Startup Campus Kampus Merdeka

Class C

# Customer Churn Prediction Using Machine Learning Models

Mentor : Muhamad Anwar Sanusi

Facil : Ari Mulyadi

Group : Jerman

# Table of contents

# 01.

# Business & Data Understanding

# Business Understanding

## Background

At this time online shopping activities are very important for the community, which means it is important for e-commerce companies to optimize marketing strategies and increase effectiveness in terms of convenience. Understanding customer needs to direct more effective marketing plans, more brilliant product designs, convenience and safety when shopping online, and pay attention to customer satisfaction.

## Objective

- Clear understanding of customer e-commerce
- Help e-commerce to predict the chances of customers leaving the service

## Problem Scope

Are customers who choose to stop shopping more or less? This case study was created to determine the percentage of customers who stopped shopping, this allows e-commerce companies to develop a deeper understanding of customer desires and improve service.

# About General Dataset

## Clickstream

6 Features

12.833.602 Rows

## Customer

15 Features

100.000 Rows

## Product

10 Features

44424 Rows

## Transaction

14 Features

852.584 Rows

# About Merged Dataset

## 15 Categorical Features

- Session id
- event name
- event time
- event id
- traffic source
- created_at
- payment_method
- payment_status
- promo_code
- shipment_date_limit
- gender
- birthdate
- device type
- device version
- home location
- first join date

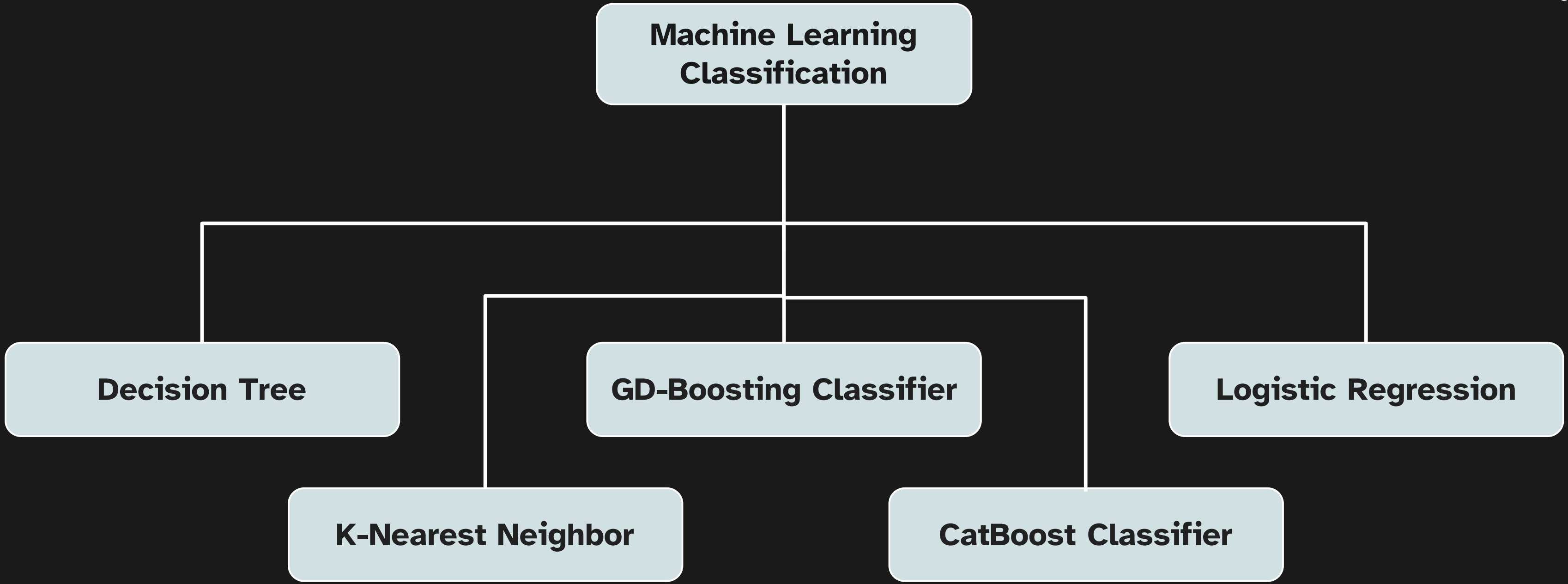
## 4 Numerical Features

- customer id
- shipment_fee
- total_amount
- promo_amount

# 02.

# Analytical Approach

# Modeling Algorithm



# Time Window Customer Churn

The timeframe for determining whether a customer churns or not is based on a customer who has not made any transactions in the last 30 days.

Last time period :
31-07-2022

Time Window

01-07-2022 - 31-07-2022

# 03.

# Data Preprocessing

# Features Engineering

**Total Promo Amount**

The sum of all the total promos that each customer gets during the transaction period

**Number Of Promo**

The number of promos that customers get from the first transaction to the last transaction

**Transaction Failed**

The number of transactions the customer canceled

**Number Of Transaction**

The number of transactions the customer has made.

**Total Shipment Fee**

The total cost of shipping goods that the customer has received during the transaction period

**Delivery Time**

The length of time it takes to deliver the goods to the customer.

# Data cleaning

- Handling Missing Value
  - None
    - promo_code
  - 0
    - number_of_promo
    - transaction_failed
    - number_of_transaction

```
[ ]  1 # check for missing value
     2 mergedata.isnull().sum()
```

```
session_id               0
event_name               0
event_time               0
traffic_source           0
created_at               0
customer_id              0
payment_method           0
payment_status           0
promo_amount             0
promo_code               0
shipment_fee             0
shipment_date_limit      0
total_amount             0
gender                   0
birthdate                0
device_type              0
device_version           0
home_location            0
first_join_date          0
total_promo_amount       0
number_of_promo          0
transaction_failed       0
number_of_transaction    0
total_shipment_fee       0
dtype: int64
```

```
1 # check for duplicate data
2 mergedata.duplicated().sum()
```

```
0
```

**duplicated** function to check for duplicate data. After checking, the data does not contain duplicate data (output 0).

# Data labeling

Library

Method

Data Labeling

**Import Library**
*Sci-kit LabelEncoder*

**Label Encoding**
*LabelEncoder()*

Event_name

Gender

Device Version

Home_Location

Number_Of_Promo

Transaction_Failed

Number_Of_Transaction

We use label encoder to convert non-numeric data type to numeric data type to build a better accuracy model

# Standardization

**Standar Scaling**
*StandarScaler()*

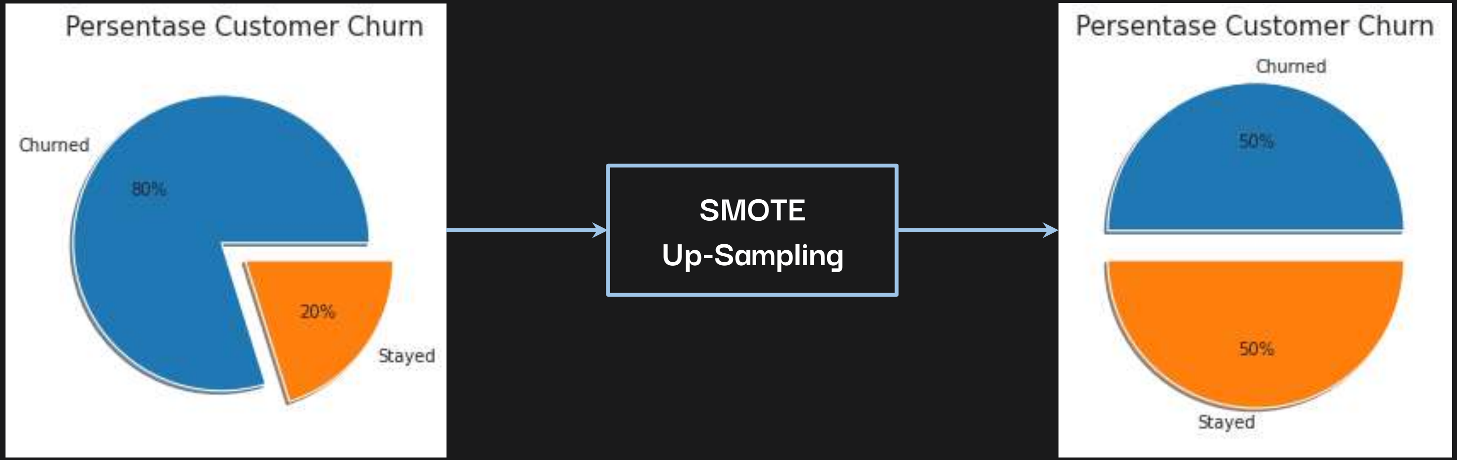
- Total Promo Amount
- Number Of Transaction
- Number Of Promo
- Transaction Failed

Method

Numerical Variable X

We use **StandardScaler()** function to standardize all our data in order to make a better accuracy model

# Handling imbalance

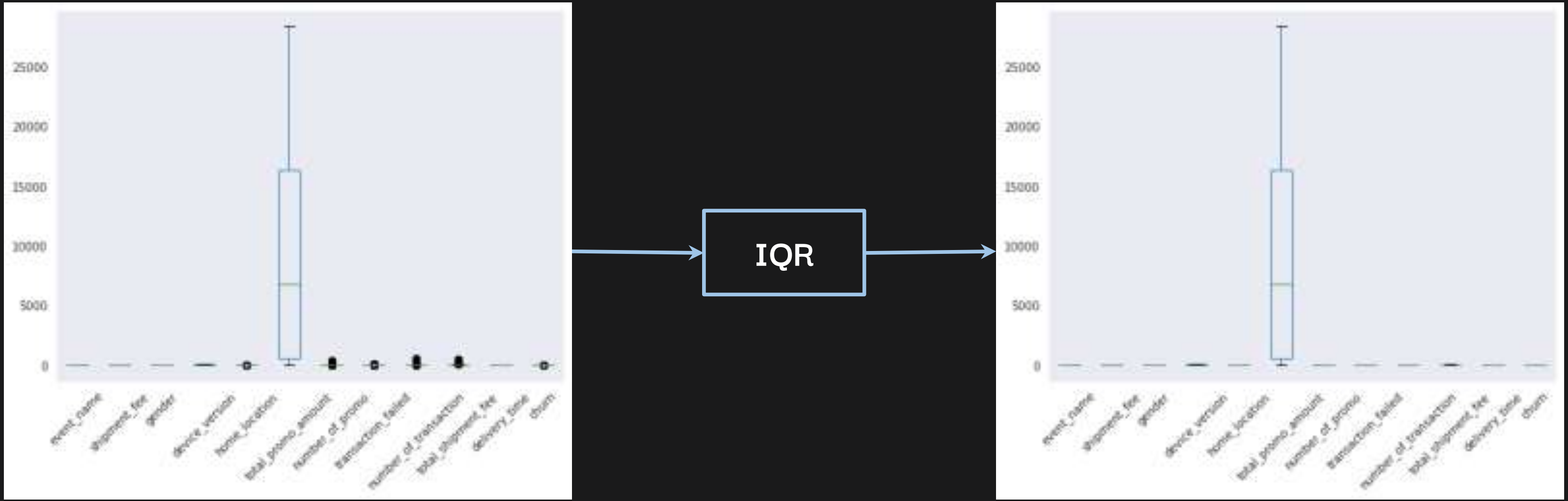



Before handling Imbalance Data

After handling Imbalance Data

We use SMOTE to handle imbalance data in our target variable because by doing that we have an advantage of no loss information and also mitigate over fitting caused by upsampling. Balancing the imbalance data is very important in ML in order to achieve the right accuracy

# Handling outliers




One of the most important steps as part of data preprocessing is detecting and treating the outliers as they can negatively affect the statistical analysis and the training process of machine learning algorithm resulting in lower accuracy

# Feature Selection

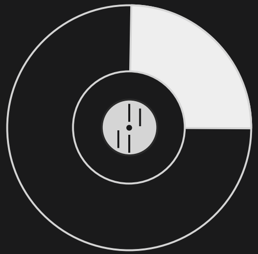
The features that correlate well with target feature:

| | Features | Score |
|---|---|---|
| 5 | total_promo_amount | 5.921728e+07 |
| 8 | number_of_transaction | 4.108033e+05 |
| 6 | number_of_promo | 1.302990e+05 |
| 7 | transaction_failed | 1.741387e+04 |
| 0 | event_name | 1.353473e+03 |
| 4 | home_location | 2.914299e+00 |
| 9 | delivery_time | 2.742382e+00 |
| 1 | shipment_fee | 9.764781e-01 |
| 3 | device_version | 1.794584e-01 |
| 2 | gender | 1.598714e-02 |

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
```

- We use feature selection to know the features that correlate well with our target feature (churn)
- We use the Chi-Square method (chi2)
- From the feature selection results obtained, we take the 8 best features to be used for modeling.

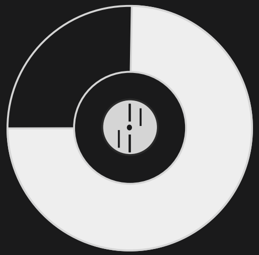
# Data Spliting



## 20%

**Data Testing**

40.563 test data



## 80%

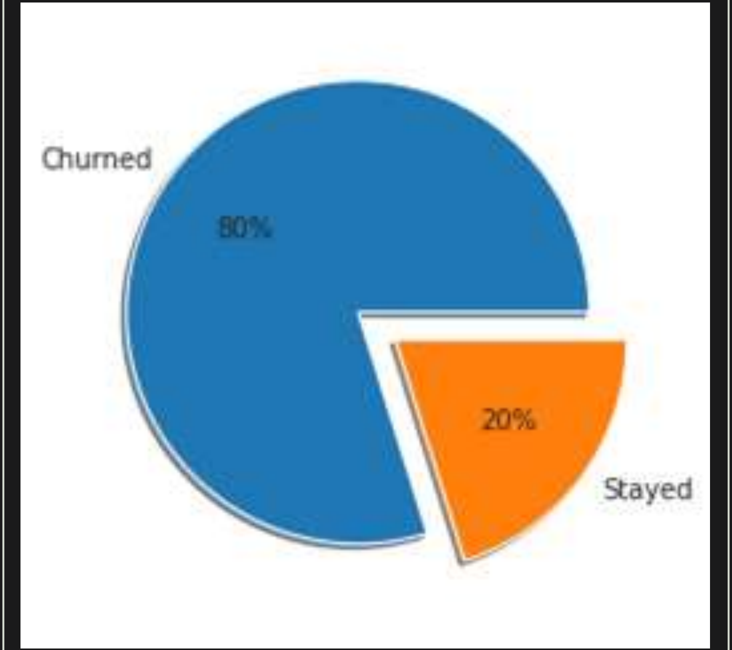
**Data Training**

10.141 train data

# 04.

# Data visualization & EDA

# Data visualization & EDA

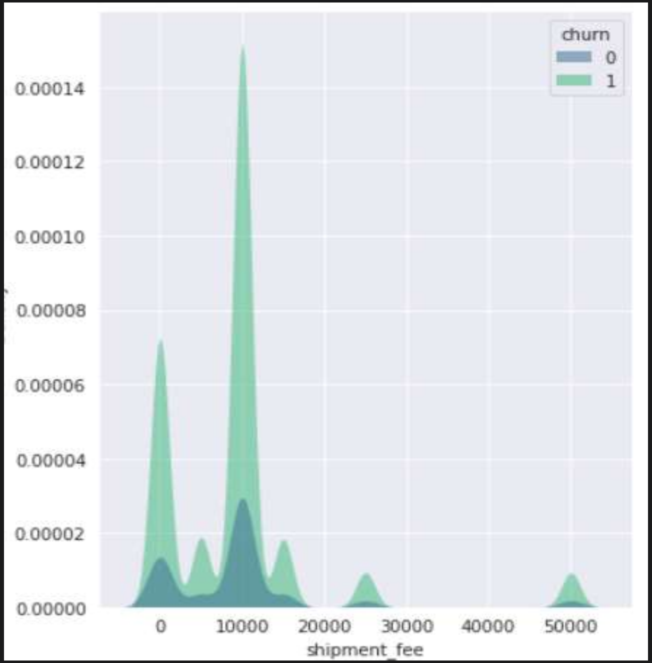



80% Churned
40.481 Customers

20% Stayed
10.223 Customers

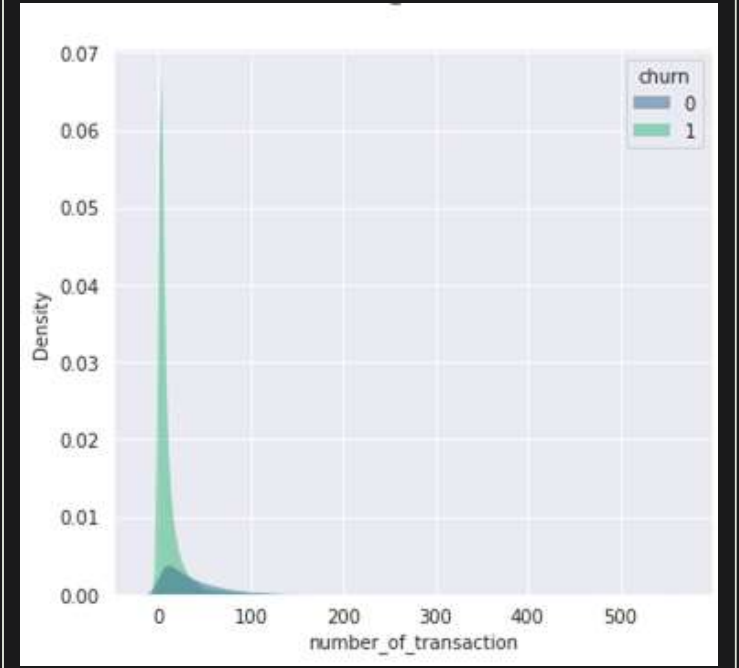
Based on the chart, it was found that most of the data distribution did not Churn, with details of CHURNED as much as 40,481 (80%) and STAYED as much as 10,223 (20%).

# Data visualization & EDA



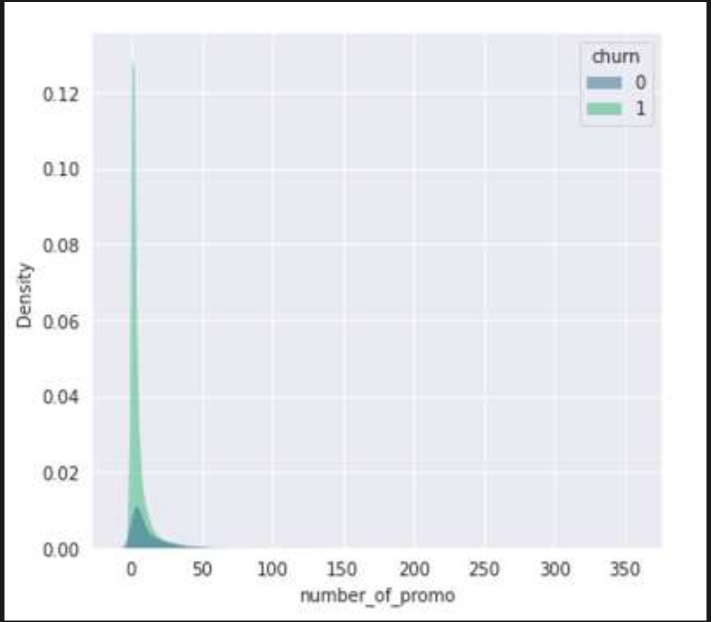

Seen from the chart on the side shows the churn rate has decreased if shipping costs can be reduced or reasonable shipping costs are provided so that it can reduce the churn rate

# Data visualization & EDA



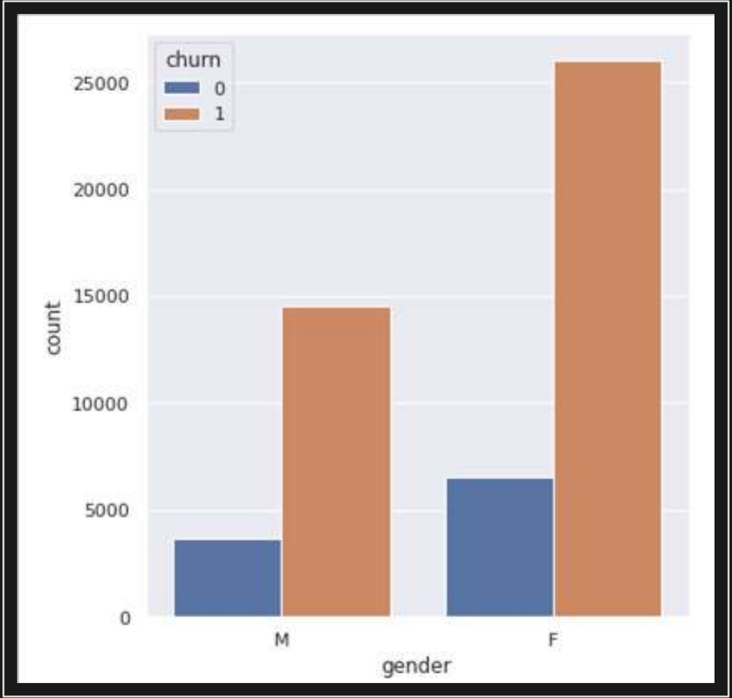

As we can see from the graph, the high churn rate is caused by too few customer transactions. Customers who rarely make transactions mean they are passive customers, therefore they are more likely to have a higher churn rate. Even on the graph, a value of 0 indicates a customer who has never made a transaction. Those customers are definitely churn.

# Data visualization & EDA



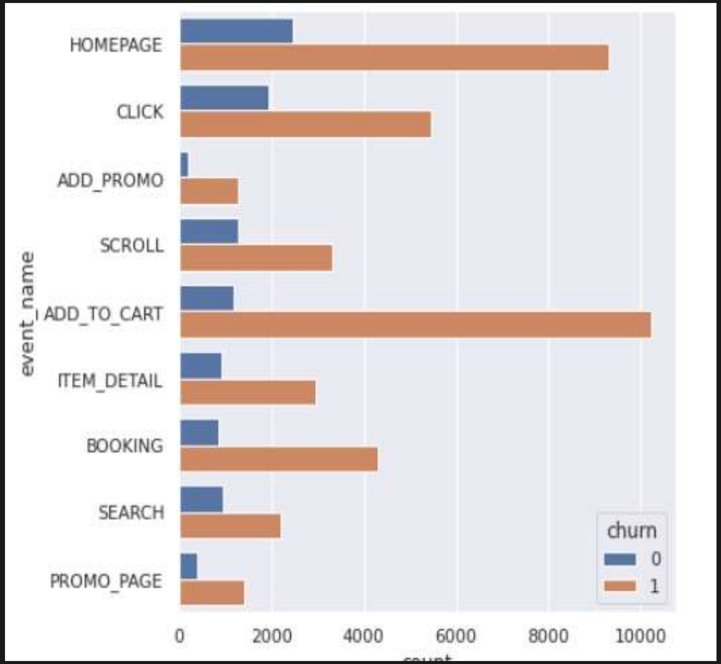

As we can see in the graph, the high churn rate is due to the small amount of promos that customers get when transacting. Customers who get few promos and never get promos during transactions will increase customer churn.

# Data visualization & EDA



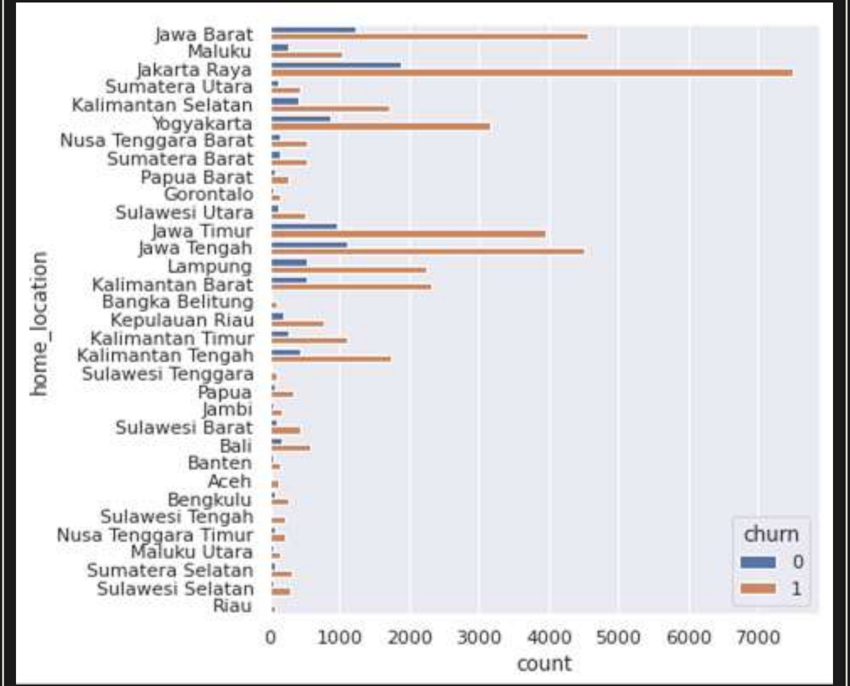

The graph shows that the sex with the most churn is female, reaching more than 25,000 people, and Male is churn close to 15,000 people.

# Data visualization & EDA



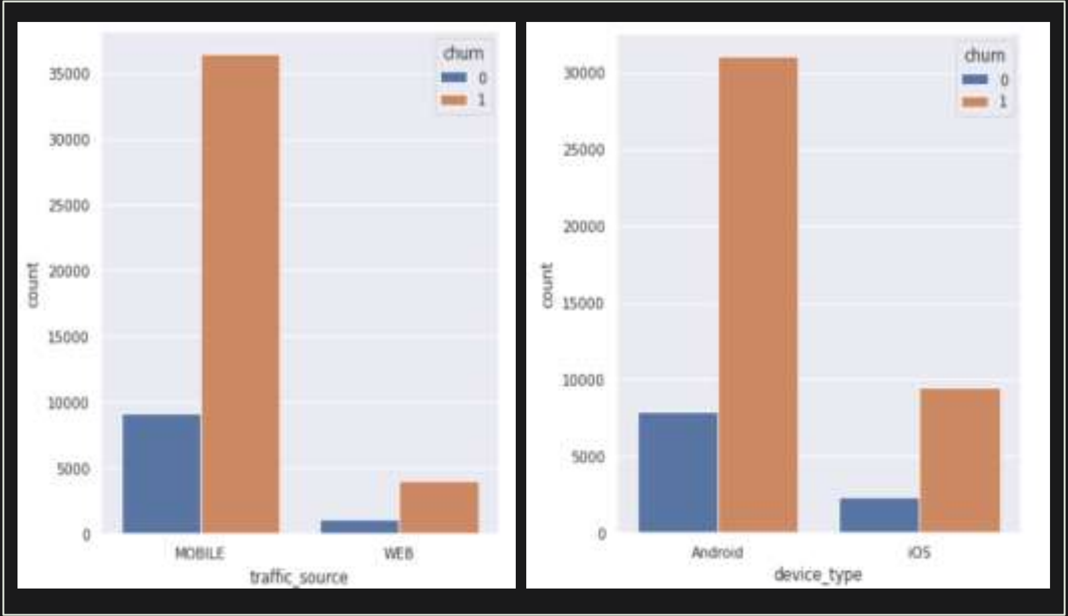

as we can see in the chart on the side that a high churn rate occurs when a customer uses the add to cart feature, this indicates a customer adding goods to the cart is often the end of the customer transacting and on the chart also the customer often ends up only on the homepage without continuing the transaction

# Data visualization & EDA



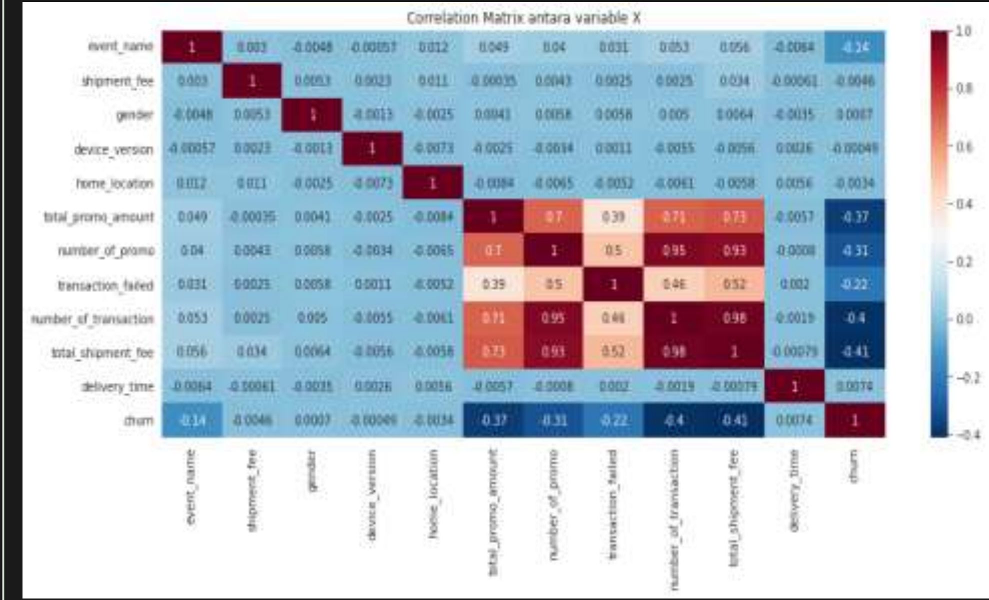

In the graph below, the majority of churn customers are from Jakarta Raya reaching more than 7,000 customers, and the fewest churn customers are from Riau.

# Data visualization & EDA




as we can see in the chart on the side that a high churn rate occurs in customers who use mobile devices, especially the Android OS

# Correlation analysis

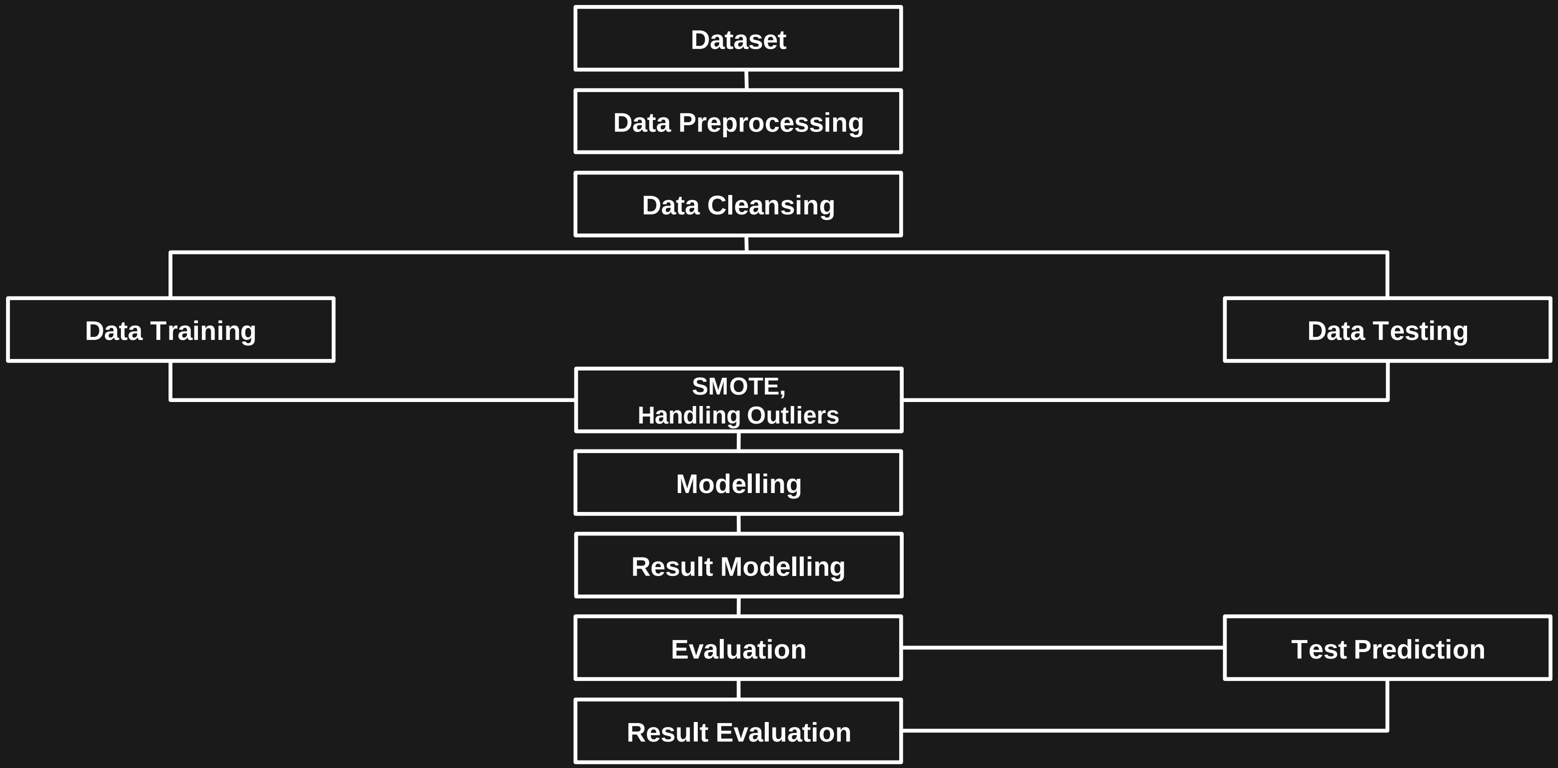



- Variables with high correlation coefficient values are, **total shipment fee** and **number of transaction**' with a correlation value of 0.98 (positive correlation), **number of transaction** and **number of promo** with correlation value 0.95 (positive correlation), **number of promo** and **total shipment fee** with correlation value 0.93 (positive correlation)
- **Total shipment fee** are not included in the model (high multicollinearity)
- The most negative correlations with **churn** is **number of transaction** (-0,4)
- The most positive correlations with **churn** is **delivery time** (0.0074).

# 05.

# Model Development & Evaluation

# Model Development



# Baseline Model

| Model | Precision | | Recall | | F1-Score | | Accuracy |
|---|---|---|---|---|---|---|---|
| | 1 | 0 | 1 | 0 | 1 | 0 | |
| Gradient Boosting | 0.88 | 0.89 | 0.93 | 0.94 | 0.91 | 0.91 | 0.85 |
| Cat Boost Classifier | 0.88 | 0.90 | 0.93 | 0.96 | 0.90 | 0.93 | 0.85 |
| Logistic Regression | 0.86 | 0.85 | 0.96 | 0.96 | 0.90 | 0.90 | 0.84 |
| KNN | 0.86 | 0.88 | 0.93 | 0.96 | 0.89 | 0.92 | 0.82 |
| Decision Tree | 0.87 | 0.99 | 0.85 | 0.99 | 0.86 | 0.99 | 0.78 |

From the five basic models that we compare, the **Gradient Boosting** has the best performance with precision score 88% and 89%, then for accuracy score has 85%.

Next, the Gradient Boosting model will be evaluated with hyperparameter tuning, let's see the results!

# Tuned Model Results - Evaluation

| Base Model - Tuned | | | | |
|---|---|---|---|---|
| Metric | Base Model | | Tuned | |
| | 1 | 0 | 1 | 0 |
| **Accuracy** | 0.85 | 0.86 | 0.86 | 0.85 |
| **Precision** | 0.88 | 0.89 | 0.89 | 0.89 |
| **Recal** | 0.93 | 0.94 | 0.93 | 0.94 |
| **F1-Score** | 0.91 | 0.91 | 0.91 | 0.92 |
| **MSE** | 0.14 | | 0.14 | |

After we did parameter tuning, the classification metric model improved in terms of its accuracy and precision metrics.

- ➔ The previous accuracy of 85% and 86% increased to 86% and 86%.
- ➔ The previous precision was 88% and 89% to 89% and 89%.

This means, gradient boosting classifier has the best performance in this case of customer churn.

# Best Parameter & Metric Error - Evaluation

```
# set parameter
grid = {
    'learning_rate':[0.01,0.05,0.1],
    'n_estimators':np.arange(100,500,100),
}
```

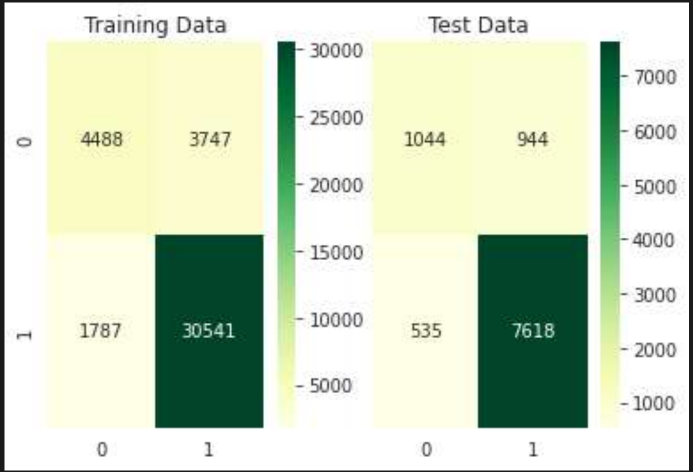
Using the GridSearchCV method, the best parameters are:

1. Learning Rate : 0.1
2. n estimator : 100

| Metrics Error | MAE | MSE | RMSE |
|---|---|---|---|
| Value | 0.14 | 0.14 | 0.38 |

This model has an error value of 14% and 38%. The error value is not yet fairly low (close to 0). In the future we will evaluate the model again to get a smaller error value.

# Confusion Matrix & Best Features - Evaluation




From the testing data it can be seen that, the predicted churn that actually churn was 7618, the predicted churn that actually didn't churn was 1044, the predictions of churn that actually didn't churn were 535 and the predictions of churn that actually didn't churn was 944.


Features Importance :

1. Number of transaction 0.77%
2. Number of Promo 0.13%
3. Total promo amount 0.019%

# 06.

# Conclusion & Recommendation

# Conclusion

1. The Gradient Boosting model is able to make predictions with fairly good model performance. The accuracy of the model in predicting customer churn is 86%.

1. Of the total 8 features as variable X, the variables that have the most positive effect on customer churn are total_promo_amount, number_of_transaction, and number_of_promo.

# Recommendations

### Develop New Features

Add a new features to review products and services to find out the level of customer satisfaction

### Develop a Questionnaire

Develop product reviews through a questionnaire, and send a feedback form via email to the customer regarding customer satisfaction with the product and user experiences

### Rewards for Loyal User

Give rewards to loyal customers, such as discounts, free gifts, and reduced shipping costs.

# Link

Google Collabs:

bit.ly/ColabsFinalProject_JermanTeam_DataScienc
e

Dashboard Visualization:

https://bit.ly/visualisasidashboard_jermanteam

## Customer Churn Prediction Using Machine Learning Models

# Thank You