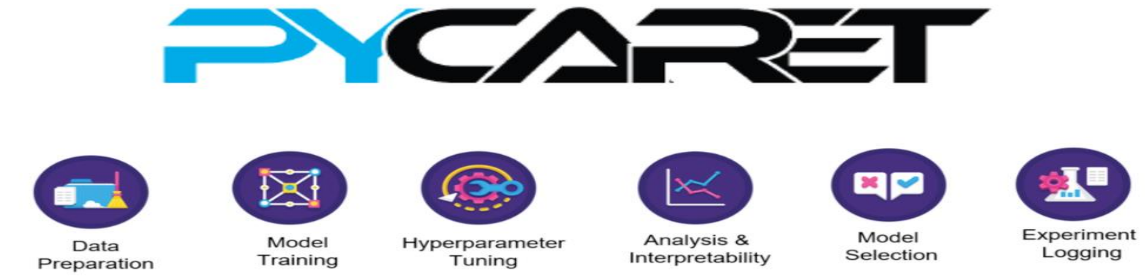
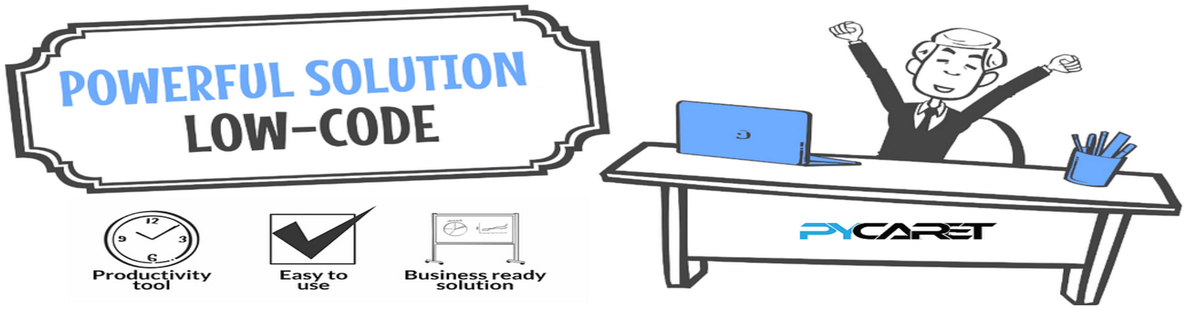
E-Commerce Customer Churn Prediction With PyCaret





Churn refers to the rate or number of customers who leave a company or service within a certain period of time. The term is often used to describe customers who stop being customers or no longer purchase products with a company, which can ultimately have a negative impact on the company's revenue.

Low churn rates are generally considered a sign of high customer satisfaction, while high churn may indicate a problem in the product, service, or customer relationship.

Stakeholder

- ***Marketing & Sales Department***
- ***Finance Department***

Problem Statement

In the competitive e-commerce business environment, the main challenge faced by companies is how to recognize and understand the behaviour of customers who have the potential to churn. The main goal is to identify churn customers accurately so that companies can design and implement appropriate and effective promos to prevent churn.

Goal

Stakeholders can find out which customers churn and do not churn so that they can approach customers and offer promos in a targeted manner.

Analytic Approach

We will analyze the data patterns that determine churn and non-churn customers and then build a classification model that will help stakeholders predict churn and non-churn customers.

Metric Evaluation

0 : Not Churn (Negative)
1 : Churn (Positive)

```
|                      |    Not Churn-Pred    |    Churn-Pred     |
| ---------------------| -------------------- | ----------------- |
|     Not Churn-Act    |         TN           |        FP         |
|     Churn-Act        |         FN           |        TP         |
```

Type Error 1 | False Positive

Interpretation : customers who in reality do not churn, but are predicted as churn customers

Consequences : the company incurs unnecessary costs

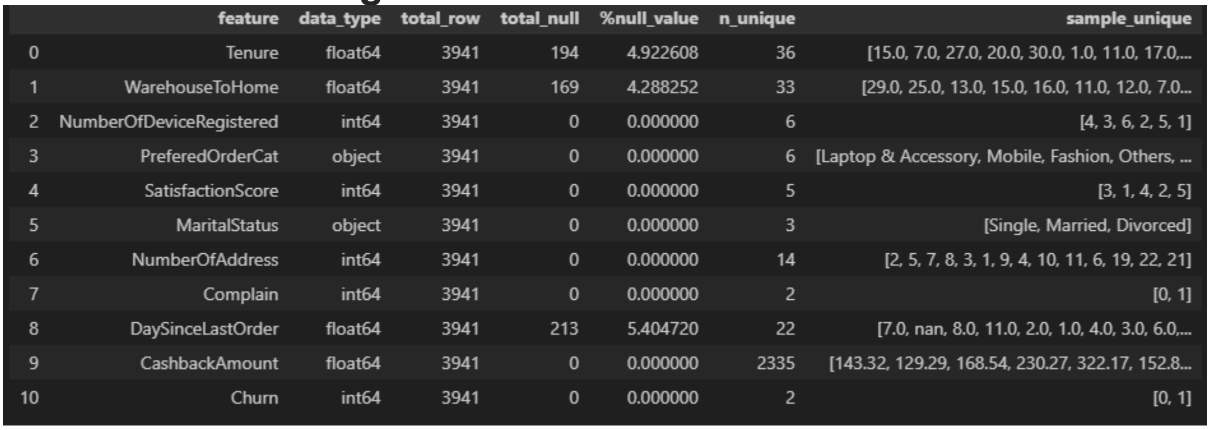
Type Error 2 |False Negative

Interpretation : customers who in fact churn, but are predicted as not churn customers

Consequences : The company will lose potential customers who could have become loyal customers.

Based on the consequences above. The risk of losing customers is much higher than spending money on wrong customers. if we lose loyal customers then we need to make a lot of advertisements and promos to attract new customers. then what we will do is focus on False Negative we will focus on recall and f2-score.

Data Understanding



	feature	data_type	total_row	total_null	%null_value	n_unique	sample_unique
0	Tenure	float64	3941	194	4.922608	36	[15.0, 7.0, 27.0, 20.0, 30.0, 1.0, 11.0, 17.0,...
1	WarehouseToHome	float64	3941	169	4.288252	33	[29.0, 25.0, 13.0, 15.0, 16.0, 11.0, 12.0, 7.0...
2	NumberOfDeviceRegistered	int64	3941	0	0.000000	6	[4, 3, 6, 2, 5, 1]
3	PreferedOrderCat	object	3941	0	0.000000	6	[Laptop & Accessory, Mobile, Fashion, Others, ...
4	SatisfactionScore	int64	3941	0	0.000000	5	[3, 1, 4, 2, 5]
5	MaritalStatus	object	3941	0	0.000000	3	[Single, Married, Divorced]
6	NumberOfAddress	int64	3941	0	0.000000	14	[2, 5, 7, 8, 3, 1, 9, 4, 10, 11, 6, 19, 22, 21]
7	Complain	int64	3941	0	0.000000	2	[0, 1]
8	DaySinceLastOrder	float64	3941	213	5.404720	22	[7.0, nan, 8.0, 11.0, 2.0, 1.0, 4.0, 3.0, 6.0,...
9	CashbackAmount	float64	3941	0	0.000000	2335	[143.32, 129.29, 168.54, 230.27, 322.17, 152.8...
10	Churn	int64	3941	0	0.000000	2	[0, 1]

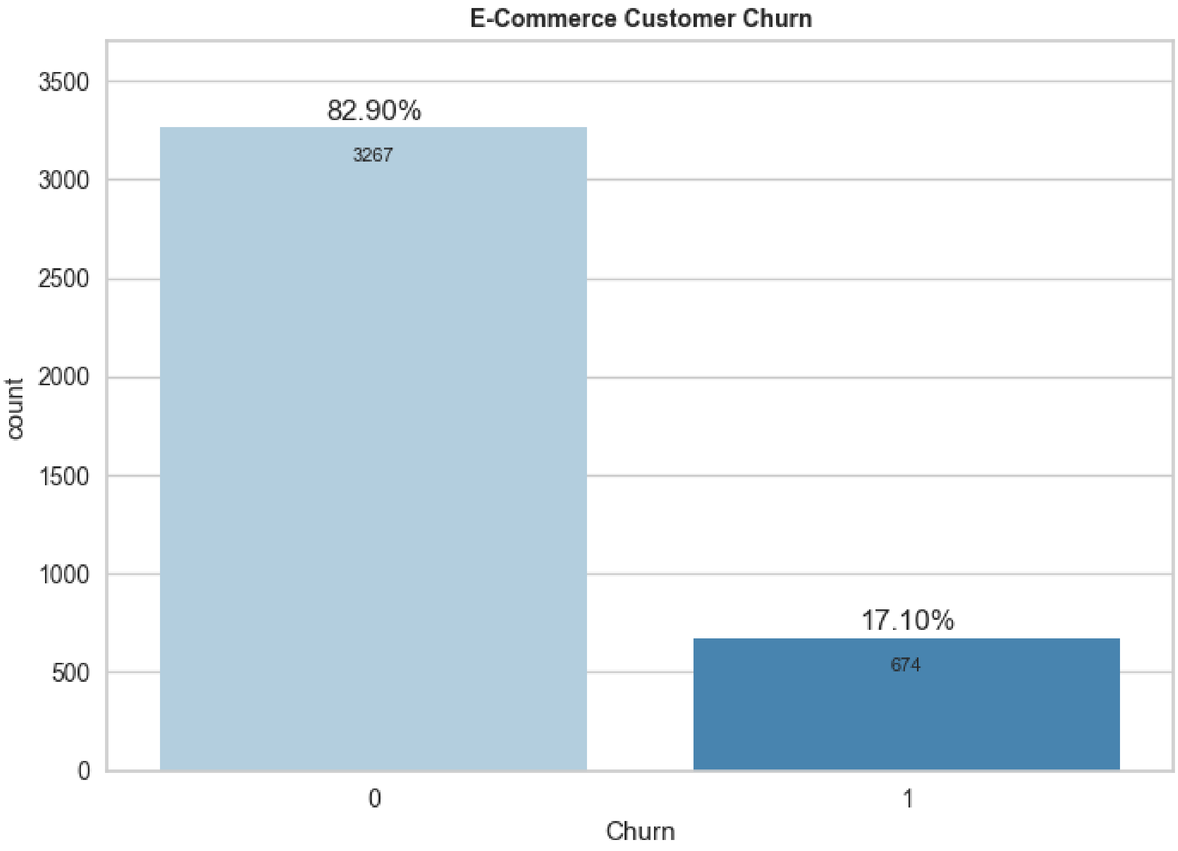
In Dataset here are 3941 rows and 11 columns and 14.61% missing value

Numerical variable :

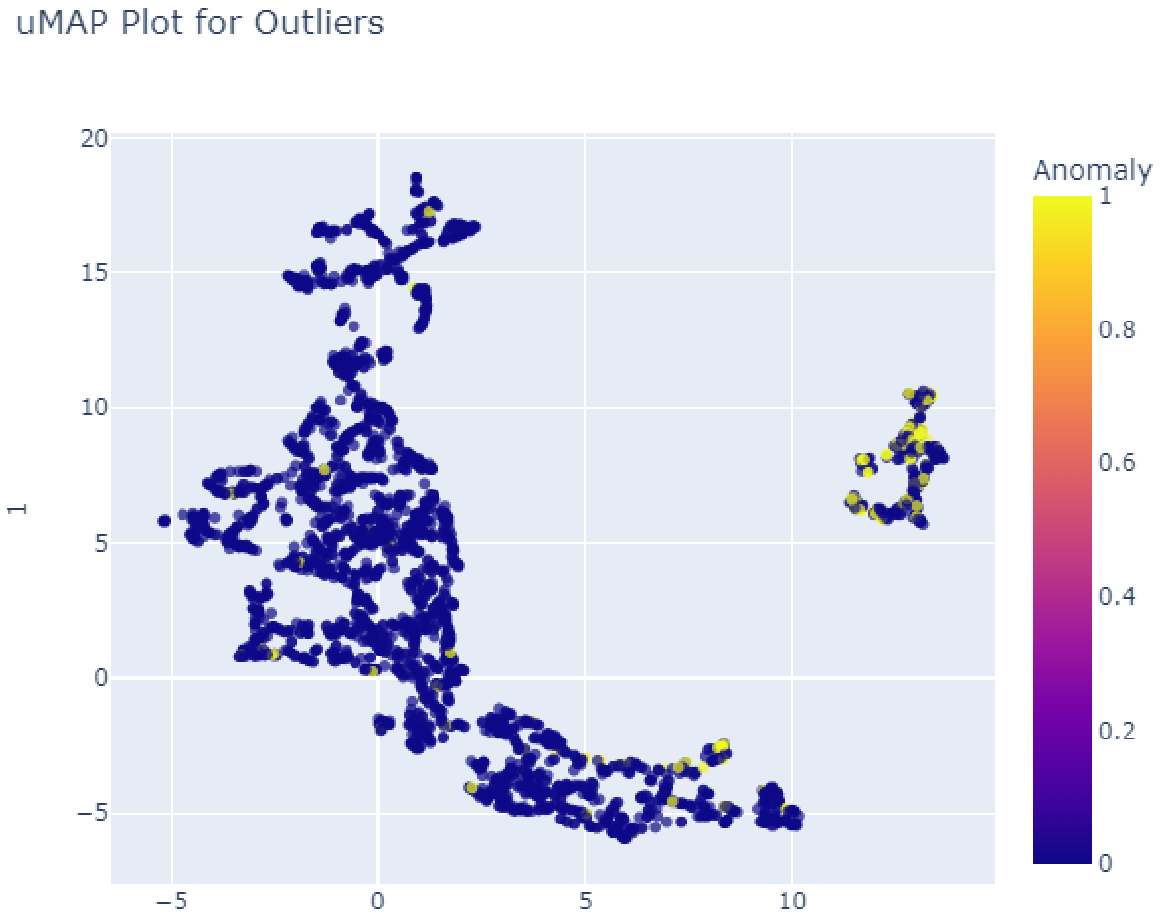
- Discrete : NumberOfDeviceRegistered, NumberOfAddress
- Continue: Tenure, WarehouseToHome, DaySinceLastOrder, CashbackAmount

Categorical variable :

- Nominal : PreferedOrderCat, MaritalStatus, Churn
- Ordinal : SatisfactionScore, Complain

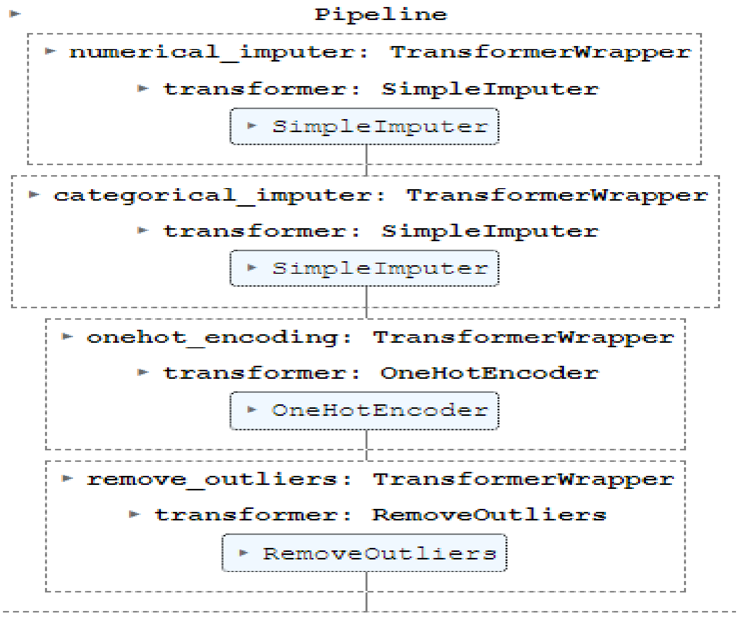


- Number of Customer who churn 674(17.10%)
- Number of Customer who not churn 3267(82.90%)



There are 5% of the total data detected as outliers, this value can still be considered reasonable.

There is a duplicate data of 672 that we will drop. for missing values we will handle with iterativeimputer .



```
▸ Pipeline
  ▸ numerical_imputer: TransformerWrapper
      ▸ transformer: SimpleImputer
          ▸ SimpleImputer
  ▸ categorical_imputer: TransformerWrapper
      ▸ transformer: SimpleImputer
          ▸ SimpleImputer
  ▸ onehot_encoding: TransformerWrapper
      ▸ transformer: OneHotEncoder
          ▸ OneHotEncoder
  ▸ remove_outliers: TransformerWrapper
      ▸ transformer: RemoveOutliers
          ▸ RemoveOutliers
```

this is the pipeline flow if there is new data coming in it will be handled according to this flow.