- 40_date_diff
- 41_date_diff_null

- Compared multiple classifiers using pycaret’s compare_models function.

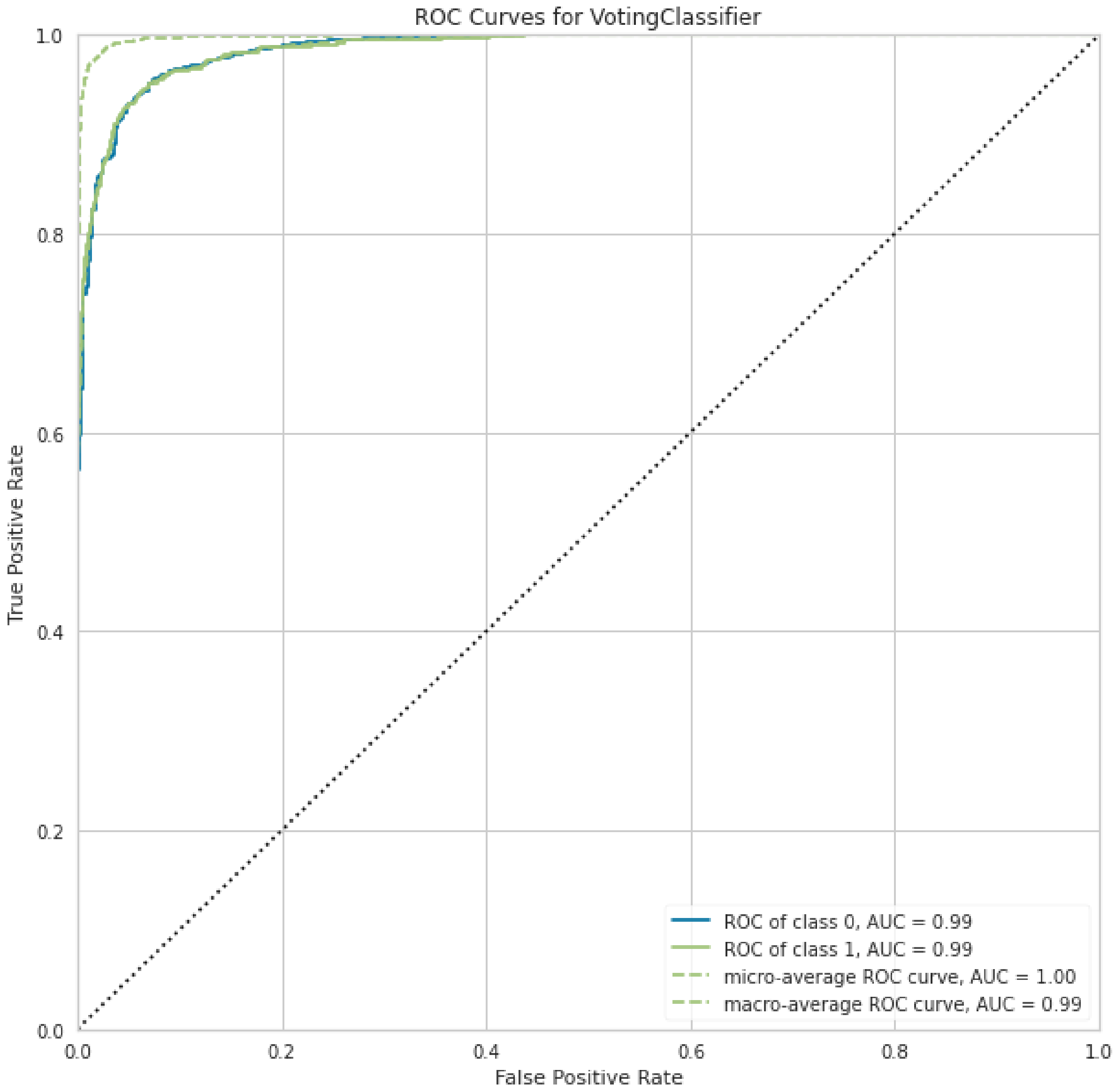
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.9754	0.9484	0.5626	0.9284	0.7005	0.6885	0.7121	8.7260
xgboost	Extreme Gradient Boosting	0.9747	0.9413	0.5733	0.8937	0.6982	0.6857	0.7043	0.5160
gbc	Gradient Boosting Classifier	0.9754	0.9472	0.5540	0.9408	0.6972	0.6853	0.7116	6.1360
ada	Ada Boost Classifier	0.9756	0.9464	0.5497	0.9518	0.6968	0.6851	0.7133	1.6220
lightgbm	Light Gradient Boosting Machine	0.9749	0.9469	0.5590	0.9162	0.6941	0.6818	0.7046	0.7820
lda	Linear Discriminant Analysis	0.9750	0.9369	0.5418	0.9452	0.6887	0.6767	0.7053	0.7080
et	Extra Trees Classifier	0.9744	0.9200	0.5411	0.9259	0.6828	0.6704	0.6969	3.8460
lr	Logistic Regression	0.9731	0.9439	0.5175	0.9212	0.6624	0.6495	0.6791	0.5660
rf	Random Forest Classifier	0.9710	0.9294	0.4546	0.9536	0.6153	0.6023	0.6474	1.2020
nb	Naive Bayes	0.9536	0.9267	0.6562	0.5399	0.5912	0.5669	0.5706	0.0680
dt	Decision Tree Classifier	0.9556	0.7950	0.6162	0.5597	0.5864	0.5630	0.5638	0.3300
ridge	Ridge Classifier	0.9443	0.0000	0.4260	0.5317	0.4539	0.4260	0.4386	0.0940
svm	SVM - Linear Kernel	0.4833	0.0000	0.9164	0.0843	0.1543	0.0671	0.1676	6.2780

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
qda	Quadratic Discriminant Analysis	0.4545	0.5871	0.7348	0.0697	0.1258	0.0363	0.0819	0.4280
knn	K Neighbors Classifier	0.9453	0.6500	0.0214	0.1901	0.0384	0.0283	0.0478	0.4960
dummy	Dummy Classifier	0.9490	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0320

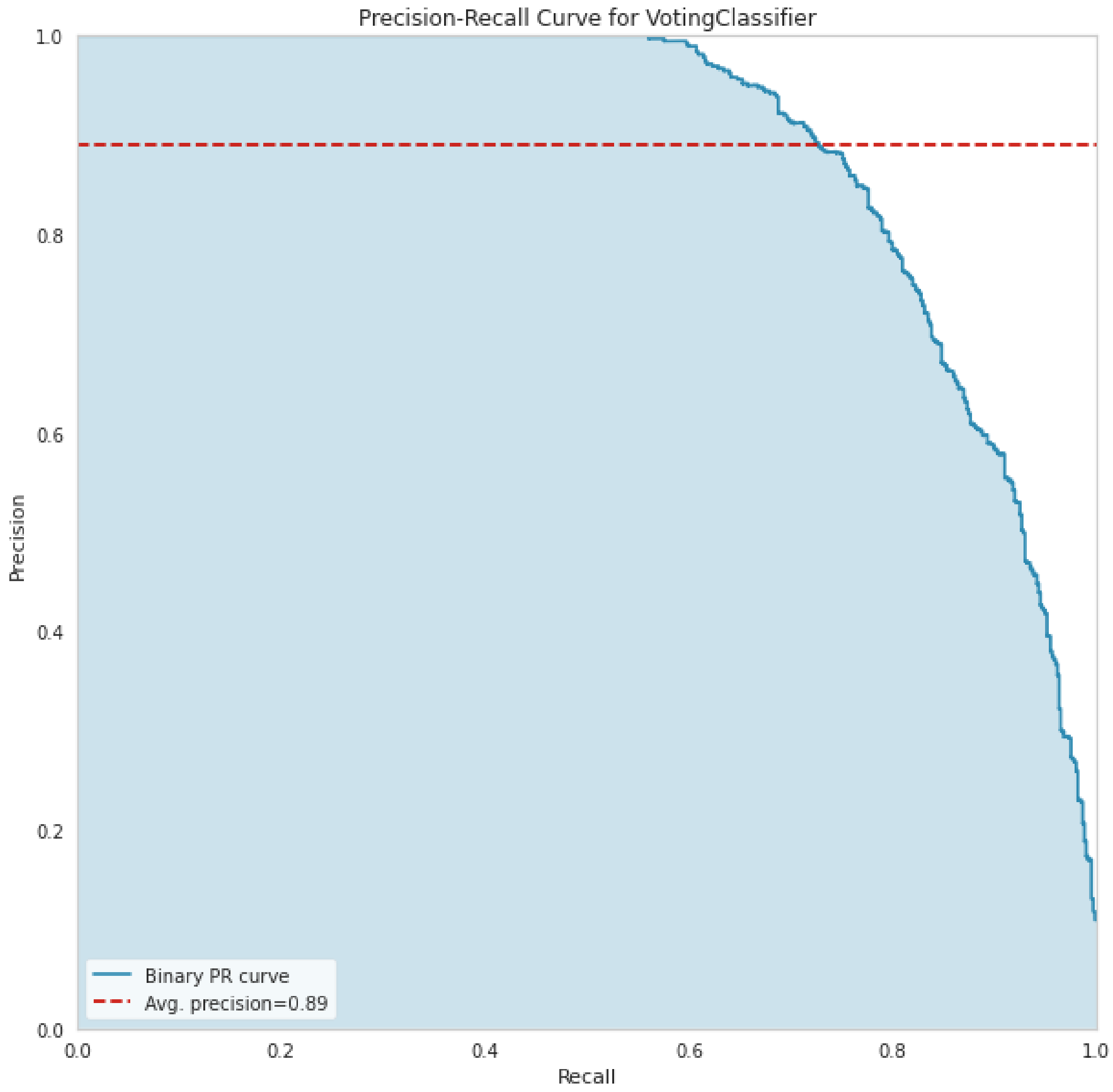
- Then took the top 3 models based on the f1 score then blend the model by using pycaret blend_models function.

Fold	Accuracy	AUC	Recall	Prec	F1	Kappa	MCC
0	0.9765	0.9532	0.5607	0.9632	0.7088	0.6974	0.7253
1	0.9761	0.9491	0.5571	0.9571	0.7043	0.6927	0.7204
2	0.9761	0.9553	0.5914	0.9066	0.7158	0.7039	0.7215
3	0.9750	0.9504	0.5607	0.9181	0.6962	0.6840	0.7067
4	0.9748	0.9382	0.5464	0.9329	0.6892	0.6770	0.7034
Mean	0.9757	0.9492	0.5633	0.9356	0.7029	0.6910	0.7155
Std	0.0007	0.0059	0.0150	0.0218	0.0093	0.0096	0.0087

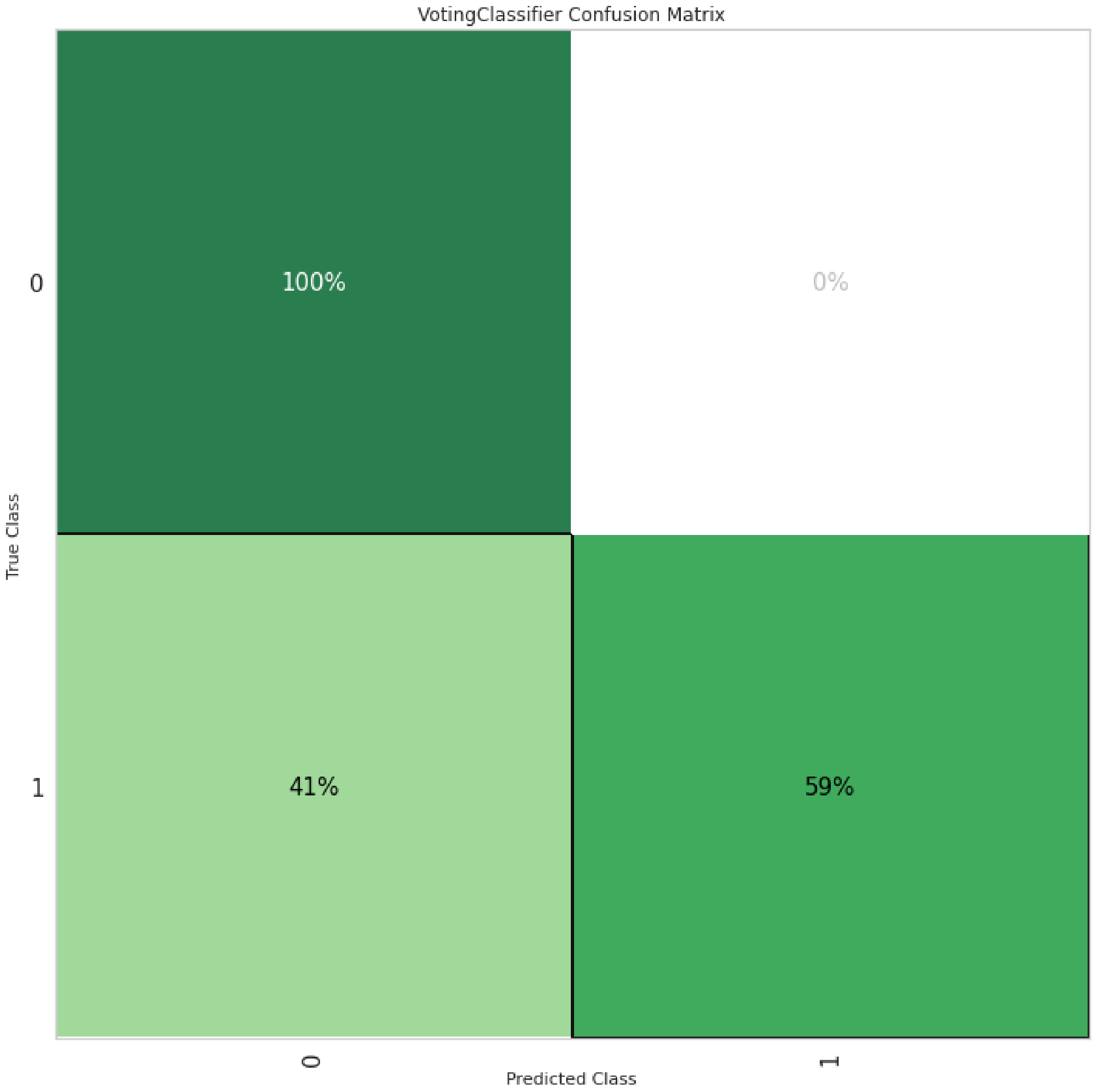
- ROC curve


ROC Curves for VotingClassifier
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC of class 0, AUC = 0.99
ROC of class 1, AUC = 0.99
micro-average ROC curve, AUC = 1.00
macro-average ROC curve, AUC = 0.99

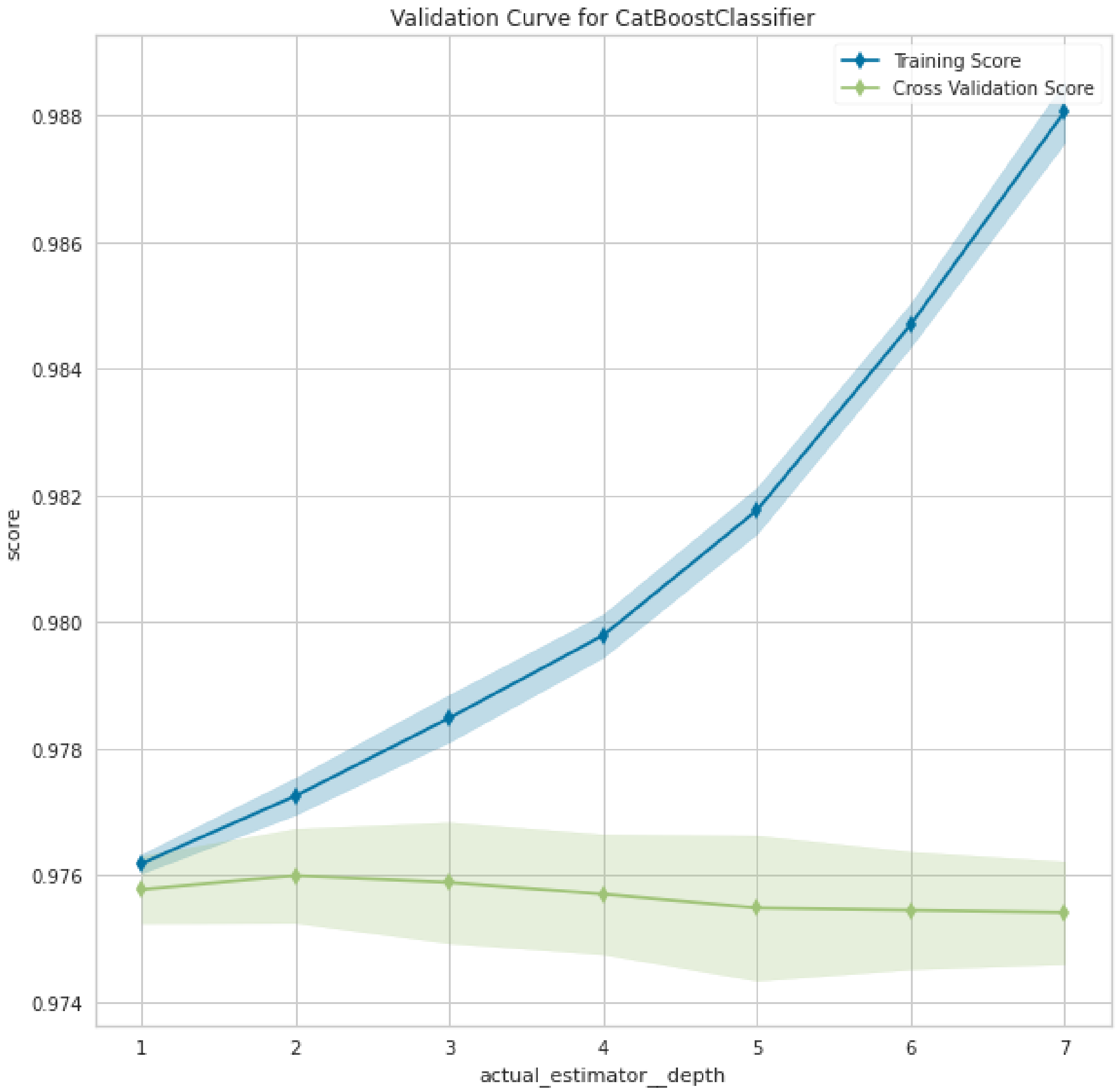
- Precision & Recall curve



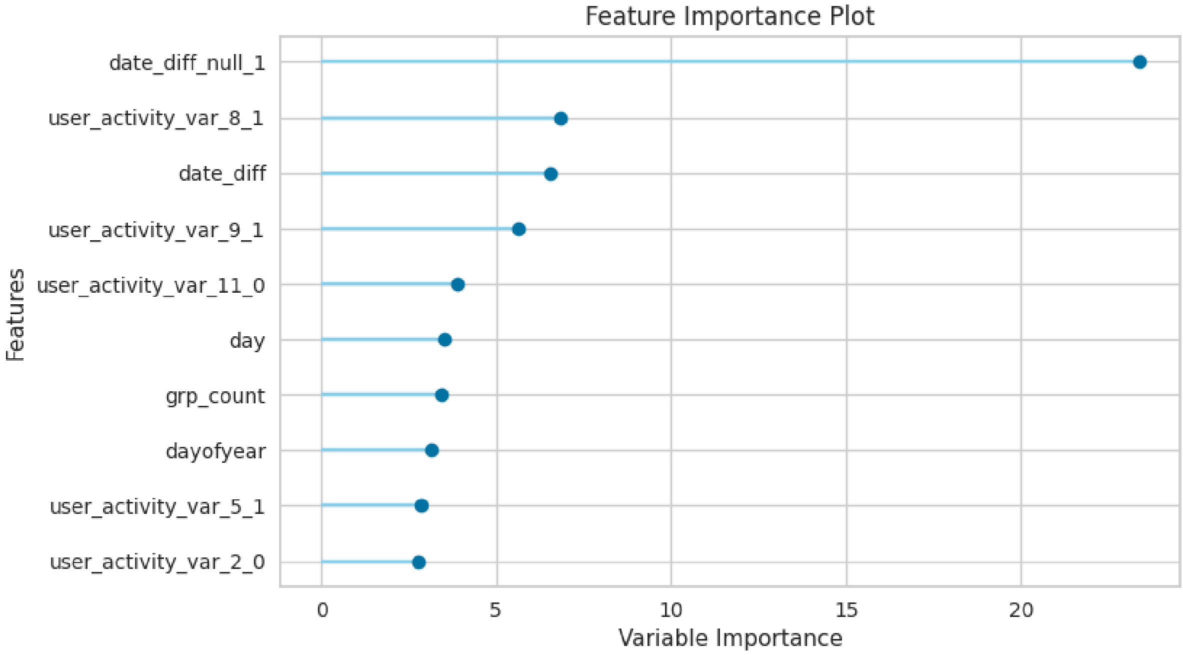
- Confusion Matrix



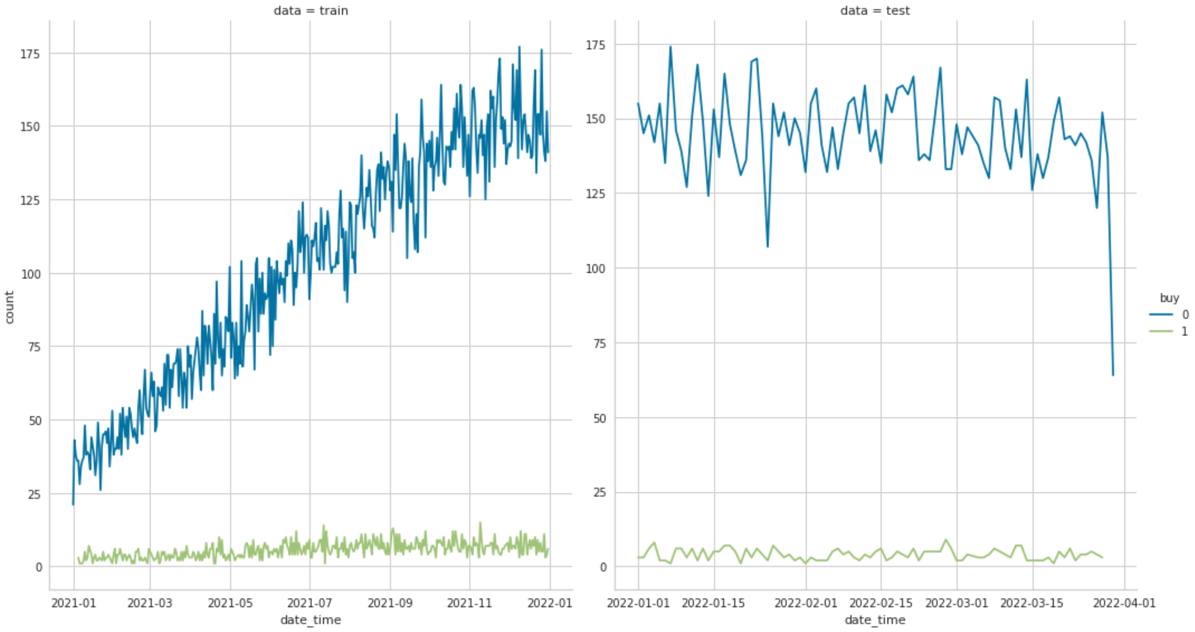
- Catboost validation curve



- Catboost model feature Importance



- Prediction plot



- Final public leader board score is 0.733727810650888