# 1.Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import (confusion_matrix ,ConfusionMatrixDisplay ,accur
```

# 2.Loading Dataset

In [2]:
```
df = pd.read_csv('Predicting Coupon Acceptance on E-commerce Platforms.csv')
```

In [3]:
```
# Quick check the data
df.head()
```

Out[3]:

| | destination | passanger | weather | temperature | coupon | expiration |
|---|---|---|---|---|---|---|
| **0** | No Urgent Place | Alone | Sunny | 55 | Restaurant(<20) | 1d |
| **1** | No Urgent Place | Friend(s) | Sunny | 80 | Coffee House | 2h |
| **2** | No Urgent Place | Friend(s) | Sunny | 80 | Carry out & Take away | 2h |
| **3** | No Urgent Place | Friend(s) | Sunny | 80 | Coffee House | 2h |
| **4** | No Urgent Place | Friend(s) | Sunny | 80 | Coffee House | 1d |

5 rows × 25 columns

In [4]:
```
df.tail()
```

Out[4]:

| | destination | passanger | weather | temperature | coupon | expirati |
|---|---|---|---|---|---|---|
| **12679** | Home | Partner | Rainy | 55 | Carry out & Take away | |
| **12680** | Work | Alone | Rainy | 55 | Carry out & Take away | |
| **12681** | Work | Alone | Snowy | 30 | Coffee House | |
| **12682** | Work | Alone | Snowy | 30 | Bar | |
| **12683** | Work | Alone | Sunny | 80 | Restaurant(20-50) | |

5 rows × 25 columns

# 3.Data Preprocessing

In [5]:

```
# checking dataset shape
df.shape
```

Out[5]:

```
(12684, 25)
```

In [6]:

```
# Checking dataset size
df.size
```

Out[6]:

```
317100
```

In [7]:

```
# Checking columns in dataset
df.columns
```

Out[7]:

```
Index(['destination', 'passanger', 'weather', 'temperature', 'coupon',
       'expiration', 'gender', 'age', 'maritalStatus', 'has_children',
       'education', 'occupation', 'income', 'car', 'Bar', 'CoffeeHouse',
       'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50',
       'toCoupon_GEQ5min', 'toCoupon_GEQ15min', 'toCoupon_GEQ25min',
       'direction_same', 'direction_opp', 'Accept(Y/N?)'],
      dtype='object')
```

In [8]:

```
# Checking the Data-Types of the Columns
df.dtypes
```

Out[8]:
```
destination             object
passanger               object
weather                 object
temperature              int64
coupon                  object
expiration              object
gender                  object
age                     object
maritalStatus           object
has_children             int64
education               object
occupation              object
income                  object
car                     object
Bar                     object
CoffeeHouse             object
CarryAway               object
RestaurantLessThan20    object
Restaurant20To50        object
toCoupon_GEQ5min         int64
toCoupon_GEQ15min        int64
toCoupon_GEQ25min        int64
direction_same           int64
direction_opp            int64
Accept(Y/N?)             int64
dtype: object
```

In [9]:
```
#Checking the information of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12684 entries, 0 to 12683
Data columns (total 25 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   destination           12684 non-null  object
 1   passanger             12684 non-null  object
 2   weather               12684 non-null  object
 3   temperature           12684 non-null  int64 
 4   coupon                12684 non-null  object
 5   expiration            12684 non-null  object
 6   gender                12684 non-null  object
 7   age                   12684 non-null  object
 8   maritalStatus         12684 non-null  object
 9   has_children          12684 non-null  int64 
 10  education             12684 non-null  object
 11  occupation            12684 non-null  object
 12  income                12684 non-null  object
 13  car                   108 non-null    object
 14  Bar                   12577 non-null  object
 15  CoffeeHouse           12467 non-null  object
 16  CarryAway             12533 non-null  object
 17  RestaurantLessThan20  12554 non-null  object
 18  Restaurant20To50      12495 non-null  object
 19  toCoupon_GEQ5min      12684 non-null  int64 
 20  toCoupon_GEQ15min     12684 non-null  int64 
 21  toCoupon_GEQ25min     12684 non-null  int64 
 22  direction_same        12684 non-null  int64 
 23  direction_opp         12684 non-null  int64 
 24  Accept(Y/N?)          12684 non-null  int64 
dtypes: int64(8), object(17)
memory usage: 2.4+ MB
```

In [10]:
```
# is_claim Is the Target column
# Checking the value counts for number of unique values
df['Accept(Y/N?)'].value_counts()
```

Out[10]:
```
Accept(Y/N?)
1    7210
0    5474
Name: count, dtype: int64
```

In [11]:
```
# Checking the percentage of number of unique values are present
df['Accept(Y/N?)'].value_counts(normalize=True)*100
```

Out[11]:
```
Accept(Y/N?)
1    56.843267
0    43.156733
Name: proportion, dtype: float64
```

### Note1: The Target Column is Balenced

## Checking Missing and Duplicate Data

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
destination                 0
passanger                   0
weather                     0
temperature                 0
coupon                      0
expiration                  0
gender                      0
age                         0
maritalStatus               0
has_children                0
education                   0
occupation                  0
income                      0
car                     12576
Bar                       107
CoffeeHouse               217
CarryAway                 151
RestaurantLessThan20      130
Restaurant20To50          189
toCoupon_GEQ5min            0
toCoupon_GEQ15min           0
toCoupon_GEQ25min           0
direction_same              0
direction_opp               0
Accept(Y/N?)                0
dtype: int64
```

In [ ]:
```

```

In [13]:
```
df.duplicated().sum() # 291 duplicated values in datas
```

Out[13]:
```
291
```

## Droping Car col. bcz too many null values are in

In [14]:
```
df.drop(columns=['car'], axis=1, inplace=True)
```

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12684 entries, 0 to 12683
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   destination           12684 non-null  object
 1   passanger             12684 non-null  object
 2   weather               12684 non-null  object
 3   temperature           12684 non-null  int64 
 4   coupon                12684 non-null  object
 5   expiration            12684 non-null  object
 6   gender                12684 non-null  object
 7   age                   12684 non-null  object
 8   maritalStatus         12684 non-null  object
 9   has_children          12684 non-null  int64 
 10  education             12684 non-null  object
 11  occupation            12684 non-null  object
 12  income                12684 non-null  object
 13  Bar                   12577 non-null  object
 14  CoffeeHouse           12467 non-null  object
 15  CarryAway             12533 non-null  object
 16  RestaurantLessThan20  12554 non-null  object
 17  Restaurant20To50      12495 non-null  object
 18  toCoupon_GEQ5min      12684 non-null  int64 
 19  toCoupon_GEQ15min     12684 non-null  int64 
 20  toCoupon_GEQ25min     12684 non-null  int64 
 21  direction_same        12684 non-null  int64 
 22  direction_opp         12684 non-null  int64 
 23  Accept(Y/N?)          12684 non-null  int64 
dtypes: int64(8), object(16)
memory usage: 2.3+ MB
```

## Imputing Missing values / Handeling Missing values

In [16]:
```
# Fill missing values with 'unknown' for categorical and median for numerica
for i in ['Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20', 'Restau
    df[i].fillna('unknown', inplace=True)
```

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
destination             0
passanger               0
weather                 0
temperature             0
coupon                  0
expiration              0
gender                  0
age                     0
maritalStatus           0
has_children            0
education               0
occupation              0
income                  0
Bar                     0
CoffeeHouse             0
CarryAway               0
RestaurantLessThan20    0
Restaurant20To50        0
toCoupon_GEQ5min        0
toCoupon_GEQ15min       0
toCoupon_GEQ25min       0
direction_same          0
direction_opp           0
Accept(Y/N?)            0
dtype: int64
```

### Handeling Duplicate data

In [18]:
```
df.drop_duplicates(inplace=True) # Dropping duplicate values
```

In [19]:
```
df.duplicated().sum() # All duplicate data handled/Drop
```

Out[19]:
```
0
```

Note 2:

Zero Missing values

Zero Duplicate values

## 4.Sorting Categorical and Numerical columns in list

In [20]:
```
categorical=[]
numerical=[]
```

Seperating categorical column with Number of unique values

In [21]:
```
for i in df.select_dtypes(include='object').columns:
    print(i,":",df[i].nunique())
    categorical.append(i)
```

```
print("\n\n")
print(categorical)
```

```
destination : 3
passanger : 4
weather : 3
coupon : 5
expiration : 2
gender : 2
age : 8
maritalStatus : 5
education : 6
occupation : 25
income : 9
Bar : 6
CoffeeHouse : 6
CarryAway : 6
RestaurantLessThan20 : 6
Restaurant20To50 : 6



['destination', 'passanger', 'weather', 'coupon', 'expiration', 'gender', 'a
ge', 'maritalStatus', 'education', 'occupation', 'income', 'Bar', 'CoffeeHou
se', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50']
```

## Separating Categorical and Numerical

In [22]:

```
categorical = df.select_dtypes(include='O').columns.tolist()
numerical = df.select_dtypes(include=['int','float']).columns.tolist()

print("Categorical columns are: \n",categorical,"\n")
print("Numerical columns are: \n",numerical)
```

```
Categorical columns are:
 ['destination', 'passanger', 'weather', 'coupon', 'expiration', 'gender',
'age', 'maritalStatus', 'education', 'occupation', 'income', 'Bar', 'CoffeeH
ouse', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50']

Numerical columns are:
 ['temperature', 'has_children', 'toCoupon_GEQ5min', 'toCoupon_GEQ15min', 't
oCoupon_GEQ25min', 'direction_same', 'direction_opp', 'Accept(Y/N?)']
```

In [23]:

```
for i in df.select_dtypes(include=['int64','float64']).columns:
    print(i,":",df[i].nunique())
    if df[i].nunique()>25:
        numerical.append(i)
    else:
        categorical.append(i)
print("\n\n")
print(numerical)
```

```
temperature : 3
has_children : 2
toCoupon_GEQ5min : 1
toCoupon_GEQ15min : 2
toCoupon_GEQ25min : 2
direction_same : 2
direction_opp : 2
Accept(Y/N?) : 2
```

```
['temperature', 'has_children', 'toCoupon_GEQ5min', 'toCoupon_GEQ15min', 'to
Coupon_GEQ25min', 'direction_same', 'direction_opp', 'Accept(Y/N?)']
```

In [24]:
```
df['income'].value_counts()
```

Out[24]:
```
income
$25000 - $37499     1972
$12500 - $24999     1795
$37500 - $49999     1760
$100000 or More     1688
$50000 - $62499     1624
Less than $12500    1013
$87500 - $99999      865
$75000 - $87499      844
$62500 - $74999      832
Name: count, dtype: int64
```

In [25]:
```
df['income'].value_counts(normalize=True)*100
```

Out[25]:
```
income
$25000 - $37499     15.912209
$12500 - $24999     14.483983
$37500 - $49999     14.201565
$100000 or More     13.620592
$50000 - $62499     13.104172
Less than $12500     8.173969
$87500 - $99999      6.979747
$75000 - $87499      6.810296
$62500 - $74999      6.713467
Name: proportion, dtype: float64
```

In [26]:
```
df['occupation'].value_counts()
```

Out[26]:
```
occupation
Unemployed                                   1828
Student                                      1550
Computer & Mathematical                      1360
Sales & Related                              1066
Education&Training&Library                    921
Management                                    806
Office & Administrative Support               631
Arts Design Entertainment Sports & Media      617
Business & Financial                          536
Retired                                       489
Food Preparation & Serving Related            293
Healthcare Support                            237
Healthcare Practitioners & Technical          237
Community & Social Services                   236
Legal                                         218
Transportation & Material Moving              214
Architecture & Engineering                    172
Personal Care & Service                       172
Protective Service                            172
Life Physical Social Science                  168
Construction & Extraction                     150
Installation Maintenance & Repair             129
Production Occupations                        108
Building & Grounds Cleaning & Maintenance      42
Farming Fishing & Forestry                     41
Name: count, dtype: int64
```

In [27]:
```
df['occupation'].value_counts(normalize=True)*100
```

Out[27]:
```
occupation
Unemployed                                   14.750262
Student                                      12.507060
Computer & Mathematical                      10.973937
Sales & Related                               8.601630
Education&Training&Library                    7.431615
Management                                    6.503671
Office & Administrative Support               5.091584
Arts Design Entertainment Sports & Media      4.978617
Business & Financial                          4.325022
Retired                                       3.945776
Food Preparation & Serving Related            2.364238
Healthcare Support                            1.912370
Healthcare Practitioners & Technical          1.912370
Community & Social Services                   1.904301
Legal                                         1.759058
Transportation & Material Moving              1.726781
Architecture & Engineering                    1.387880
Personal Care & Service                       1.387880
Protective Service                            1.387880
Life Physical Social Science                  1.355604
Construction & Extraction                     1.210361
Installation Maintenance & Repair             1.040910
Production Occupations                        0.871460
Building & Grounds Cleaning & Maintenance     0.338901
Farming Fishing & Forestry                    0.330832
Name: proportion, dtype: float64
```

Income group of 2500-37499 Has highest accept For yes

Similarly for lowest accept for yes And for no as well

In [ ]:

In [ ]:

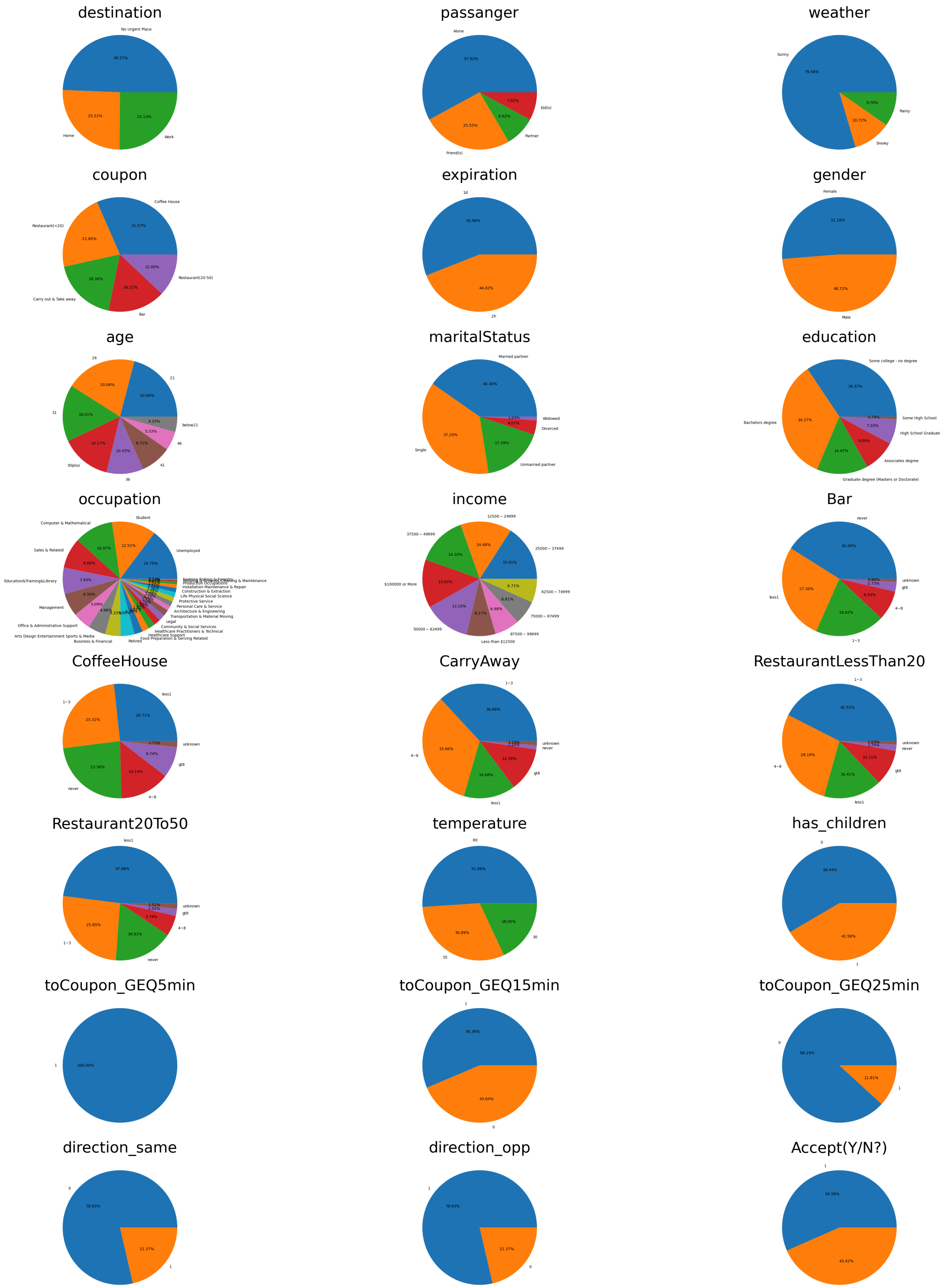
# 5 Exploratory Data Analysis(EDA)

A] Categorical EDA

Pie chart

In [28]:
```
fig=plt.figure(figsize=(40,90))
count=1
for i in categorical:
     if df[i].nunique():
         keys=df[i].value_counts().keys()
         values=df[i].value_counts().values
         plt.subplot(15,3,count)
         plt.pie(x=values, labels=keys, autopct='%0.2f%%')
         plt.title(i, fontsize=40)
```

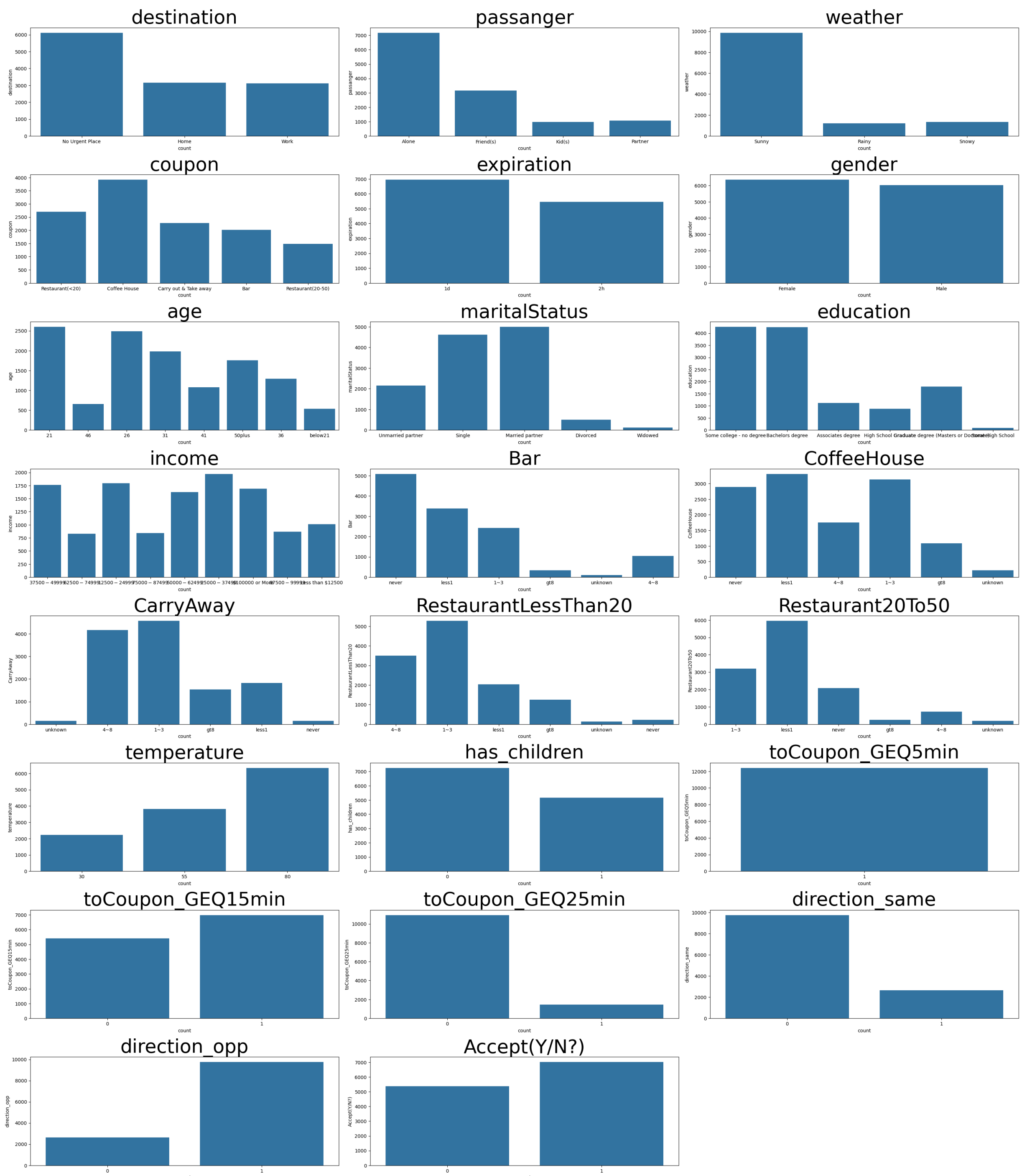
```
        fig.tight_layout()
        count+=1
plt.show()
```




## Count Plot

In [29]:

```
# Example countplot for a single categorical variable
count=1
fig=plt.figure(figsize=(30,60))
for i in categorical:
     if df[i].nunique()<25:
         plt.subplot(14,3,count)
         sns.countplot(x=i, data=df,)
         plt.title(i, fontsize=40)
         plt.xlabel('count')
         plt.ylabel(i)
         fig.tight_layout()
         count+=1
plt.show()
```



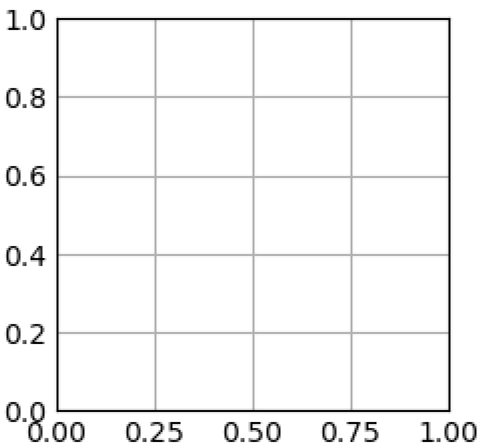
In [ ]:

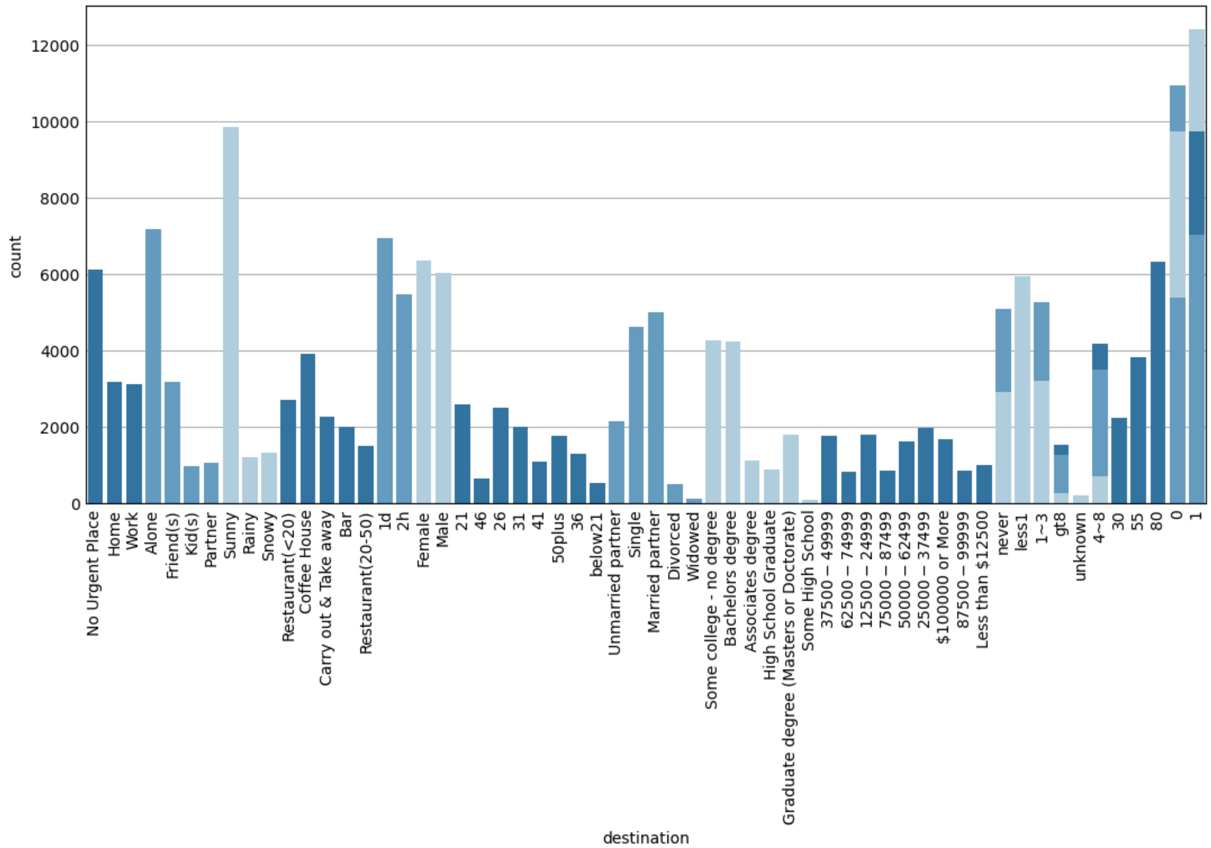
```

```

Stem plot

In [30]:

```
plt.style.use('_mpl-gallery')

# make data
#x = 0.5 + np.arange(8)
#y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]

# plot
fig, ax = plt.subplots()
count=1
fig=plt.figure(figsize=(120,140))
for i in categorical:
    if df[i].nunique() <=11:
        plt.subplot(26,10,count)
        sns.countplot(x=i, data=df,)
        plt.xticks(rotation=90)

plt.show()
```






In [33]:

```
# Counting no.of unique values in every catogarical columns
df[categorical].nunique()
```

Out[33]:

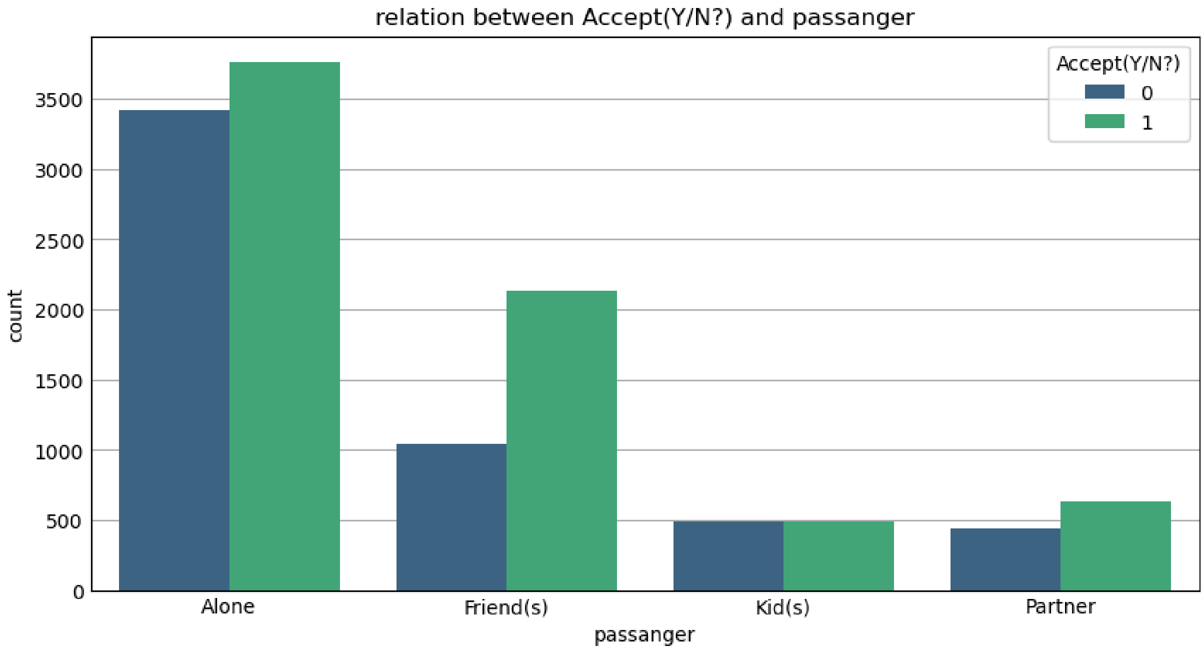
```
destination              3
passanger                4
weather                  3
coupon                   5
expiration               2
gender                   2
age                      8
maritalStatus            5
education                6
occupation              25
income                   9
Bar                      6
CoffeeHouse              6
CarryAway                6
RestaurantLessThan20     6
Restaurant20To50         6
temperature              3
has_children             2
toCoupon_GEQ5min         1
toCoupon_GEQ15min        2
toCoupon_GEQ25min        2
direction_same           2
direction_opp            2
Accept(Y/N?)             2
dtype: int64
```

In [34]:
```
# checking unique values of categorical columns which are less than 25
for i in categorical:
    if df[i].nunique()<25:
        print(i,df[i].unique(),end="\n\n")
```

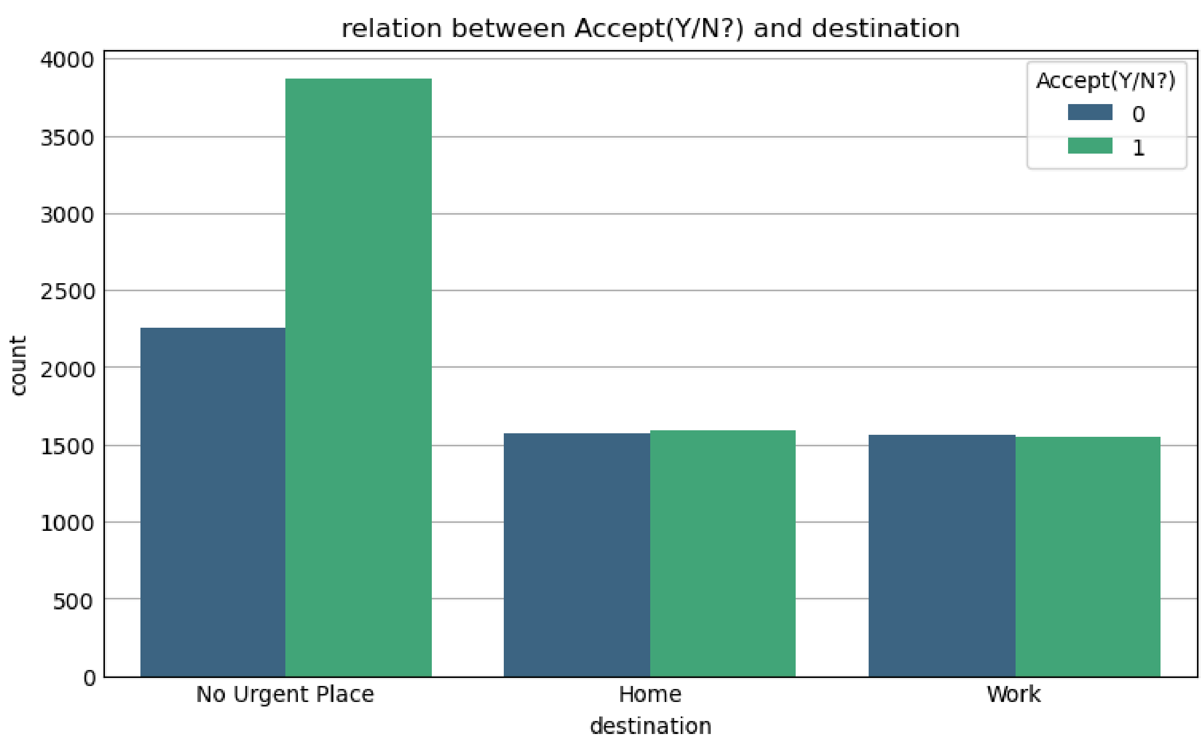
```
destination ['No Urgent Place' 'Home' 'Work']

passanger ['Alone' 'Friend(s)' 'Kid(s)' 'Partner']

weather ['Sunny' 'Rainy' 'Snowy']

coupon ['Restaurant(<20)' 'Coffee House' 'Carry out & Take away' 'Bar'
 'Restaurant(20-50)']

expiration ['1d' '2h']

gender ['Female' 'Male']

age ['21' '46' '26' '31' '41' '50plus' '36' 'below21']

maritalStatus ['Unmarried partner' 'Single' 'Married partner' 'Divorced' 'Wi
dowed']

education ['Some college - no degree' 'Bachelors degree' 'Associates degree'
 'High School Graduate' 'Graduate degree (Masters or Doctorate)'
 'Some High School']

income ['$37500 - $49999' '$62500 - $74999' '$12500 - $24999' '$75000 - $874
99'
 '$50000 - $62499' '$25000 - $37499' '$100000 or More' '$87500 - $99999'
 'Less than $12500']

Bar ['never' 'less1' '1~3' 'gt8' 'unknown' '4~8']

CoffeeHouse ['never' 'less1' '4~8' '1~3' 'gt8' 'unknown']

CarryAway ['unknown' '4~8' '1~3' 'gt8' 'less1' 'never']

RestaurantLessThan20 ['4~8' '1~3' 'less1' 'gt8' 'unknown' 'never']

Restaurant20To50 ['1~3' 'less1' 'never' 'gt8' '4~8' 'unknown']

temperature [55 80 30]

has_children [1 0]

toCoupon_GEQ5min [1]

toCoupon_GEQ15min [0 1]

toCoupon_GEQ25min [0 1]

direction_same [0 1]

direction_opp [1 0]

Accept(Y/N?) [1 0]
```

Count plot with hue of target column(target_col= 'Accept(Y/N?)')

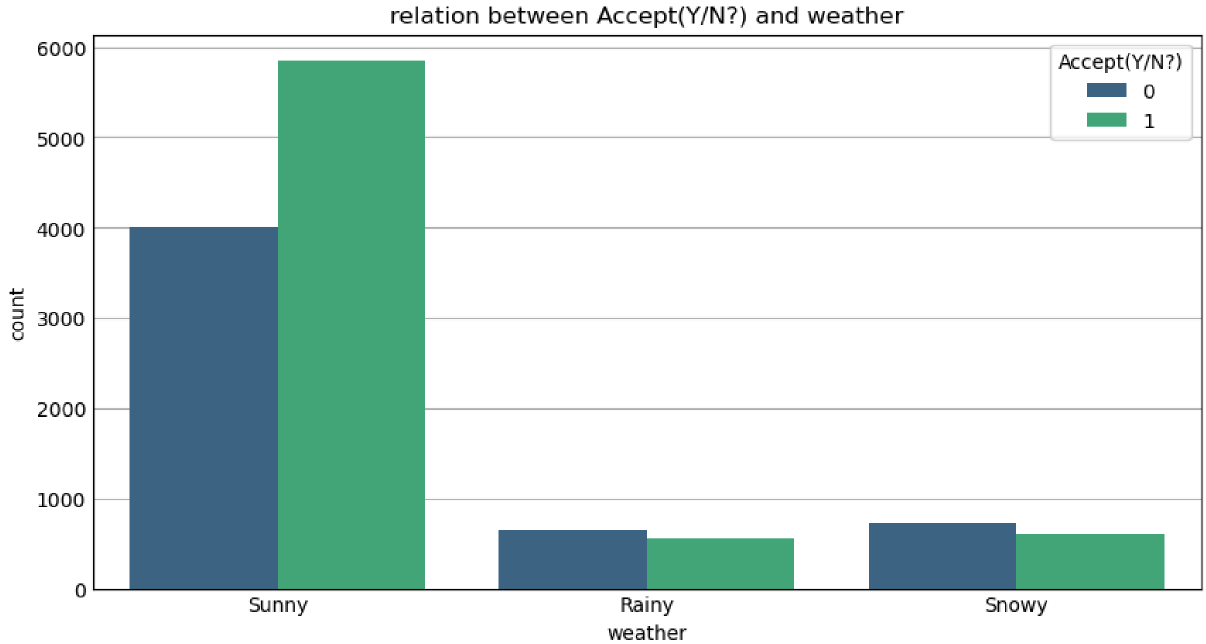
In [35]:
```
# Count plot with hue of target column
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "passanger" , hue ="Accept(Y/N?)", palette='viri
plt.title("relation between Accept(Y/N?) and passanger")
plt.show()
```



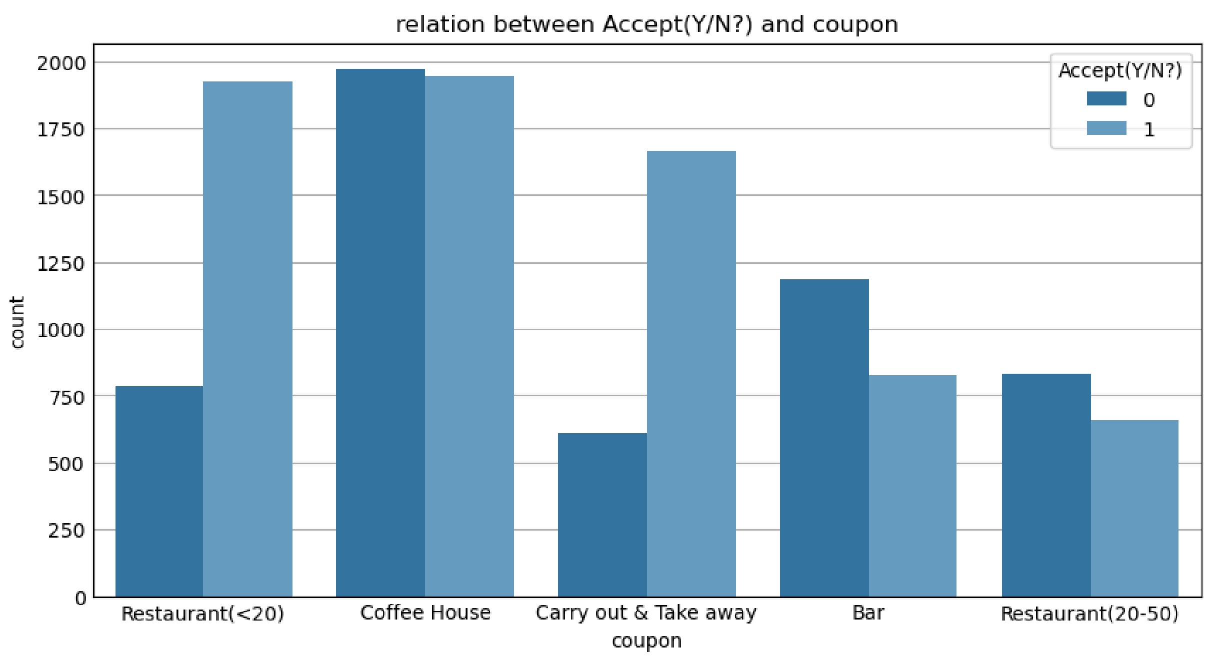

In [36]:
```
plt.figure(figsize=(7,4))
sns.countplot(data = df ,x= "destination" , hue ="Accept(Y/N?)", palette='vi
plt.title("relation between Accept(Y/N?) and destination")
plt.show()
```



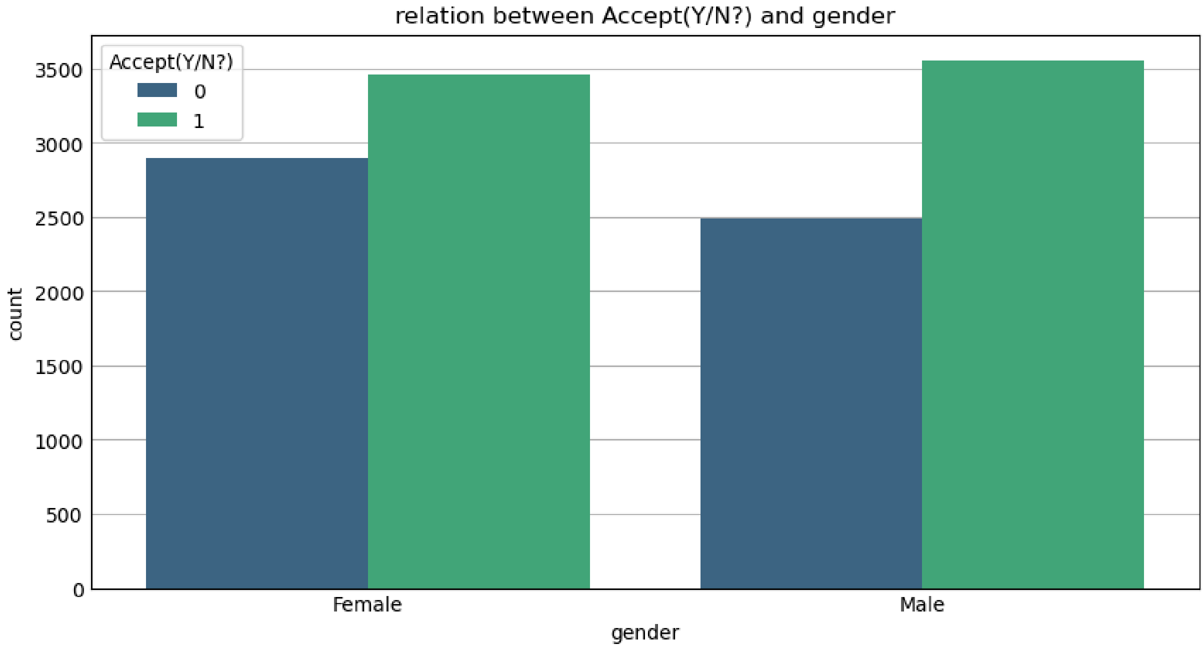
In [37]:
```
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "weather" , hue ="Accept(Y/N?)", palette='viridi
plt.title("relation between Accept(Y/N?) and weather")
plt.show()
```



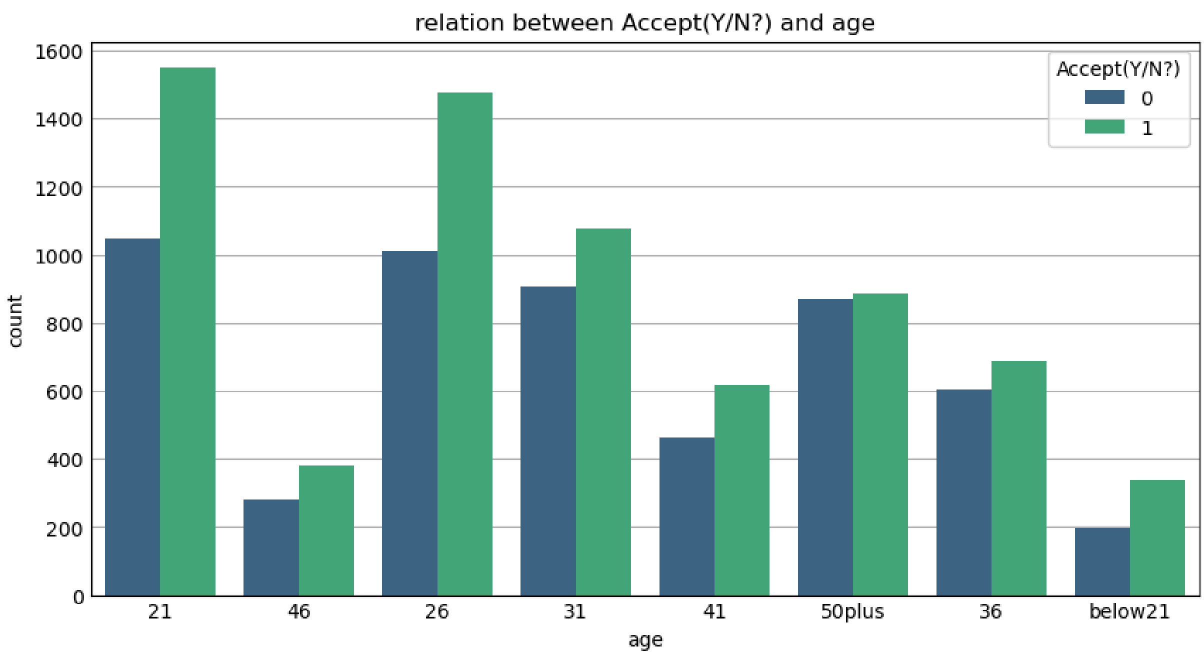

In [38]:
```
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "coupon" , hue ="Accept(Y/N?)",)
plt.title("relation between Accept(Y/N?) and coupon")
plt.show()
```



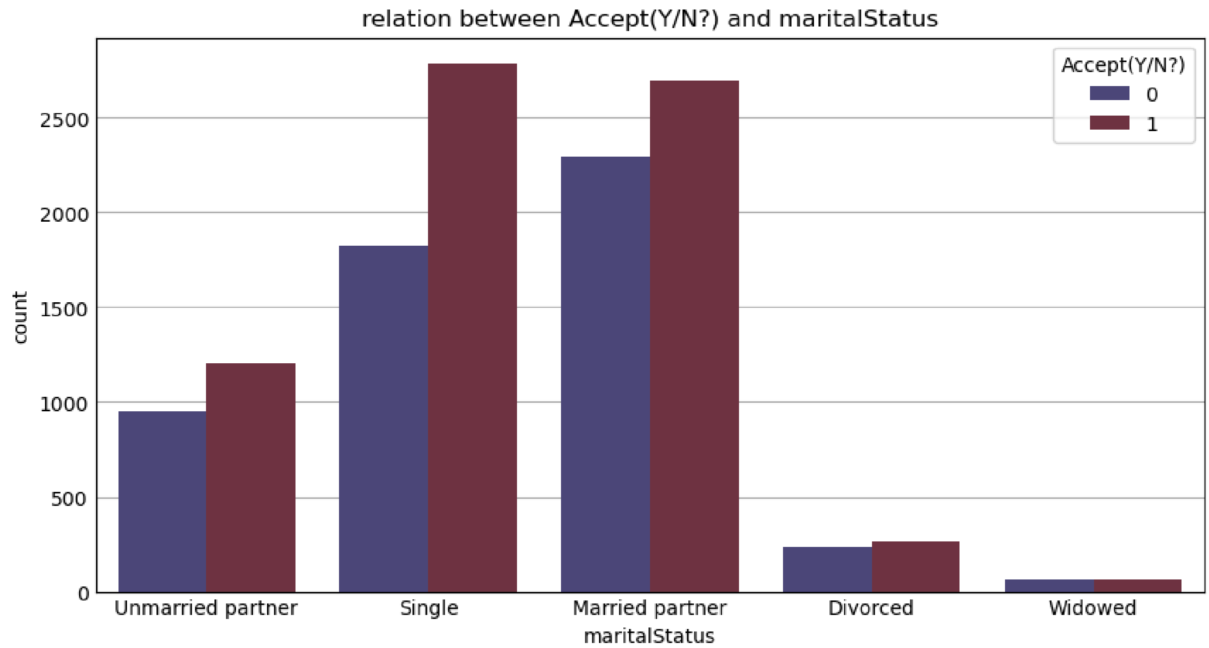

In [39]:
```
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "gender" , hue ="Accept(Y/N?)", palette='viridis
plt.title("relation between Accept(Y/N?) and gender")
plt.show()
```



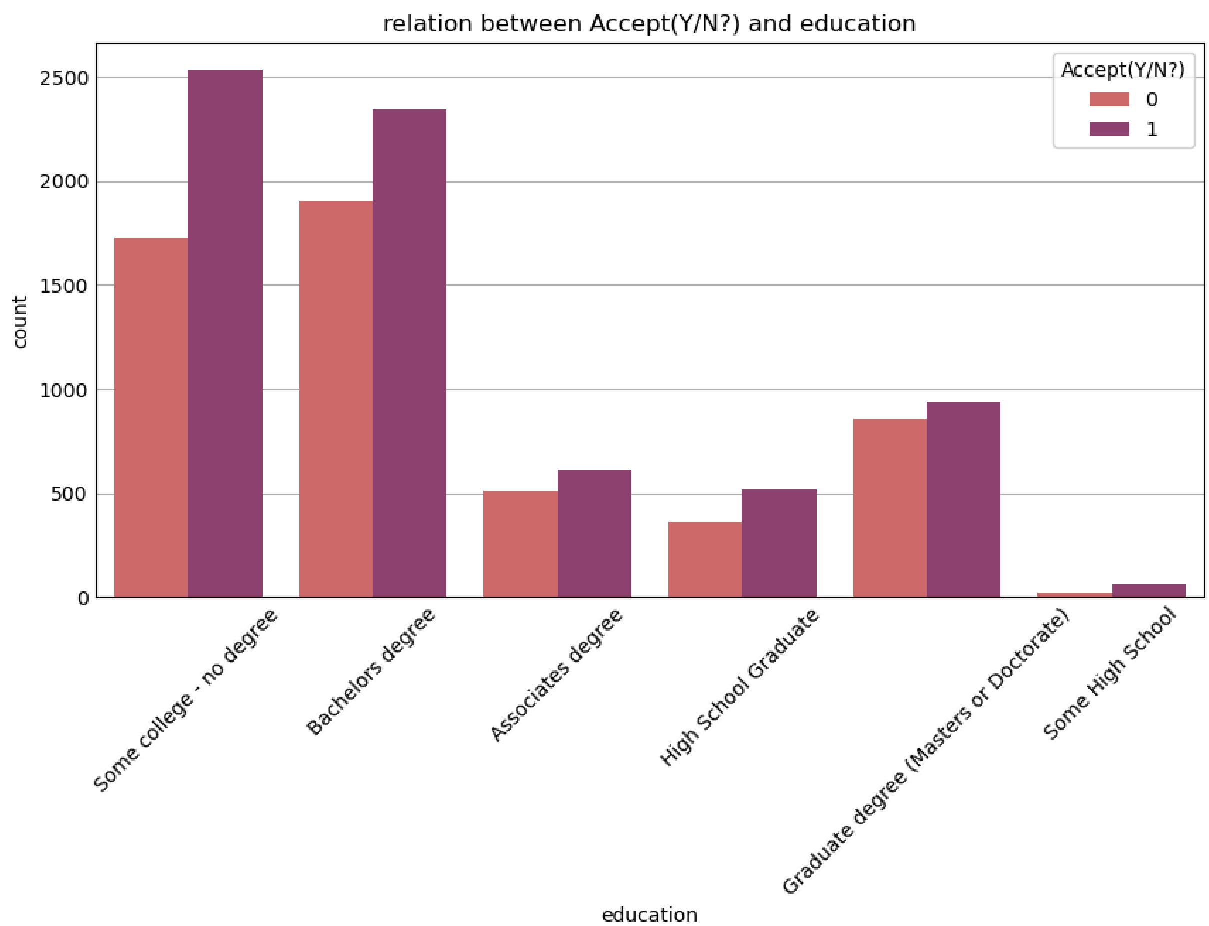

In [40]:
```
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "age" , hue ="Accept(Y/N?)", palette='viridis')
plt.title("relation between Accept(Y/N?) and age")
plt.show()
```



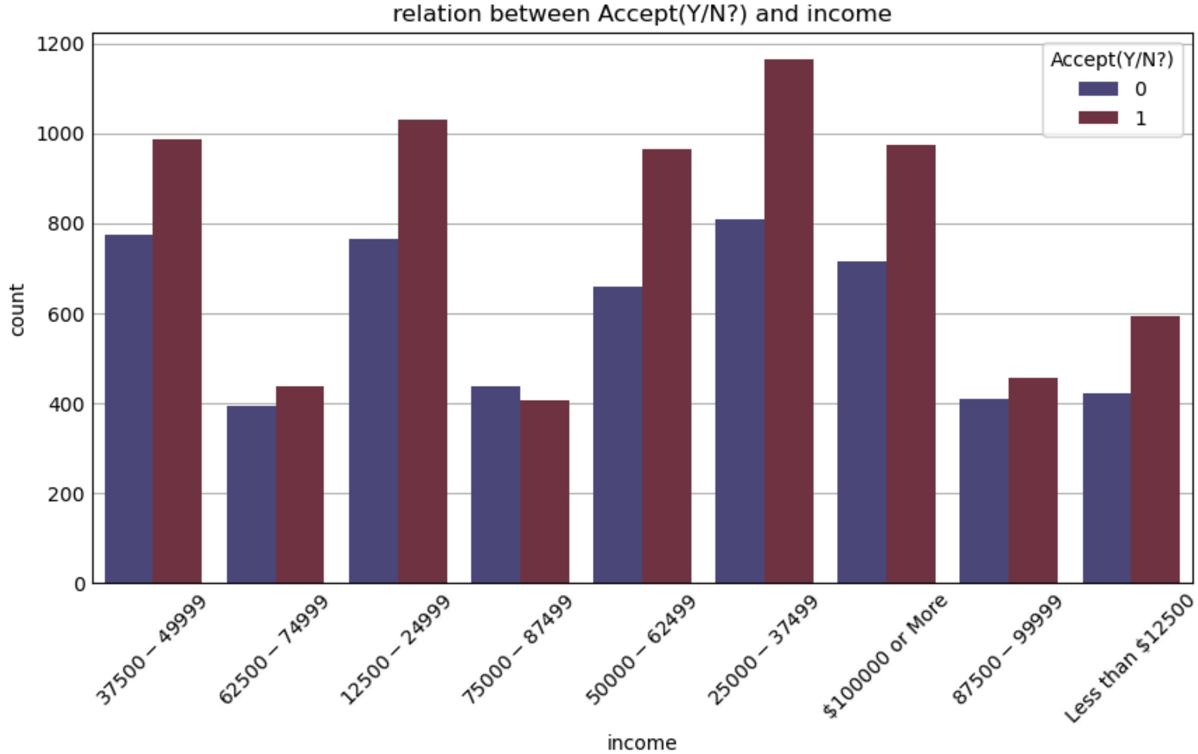

In [41]:
```
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "maritalStatus" , hue ="Accept(Y/N?)", palette='
plt.title("relation between Accept(Y/N?) and maritalStatus")
plt.show()
```




In [42]:
```
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "education" , hue ="Accept(Y/N?)", palette='flar
plt.title("relation between Accept(Y/N?) and education")
plt.xticks(rotation=45)
plt.show()
```



In [43]:
```
plt.figure(figsize=(8,4))
sns.countplot(data = df ,x= "income" , hue ="Accept(Y/N?)", palette='icefire
plt.title("relation between Accept(Y/N?) and income")
plt.xticks(rotation=45)
plt.show()
```

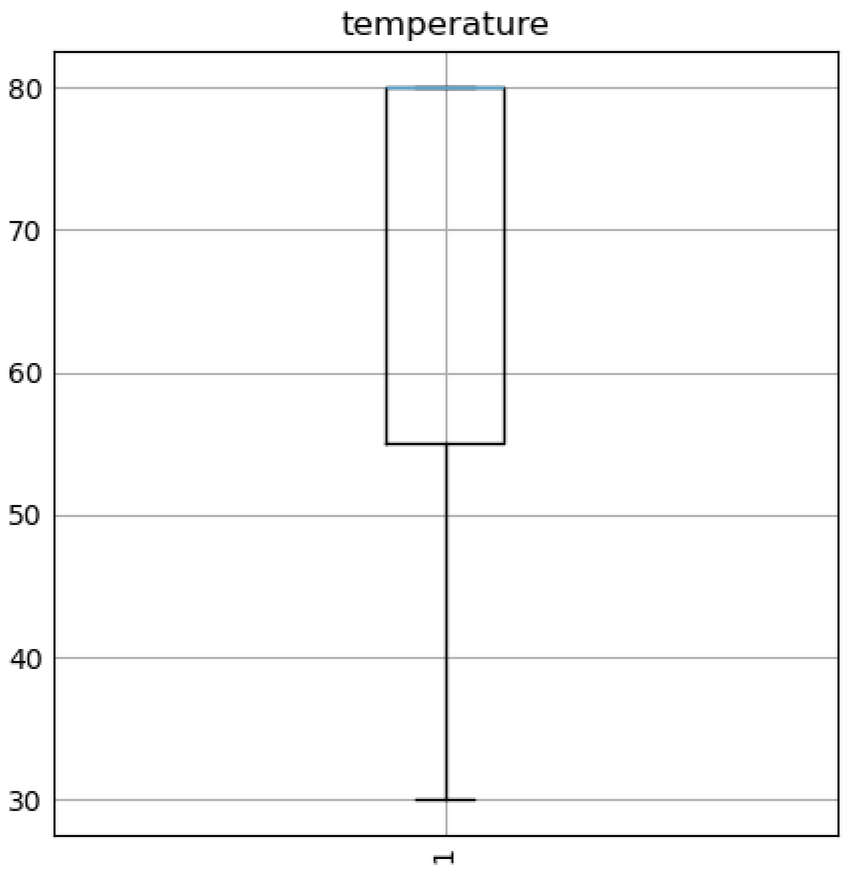


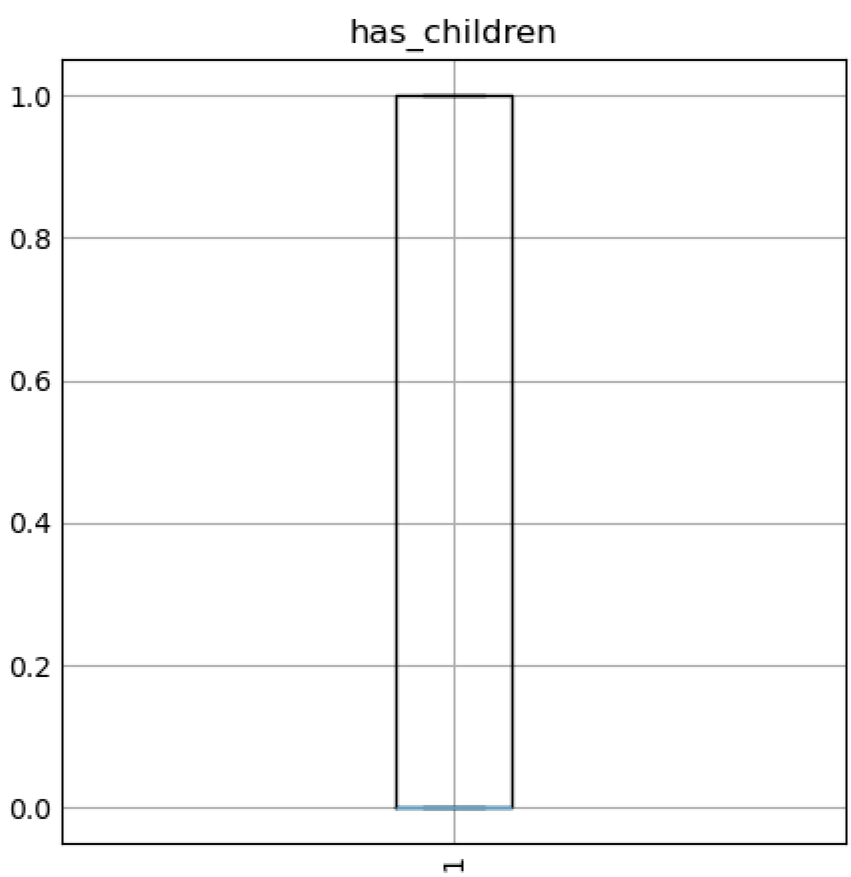

In [ ]:
```
```

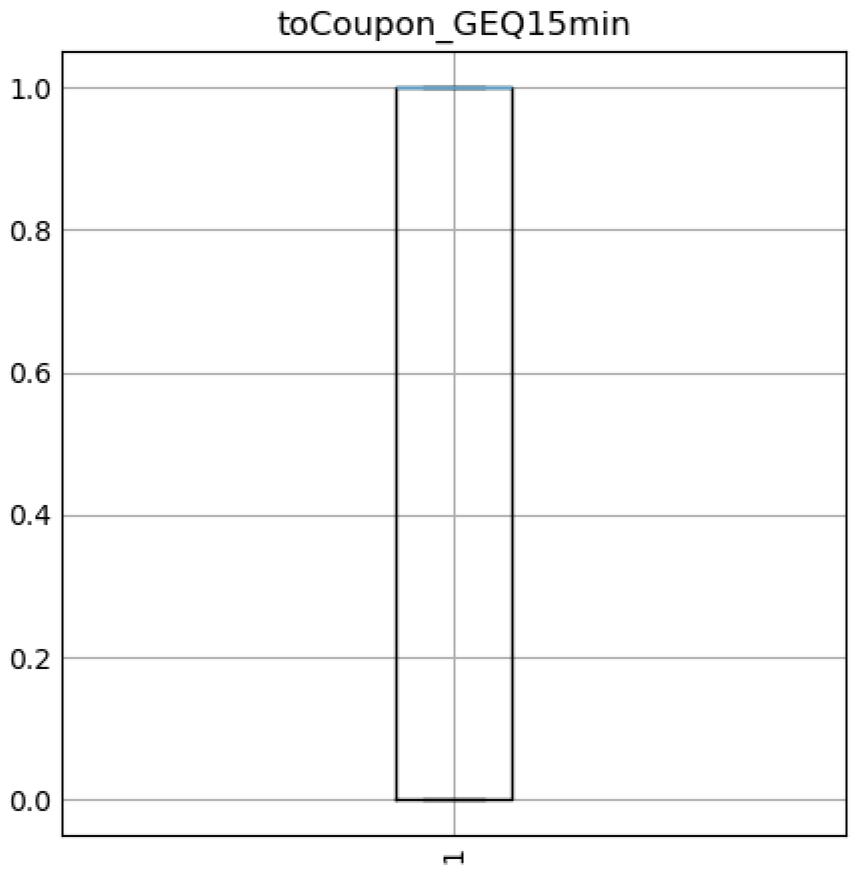
# B] Numerical EDA

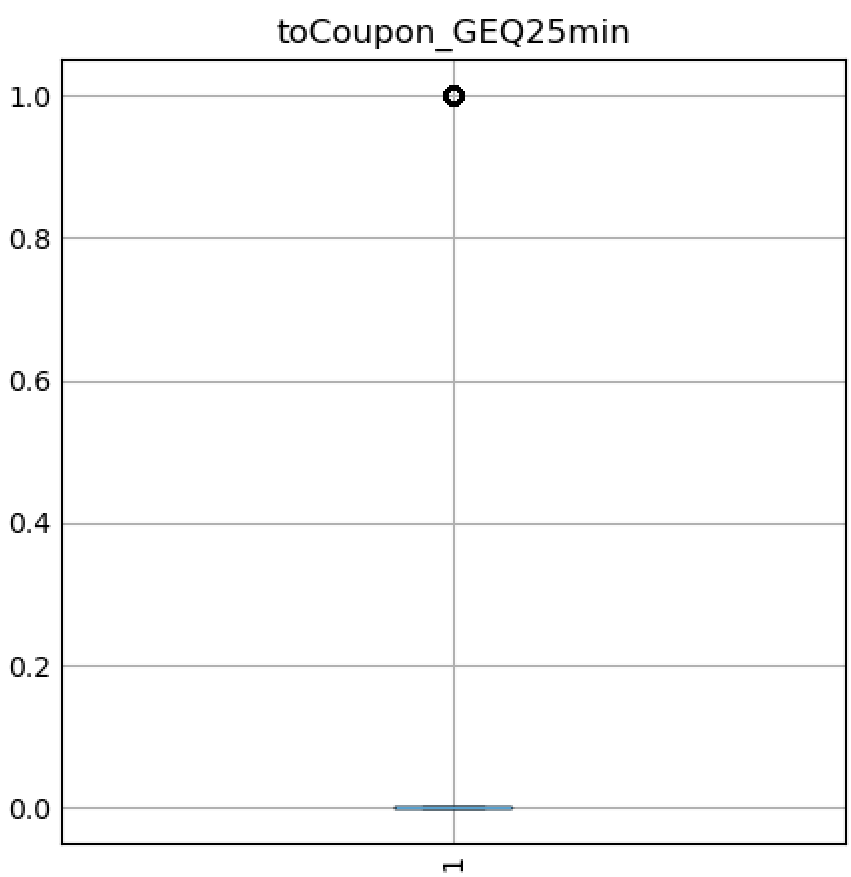
# Box Plot

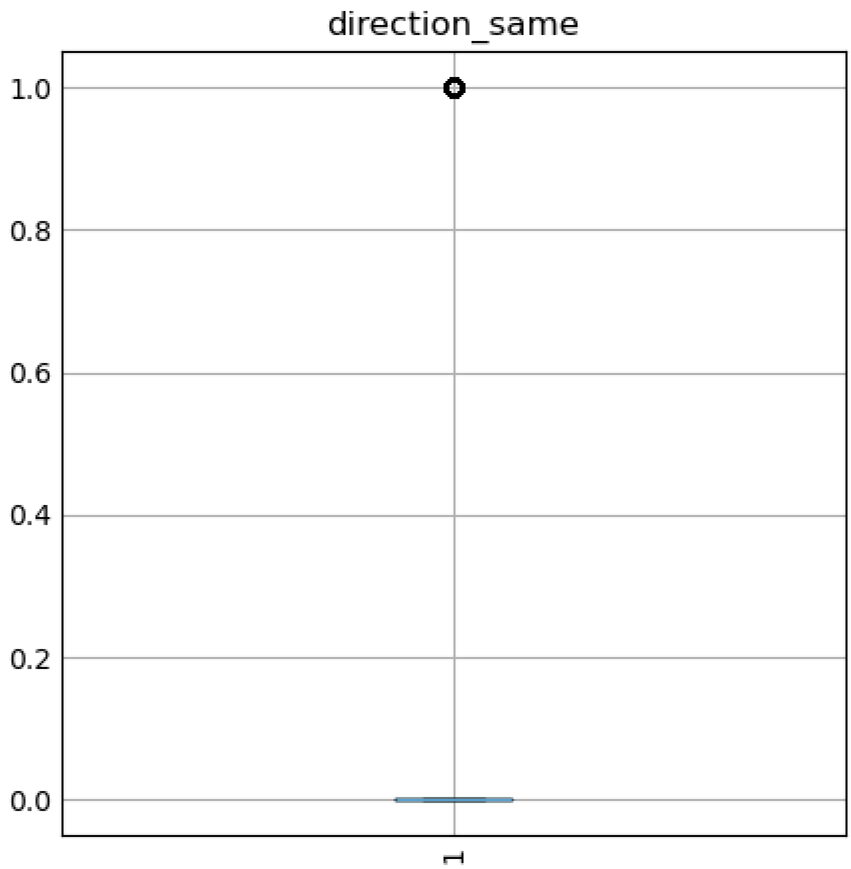
In [66]:
```
# whole numerical col. visulizing in boxplot
for i in numerical:
   # if df[i].nunique()<1000:

       plt.figure(figsize=(4,4))
       plt.title(i)
       plt.boxplot(x=i, data=df)
       plt.xticks(rotation=90)
plt.show()
```

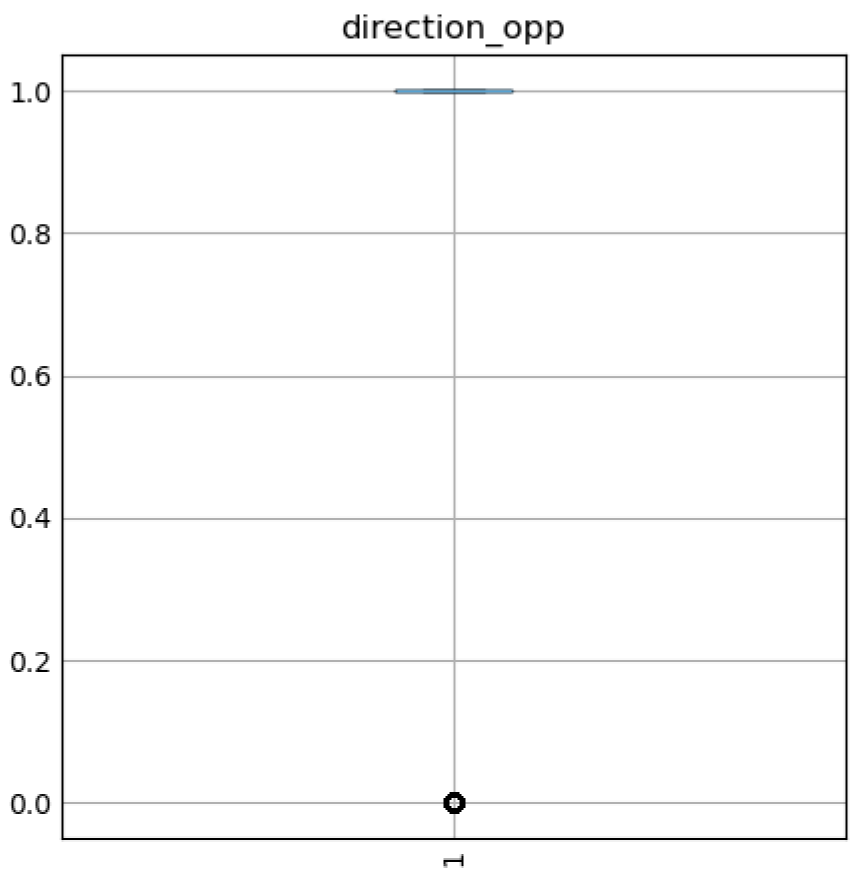

temperature
80
70
60
50
40
30
1

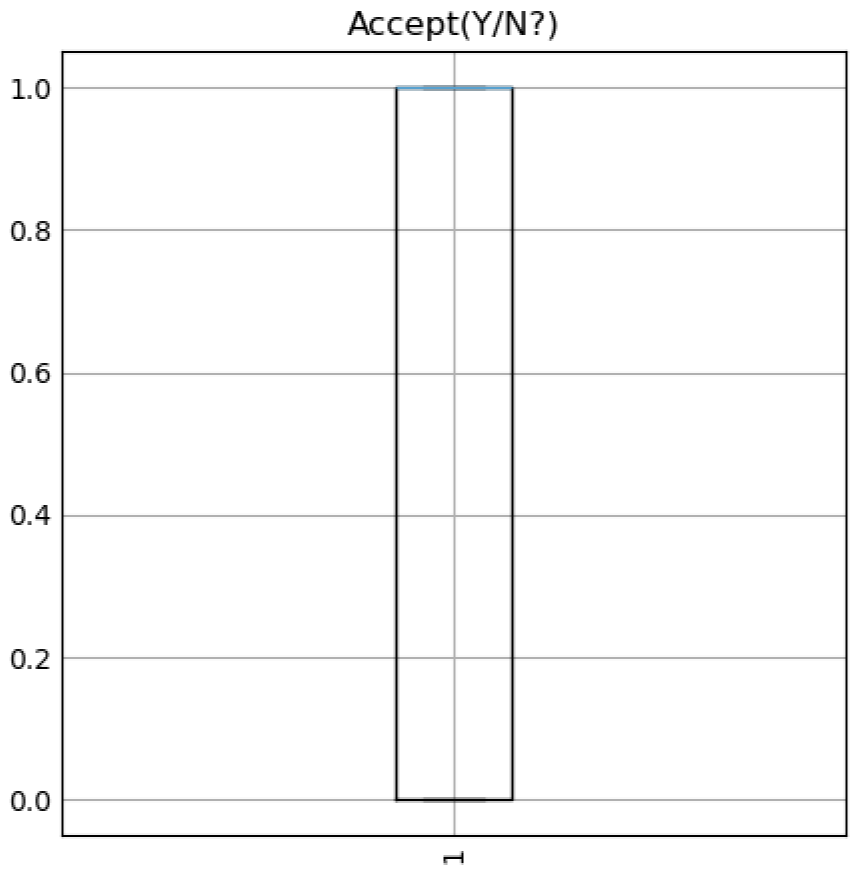

has_children
1.0
0.8
0.6
0.4
0.2
0.0
1


toCoupon_GEQ15min
1.0
0.8
0.6
0.4
0.2
0.0
1


toCoupon_GEQ25min
1.0
0.8
0.6
0.4
0.2
0.0
1


direction_same
1.0
0.8
0.6
0.4
0.2
0.0
1


direction_opp
1.0
0.8
0.6
0.4
0.2
0.0
1




There are none outliers in Boxplot

In [45]:
```
df[numerical].describe()
```

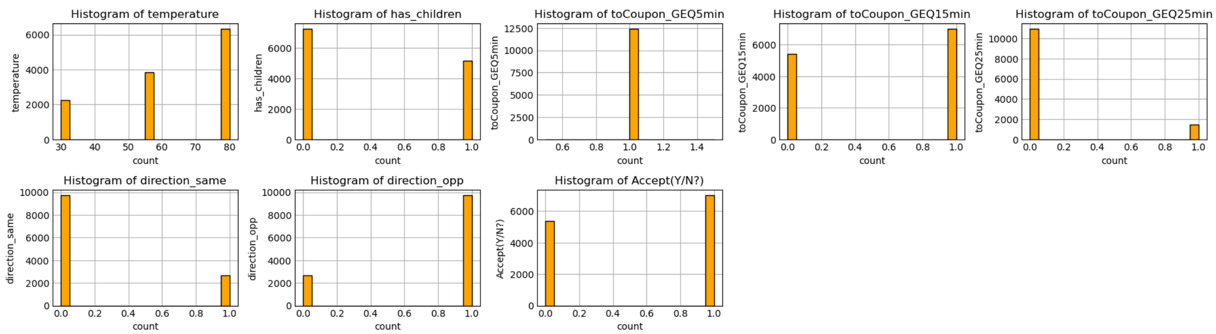
Out[45]:

| | temperature | has_children | toCoupon_GEQ5min | toCoupon_GEQ15min |
|---|---|---|---|---|
| **count** | 12393.000000 | 12393.000000 | 12393.0 | 12393.000000 |
| **mean** | 63.252643 | 0.415557 | 1.0 | 0.563625 |
| **std** | 19.075396 | 0.492838 | 0.0 | 0.495955 |
| **min** | 30.000000 | 0.000000 | 1.0 | 0.000000 |
| **25%** | 55.000000 | 0.000000 | 1.0 | 0.000000 |
| **50%** | 80.000000 | 0.000000 | 1.0 | 1.000000 |
| **75%** | 80.000000 | 1.000000 | 1.0 | 1.000000 |
| **max** | 80.000000 | 1.000000 | 1.0 | 1.000000 |

## Histogram

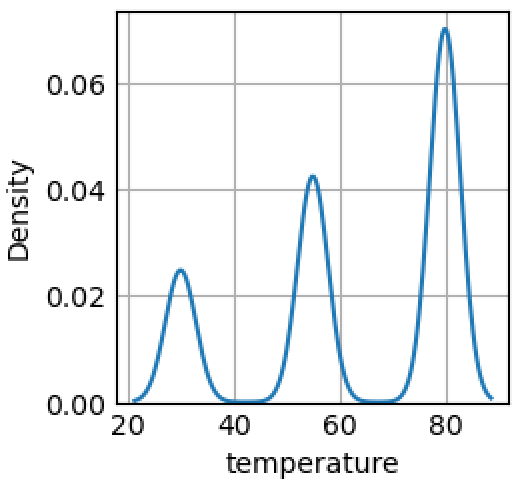
In [46]:
```
plt.figure(figsize=(18, 12))
for idx, i in enumerate(numerical):
    plt.subplot(5, 5, idx + 1)  # Adjust grid size (5x5) based on number of
    plt.hist(df[i], bins=20, color='orange', edgecolor='black')
    plt.title(f'Histogram of {i}')
```

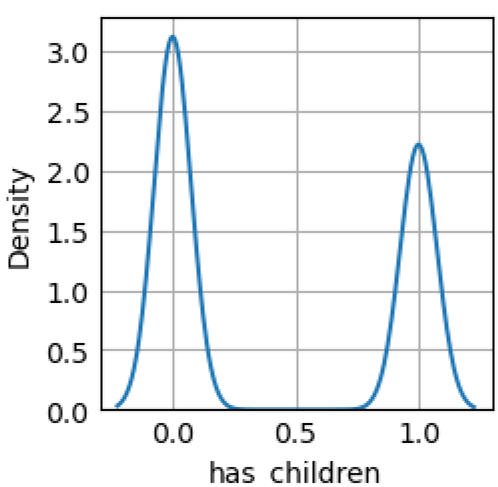
```
    plt.xlabel('count')
    plt.ylabel(i)
plt.tight_layout()
plt.show()
```



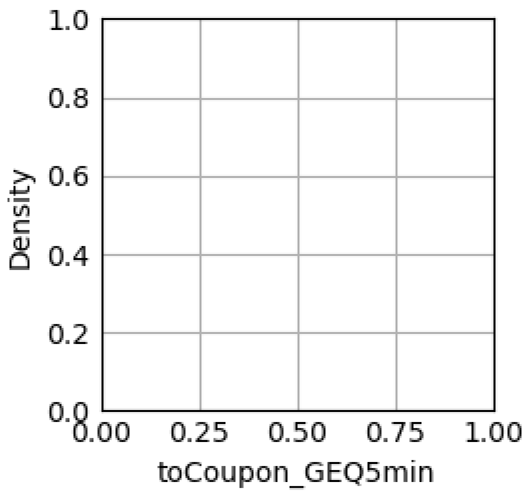

## KDE Plot

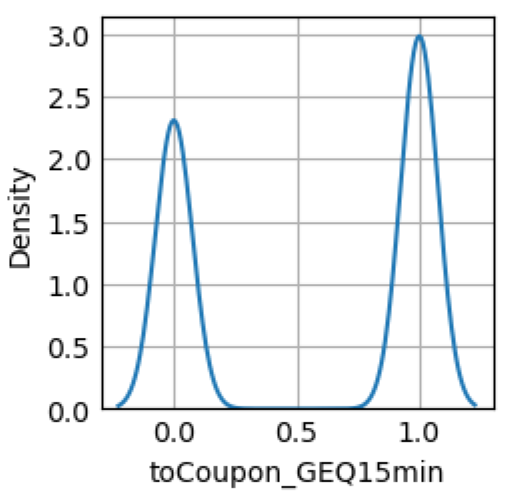
In [47]:
```
for i in numerical:
    if df[i].nunique() < 1000:
        sns.kdeplot(data=df[i])
        plt.show()
```

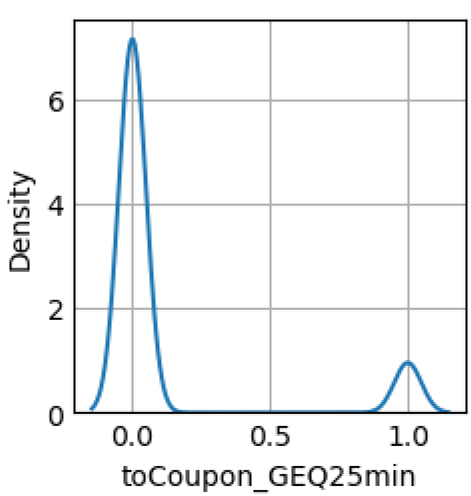


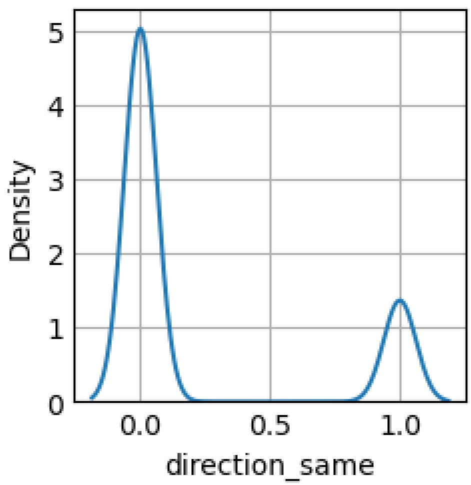



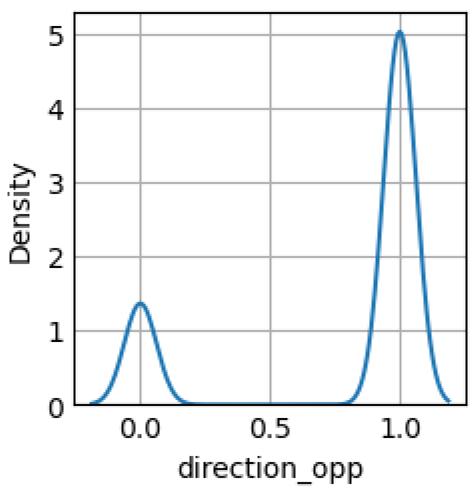
```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_12880\2305473507.py:3: UserWar
ning: Dataset has 0 variance; skipping density estimate. Pass `warn_singular
=False` to disable this warning.
  sns.kdeplot(data=df[i])
```

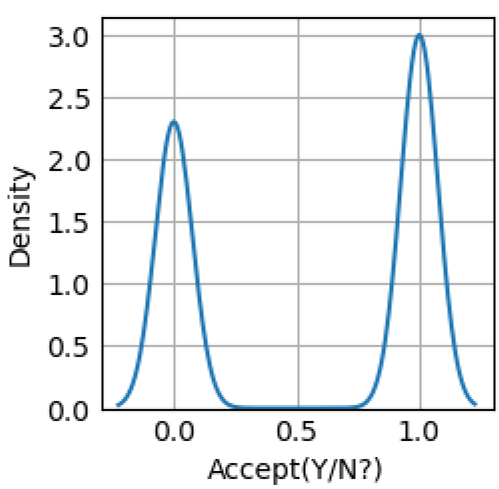





Correlation matrix of numerical column

In [48]:
```
df[numerical].corr()
```

Out[48]:

| | temperature | has_children | toCoupon_GEQ5min | toCoupo |
|---|---|---|---|---|
| **temperature** | 1.000000 | -0.016963 | NaN | |
| **has_children** | -0.016963 | 1.000000 | NaN | |
| **toCoupon_GEQ5min** | NaN | NaN | NaN | |
| **toCoupon_GEQ15min** | -0.141124 | 0.078686 | NaN | |
| **toCoupon_GEQ25min** | -0.230067 | -0.011651 | NaN | |
| **direction_same** | 0.088885 | -0.032276 | NaN | |
| **direction_opp** | -0.088885 | 0.032276 | NaN | |
| **Accept(Y/N?)** | 0.064074 | -0.044889 | NaN | |

In [49]:
```
df['toCoupon_GEQ5min'].isnull().sum()
```

Out[49]:
```
0
```

In [50]:
```
df['toCoupon_GEQ5min'].dtype
```

Out[50]:
```
dtype('int64')
```

In [51]:
```
df['toCoupon_GEQ5min'].nunique()
```

Out[51]:
```
1
```

In [52]:
```
df[numerical].nunique()
```

Out[52]:
```
temperature          3
has_children         2
toCoupon_GEQ5min     1
toCoupon_GEQ15min    2
toCoupon_GEQ25min    2
direction_same       2
direction_opp        2
Accept(Y/N?)         2
dtype: int64
```

## Dropping 'toCoupon_GEQ5min' because there is nan values in column

In [53]:
```
df.drop(columns=['toCoupon_GEQ5min'], axis=1, inplace=True)
```

In [59]:
```
print(df.columns) ## checking the column is it drop or not..
```

```
Index(['destination', 'passanger', 'weather', 'temperature', 'coupon',
       'expiration', 'gender', 'age', 'maritalStatus', 'has_children',
       'education', 'occupation', 'income', 'Bar', 'CoffeeHouse', 'CarryAwa
y',
       'RestaurantLessThan20', 'Restaurant20To50', 'toCoupon_GEQ15min',
       'toCoupon_GEQ25min', 'direction_same', 'direction_opp', 'Accept(Y/
N?)'],
      dtype='object')
```

In [60]:

```
numerical = [col for col in df.select_dtypes(include=['int64', 'float64']).c
print(numerical)  # Ensure 'toCoupon_GEQ5min' is not in this list
```

```
['temperature', 'has_children', 'toCoupon_GEQ15min', 'toCoupon_GEQ25min', 'd
irection_same', 'direction_opp', 'Accept(Y/N?)']
```

In [61]:

```
df[numerical].corr()
```

Out[61]:

| | temperature | has_children | toCoupon_GEQ15min | toCoup |
|---|---|---|---|---|
| **temperature** | 1.000000 | -0.016963 | -0.141124 | |
| **has_children** | -0.016963 | 1.000000 | 0.078686 | |
| **toCoupon_GEQ15min** | -0.141124 | 0.078686 | 1.000000 | |
| **toCoupon_GEQ25min** | -0.230067 | -0.011651 | 0.321919 | |
| **direction_same** | 0.088885 | -0.032276 | -0.297284 | |
| **direction_opp** | -0.088885 | 0.032276 | 0.297284 | |
| **Accept(Y/N?)** | 0.064074 | -0.044889 | -0.086050 | |

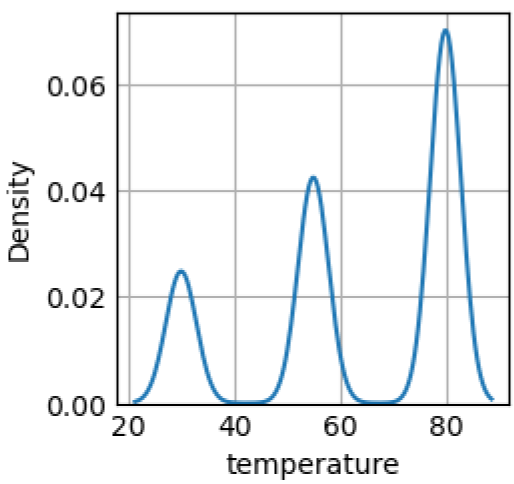
In [62]:

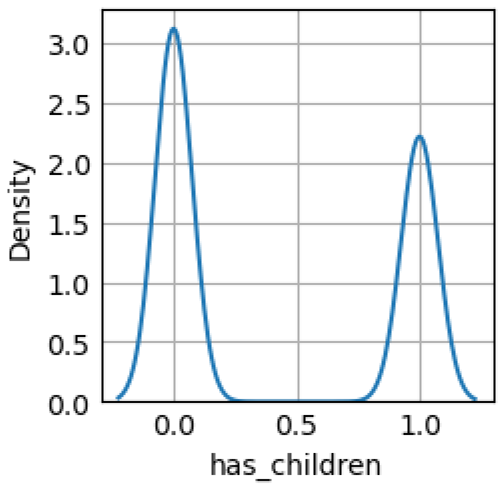
```
df.isnull().sum() # Checking Null values again
```

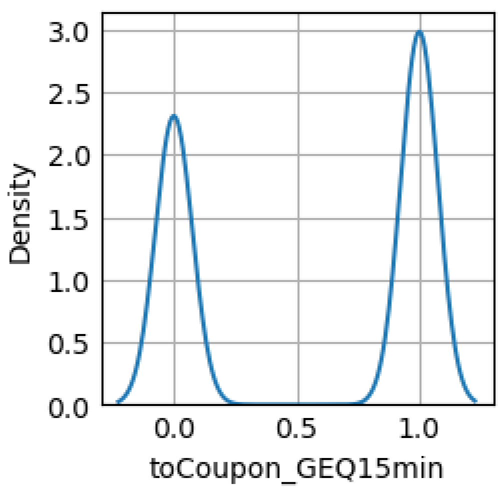
```
Out[62]: destination              0
         passanger                0
         weather                  0
         temperature              0
         coupon                   0
         expiration               0
         gender                   0
         age                      0
         maritalStatus            0
         has_children             0
         education                0
         occupation               0
         income                   0
         Bar                      0
         CoffeeHouse              0
         CarryAway                0
         RestaurantLessThan20     0
         Restaurant20To50         0
         toCoupon_GEQ15min        0
         toCoupon_GEQ25min        0
         direction_same           0
         direction_opp            0
         Accept(Y/N?)             0
         dtype: int64
```

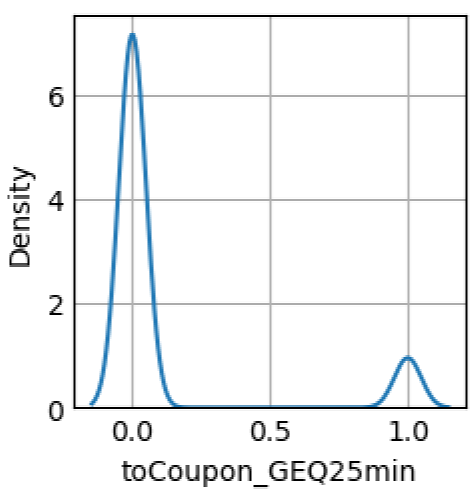
KDE plot after removing nan values col.

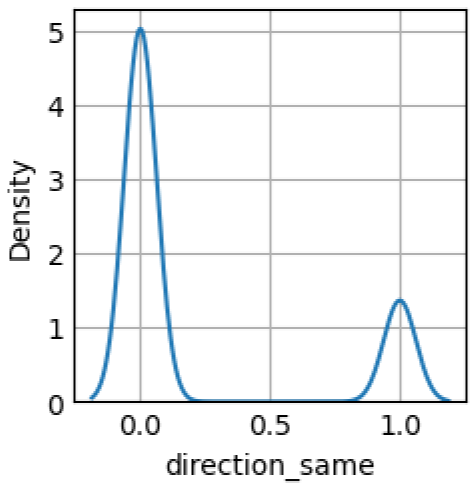
```
In [63]: for i in numerical:
             if df[i].nunique():
                 sns.kdeplot(data=df[i])
                 plt.show()
```

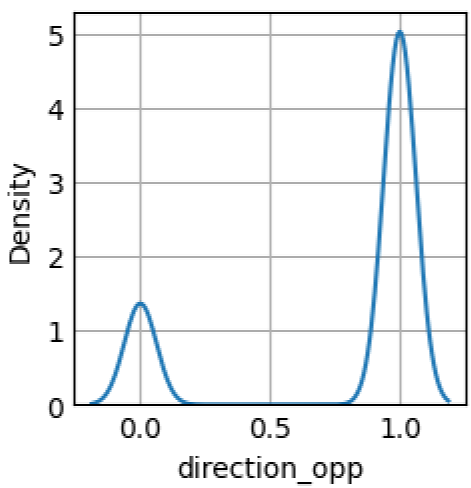


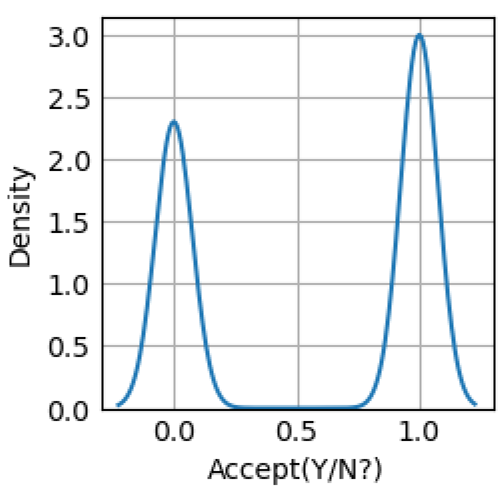

Density
has_children


Density
toCoupon_GEQ15min


Density
toCoupon_GEQ25min

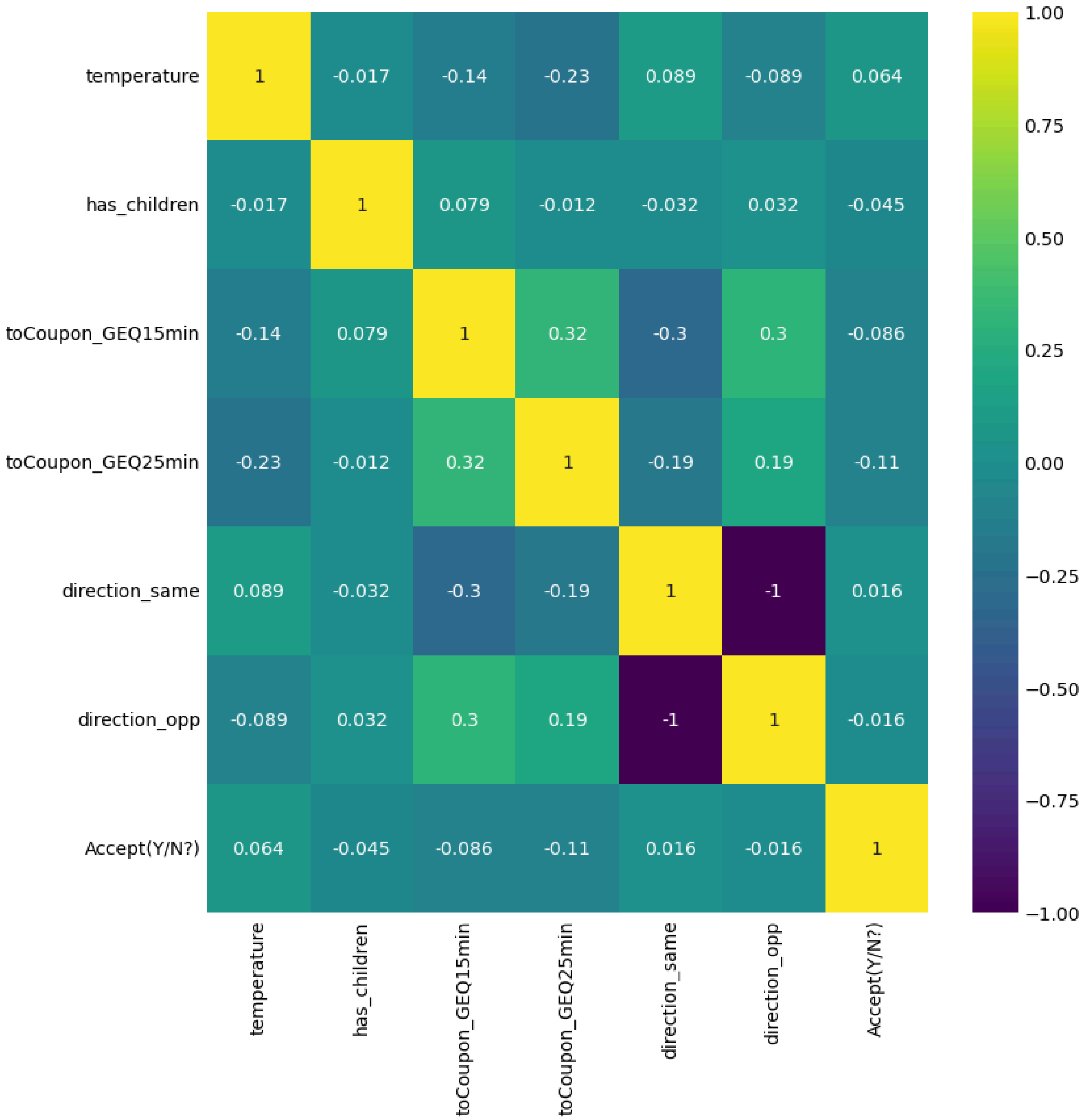



## Heatmap

In [64]:
```
plt.figure(figsize=(7,7))
sns.heatmap(data=df[numerical].corr(),cmap='viridis',vmin=-1,vmax=1,annot=Tr
plt.show()
```




# 6 Lable encoding

In [73]:
```
from sklearn.preprocessing import LabelEncoder
```

In [75]:
```
le=LabelEncoder()
```

In [78]:
```
df2=pd.DataFrame()
for i in df.columns:
      df2[i]=le.fit_transform(df[i])
df2
```

Out[78]:

| | destination | passanger | weather | temperature | coupon | expiration | ge |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 2 | 1 | 4 | 0 | |
| **1** | 1 | 1 | 2 | 2 | 2 | 1 | |
| **2** | 1 | 1 | 2 | 2 | 1 | 1 | |
| **3** | 1 | 1 | 2 | 2 | 2 | 1 | |
| **4** | 1 | 1 | 2 | 2 | 2 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | |
| **12388** | 0 | 3 | 0 | 1 | 1 | 0 | |
| **12389** | 2 | 0 | 0 | 1 | 1 | 0 | |
| **12390** | 2 | 0 | 1 | 0 | 2 | 0 | |
| **12391** | 2 | 0 | 1 | 0 | 0 | 0 | |
| **12392** | 2 | 0 | 2 | 2 | 3 | 1 | |

12393 rows × 23 columns

# 7 X&Y Train Test split

In [79]:

```
x=df2.drop(columns=['Accept(Y/N?)'],axis=1)
y=df2['Accept(Y/N?)']
```

In [80]:

```
x.head()
```

Out[80]:

| | destination | passanger | weather | temperature | coupon | expiration | gender |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 2 | 1 | 4 | 0 | 0 |
| **1** | 1 | 1 | 2 | 2 | 2 | 1 | 0 |
| **2** | 1 | 1 | 2 | 2 | 1 | 1 | 0 |
| **3** | 1 | 1 | 2 | 2 | 2 | 1 | 0 |
| **4** | 1 | 1 | 2 | 2 | 2 | 0 | 0 |

5 rows × 22 columns

In [81]:

```
y.head()
```

Out[81]:

```
0    1
1    0
2    1
3    0
4    0
Name: Accept(Y/N?), dtype: int64
```

In [82]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=42,test_size
```

In [83]:
```
x_train.head()
```

Out[83]:

| | destination | passanger | weather | temperature | coupon | expiration | ge |
|---|---|---|---|---|---|---|---|
| **3288** | 2 | 0 | 2 | 2 | 3 | 0 | |
| **10934** | 0 | 0 | 1 | 0 | 2 | 1 | |
| **9835** | 2 | 0 | 0 | 1 | 1 | 1 | |
| **7660** | 2 | 0 | 0 | 1 | 3 | 0 | |
| **1691** | 1 | 2 | 2 | 2 | 1 | 1 | |

5 rows × 22 columns

In [84]:
```
x_test.head()
```

Out[84]:

| | destination | passanger | weather | temperature | coupon | expiration | ge |
|---|---|---|---|---|---|---|---|
| **10146** | 1 | 0 | 0 | 1 | 2 | 0 | |
| **9834** | 2 | 0 | 2 | 0 | 0 | 0 | |
| **6902** | 0 | 0 | 2 | 0 | 1 | 1 | |
| **2145** | 1 | 1 | 2 | 2 | 4 | 1 | |
| **2344** | 0 | 0 | 2 | 1 | 3 | 0 | |

5 rows × 22 columns

In [85]:
```
y_train.head()
```

Out[85]:
```
3288     0
10934    0
9835     1
7660     1
1691     1
Name: Accept(Y/N?), dtype: int64
```

In [86]:
```
y_test.head()
```

Out[86]:
```
10146    1
9834     1
6902     1
2145     1
2344     1
Name: Accept(Y/N?), dtype: int64
```

# 8 Model Building

## Logistic Regression

In [88]:
```
lr=LogisticRegression()
```

In [89]:
```
lr.fit(x_train,y_train)
y_train_predict_lr=lr.predict(x_train)
y_test_predict_lr=lr.predict(x_test)
```

In [90]:
```
print("train")
print(f"accuracy_score{accuracy_score(y_train,y_train_predict_lr)},precision
```

```
train
accuracy_score0.6301190236029857,precision_score0.6461538461538462,recall_sc
ore0.7740563530037214
```

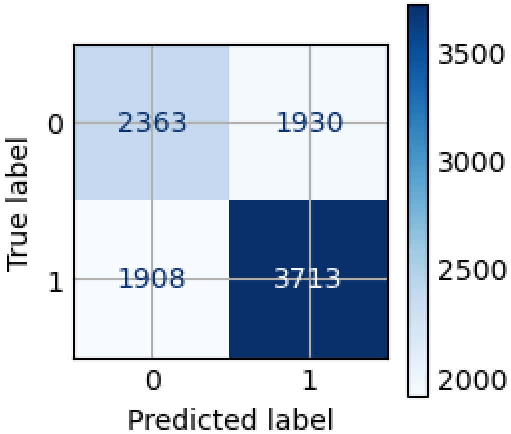
In [92]:
```
print("test")
print(f"accuracy_score{accuracy_score(y_test,y_test_predict_lr)},precision_s
```

```
test
accuracy_score0.6232351754739814,precision_score0.6275659824046921,recall_sc
ore0.7815924032140248
```

## Decision Tree

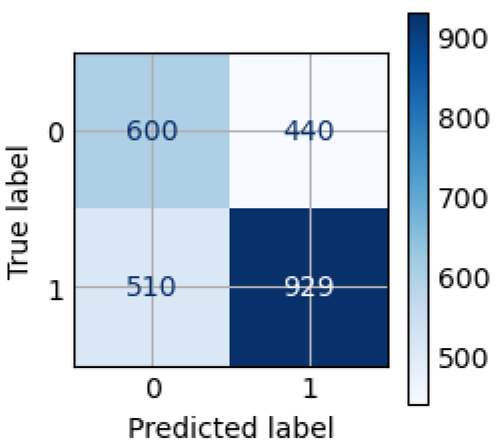
In [93]:
```
DTC =DecisionTreeClassifier(max_depth=2,criterion='entropy')
DTC.fit(x_train,y_train)

y_pred_train = DTC.predict(x_train)
cm = confusion_matrix(y_pred_train,y_train )

disp = ConfusionMatrixDisplay(confusion_matrix=cm)

print("recall acc for train : " , recall_score(y_pred_train,y_train))
print("precision for train : " ,precision_score(y_pred_train,y_train))
print("f1_score for train : " ,f1_score(y_pred_train,y_train))
print("accuracy : " ,accuracy_score(y_pred_train,y_train))
disp.plot(cmap='Blues')

plt.show()
```

```
recall acc for train :  0.6605586194627291
precision for train :  0.6579833421938686
f1_score for train :  0.6592684659090909
accuracy :  0.6128706879160782
```




In [94]:
```
y_pred_test = DTC.predict(x_test)
cm = confusion_matrix(y_pred_test,y_test )

disp = ConfusionMatrixDisplay(confusion_matrix=cm)

print("recall acc for test : " , recall_score(y_pred_test,y_test))
print("precision for test : " ,  precision_score(y_pred_test,y_test))
print("f1_score for test : " ,   f1_score(y_pred_test,y_test))
print("accuracy : " ,accuracy_score(y_pred_test,y_test))

disp.plot(cmap='Blues')

plt.show()
```

```
recall acc for test :  0.645587213342599
precision for test :  0.6785975164353543
f1_score for test :  0.6616809116809116
accuracy :  0.6167809600645422
```




## Random Forest Clasifier

In [112…
```
for i in range(5,105,5):
    rfc=RandomForestClassifier(n_estimators=i,criterion = 'entropy',max_dept
    rfc.fit(x_train,y_train)
    y_train_predict=rfc.predict(x_train)
    y_test_predict=rfc.predict(x_test)
```

```
    print("n_estimator =",i)
    print("train")
    print(f"accuracy_score={accuracy_score(y_train,y_train_predict)},precisi
    print("test")
    print(f"accuracy_score={accuracy_score(y_test,y_test_predict)},precision
    print("")
```

```
n_estimator = 5
train
accuracy_score=0.739560217873714,precision_score=0.7251728703839929,recall_s
core=0.873471557682084
test
accuracy_score=0.672045179507866,precision_score=0.6581342434584755,recall_s
core=0.8451424397370343

n_estimator = 10
train
accuracy_score=0.7479322170667743,precision_score=0.7252148997134671,recall_
score=0.8970405812511075
test
accuracy_score=0.6926179911254539,precision_score=0.670601461495222,recall_s
core=0.8714390065741418

n_estimator = 15
train
accuracy_score=0.7489408916683478,precision_score=0.7252857142857143,recall_
score=0.899698741804005
test
accuracy_score=0.6942315449778136,precision_score=0.6701949860724234,recall_
score=0.8787436084733382

n_estimator = 20
train
accuracy_score=0.751866048012911,precision_score=0.7276498355027893,recall_s
core=0.9014708488392699
test
accuracy_score=0.7006857603872529,precision_score=0.6778218944980148,recall_
score=0.872899926953981

n_estimator = 25
train
accuracy_score=0.7563042162598346,precision_score=0.7315917898665135,recall_
score=0.9032429558745348
test
accuracy_score=0.7027027027027027,precision_score=0.6801596351197263,recall_
score=0.8714390065741418

n_estimator = 30
train
accuracy_score=0.7568085535606214,precision_score=0.7320505456634119,recall_
score=0.9034201665780613
test
accuracy_score=0.6990722065348931,precision_score=0.6768881317433276,recall_
score=0.870708546384222

n_estimator = 35
train
accuracy_score=0.758119830542667,precision_score=0.7333525097080397,recall_s
core=0.9035973772815878
test
accuracy_score=0.7014925373134329,precision_score=0.6781869688385269,recall_
score=0.8743608473338204
```

```
n_estimator = 40
train
accuracy_score=0.7616501916481743,precision_score=0.7356999137683242,recall_
score=0.9071415913521177
test
accuracy_score=0.7039128680919726,precision_score=0.6805002842524162,recall_
score=0.8743608473338204

n_estimator = 45
train
accuracy_score=0.7615493241880169,precision_score=0.7364095169430426,recall_
score=0.9050150629097997
test
accuracy_score=0.7035094796288827,precision_score=0.6790960451977401,recall_
score=0.8780131482834186

n_estimator = 50
train
accuracy_score=0.7610449868872302,precision_score=0.7350660539919587,recall_
score=0.9071415913521177
test
accuracy_score=0.7022993142396128,precision_score=0.6781479390175043,recall_
score=0.8772826880934989

n_estimator = 55
train
accuracy_score=0.7631632035505346,precision_score=0.7369480799654825,recall_
score=0.9080276448697502
test
accuracy_score=0.7022993142396128,precision_score=0.6785512167515563,recall_
score=0.8758217677136596

n_estimator = 60
train
accuracy_score=0.7631632035505346,precision_score=0.7374261420954028,recall_
score=0.9067871699450647
test
accuracy_score=0.7067365873336022,precision_score=0.6821793416572077,recall_
score=0.8780131482834186

n_estimator = 65
train
accuracy_score=0.763566673391164,precision_score=0.736826992103374,recall_sc
ore=0.9094453304979621
test
accuracy_score=0.7063331988705123,precision_score=0.6809712027103332,recall_
score=0.8809349890430972

n_estimator = 70
train
accuracy_score=0.7645753479927375,precision_score=0.7387790445951797,recall_
score=0.9071415913521177
test
accuracy_score=0.709156918112142,precision_score=0.6840909090909091,recall_s
core=0.8794740686632578
```

```
n_estimator = 75
train
accuracy_score=0.7645753479927375,precision_score=0.7385036759406083,recall_
score=0.9078504341662237
test
accuracy_score=0.7075433642597821,precision_score=0.6829545454545455,recall_
score=0.8780131482834186

n_estimator = 80
train
accuracy_score=0.7638692757716361,precision_score=0.7373490511788384,recall_
score=0.9089136983873826
test
accuracy_score=0.7043162565550625,precision_score=0.6796610169491526,recall_
score=0.8787436084733382

n_estimator = 85
train
accuracy_score=0.7657857575146257,precision_score=0.7395758187851681,recall_
score=0.9083820662768031
test
accuracy_score=0.7031060911657927,precision_score=0.67871259175607,recall_sc
ore=0.8780131482834186

n_estimator = 90
train
accuracy_score=0.7656848900544684,precision_score=0.7393310265282583,recall_
score=0.9087364876838561
test
accuracy_score=0.7043162565550625,precision_score=0.6794582392776524,recall_
score=0.8794740686632578

n_estimator = 95
train
accuracy_score=0.765281420213839,precision_score=0.7389737676563851,recall_s
core=0.9085592769803296
test
accuracy_score=0.7059298104074223,precision_score=0.6799775028121485,recall_
score=0.8831263696128561

n_estimator = 100
train
accuracy_score=0.7665926971958846,precision_score=0.7402222542935488,recall_
score=0.9089136983873826
test
accuracy_score=0.7043162565550625,precision_score=0.6786516853932584,recall_
score=0.8823959094229364
```

In [113…

```
for i in range(5,105,5):
    rfc=RandomForestClassifier(n_estimators=i,criterion = 'gini',max_depth=7
    rfc.fit(x_train,y_train)
    y_train_predict=rfc.predict(x_train)
    y_test_predict=rfc.predict(x_test)
    print("n_estimator =",i)
    print("train")
```

```
    print(f"accuracy_score={accuracy_score(y_train,y_train_predict)},precisi
    print("test")
    print(f"accuracy_score={accuracy_score(y_test,y_test_predict)},precision
    print("")
```

```
n_estimator = 5
train
accuracy_score=0.711014726649183,precision_score=0.700201787258576,recall_sc
ore=0.860889597731703
test
accuracy_score=0.6692214602662364,precision_score=0.6583958453548759,recall_
score=0.83345507669832

n_estimator = 10
train
accuracy_score=0.7218075448860197,precision_score=0.7017764722338788,recall_
score=0.8890660995924153
test
accuracy_score=0.6829366680112948,precision_score=0.6627582356225572,recall_
score=0.8670562454346238

n_estimator = 15
train
accuracy_score=0.7241274964696389,precision_score=0.7024505708716235,recall_
score=0.8940279992911572
test
accuracy_score=0.6797095603065753,precision_score=0.6584022038567493,recall_
score=0.872899926953981

n_estimator = 20
train
accuracy_score=0.7276578575751462,precision_score=0.706700379266751,recall_s
core=0.8915470494417863
test
accuracy_score=0.6873739411052844,precision_score=0.6639072847682119,recall_
score=0.8787436084733382

n_estimator = 25
train
accuracy_score=0.7297760742384507,precision_score=0.7072727272727273,recall_
score=0.8961545277334751
test
accuracy_score=0.6793061718434853,precision_score=0.6583885209713024,recall_
score=0.8714390065741418

n_estimator = 30
train
accuracy_score=0.7322977607423845,precision_score=0.7094017094017094,recall_
score=0.897217791954634
test
accuracy_score=0.6873739411052844,precision_score=0.6640883977900552,recall_
score=0.8780131482834186

n_estimator = 35
train
accuracy_score=0.7300786766189228,precision_score=0.7075696096264167,recall_
score=0.8961545277334751
test
accuracy_score=0.6877773295683743,precision_score=0.6642738818332413,recall_
score=0.8787436084733382
```

```
n_estimator = 40
train
accuracy_score=0.7317934234415977,precision_score=0.708905068608233,recall_s
core=0.897217791954634
test
accuracy_score=0.691407825736184,precision_score=0.6666666666666666,recall_s
core=0.8823959094229364

n_estimator = 45
train
accuracy_score=0.7317934234415977,precision_score=0.7093153759820426,recall_
score=0.8959773170299486
test
accuracy_score=0.691407825736184,precision_score=0.667220376522702,recall_sc
ore=0.8802045288531775

n_estimator = 50
train
accuracy_score=0.729574339318136,precision_score=0.7070170533967012,recall_s
core=0.8963317384370016
test
accuracy_score=0.6897942718838241,precision_score=0.665380374862183,recall_s
core=0.8816654492330168

n_estimator = 55
train
accuracy_score=0.7306838813798668,precision_score=0.7083975886723679,recall_
score=0.8954456849193692
test
accuracy_score=0.6922146026623639,precision_score=0.6663007683863886,recall_
score=0.8867786705624543

n_estimator = 60
train
accuracy_score=0.7330038329634859,precision_score=0.7108077680833099,recall_
score=0.8950912635123162
test
accuracy_score=0.6938281565147236,precision_score=0.6686946902654868,recall_
score=0.8831263696128561

n_estimator = 65
train
accuracy_score=0.7332055678838006,precision_score=0.7107112735451223,recall_
score=0.8959773170299486
test
accuracy_score=0.6938281565147236,precision_score=0.6688815060908084,recall_
score=0.8823959094229364

n_estimator = 70
train
accuracy_score=0.733306435343958,precision_score=0.7110485573539761,recall_s
core=0.8952684742158427
test
accuracy_score=0.6970552642194433,precision_score=0.6710963455149501,recall_
score=0.885317750182615
```

```
n_estimator = 75
train
accuracy_score=0.735021182166633,precision_score=0.7125740062024246,recall_s
core=0.8958001063264222
test
accuracy_score=0.6982654296087132,precision_score=0.672787979966611,recall_s
core=0.8831263696128561

n_estimator = 80
train
accuracy_score=0.7361307242283639,precision_score=0.7134395712875476,recall_
score=0.896508949140528
test
accuracy_score=0.6962484872932634,precision_score=0.6709211986681465,recall_
score=0.8831263696128561

n_estimator = 85
train
accuracy_score=0.7357272543877346,precision_score=0.7135792002260845,recall_
score=0.8949140528087897
test
accuracy_score=0.6974586526825333,precision_score=0.6716583471991125,recall_
score=0.8845872899926954

n_estimator = 90
train
accuracy_score=0.734415977405689,precision_score=0.7115546809108799,recall_s
core=0.8970405812511075
test
accuracy_score=0.6954417103670835,precision_score=0.6703662597114317,recall_
score=0.8823959094229364

n_estimator = 95
train
accuracy_score=0.7341133750252169,precision_score=0.711136076393765,recall_s
core=0.8973950026581605
test
accuracy_score=0.6954417103670835,precision_score=0.6711259754738016,recall_
score=0.8794740686632578

n_estimator = 100
train
accuracy_score=0.7343151099455316,precision_score=0.7113952508079247,recall_
score=0.897217791954634
test
accuracy_score=0.6966518757563533,precision_score=0.6720580033463469,recall_
score=0.8802045288531775
```

Note: rfc=RandomForestClassifier(n_estimators=70,criterion = 'entropy',max_depth=8,random_state=42)

In [116…

```
rfc=RandomForestClassifier(n_estimators=70,criterion = 'entropy',max_depth=8
rfc.fit(x_train,y_train)
y_train_predict=rfc.predict(x_train)
```

```
y_test_predict=rfc.predict(x_test)
print("n_estimator =",70)
print("train")
print(f"accuracy_score={accuracy_score(y_train,y_train_predict)},precision_s
print("test")
print(f"accuracy_score={accuracy_score(y_test,y_test_predict)},precision_sco
print("")
```

```
n_estimator = 70
train
accuracy_score=0.7645753479927375,precision_score=0.7387790445951797,recall_
score=0.9071415913521177
test
accuracy_score=0.709156918112142,precision_score=0.6840909090909091,recall_s
core=0.8794740686632578
```

In [118…

```
pip install xgboost
```

```
Collecting xgboostNote: you may need to restart the kernel to use updated pa
ckages.

  Downloading xgboost-2.1.3-py3-none-win_amd64.whl.metadata (2.1 kB)
Requirement already satisfied: numpy in c:\users\welcome\anaconda3\lib\site-
packages (from xgboost) (1.26.4)
Requirement already satisfied: scipy in c:\users\welcome\anaconda3\lib\site-
packages (from xgboost) (1.11.4)
Downloading xgboost-2.1.3-py3-none-win_amd64.whl (124.9 MB)
   ---------------------------------------- 0.0/124.9 MB ? eta -:--:--
   ---------------------------------------- 0.0/124.9 MB ? eta -:--:--
   ---------------------------------------- 0.0/124.9 MB ? eta -:--:--
   ---------------------------------------- 0.0/124.9 MB 393.8 kB/s eta 0:0
5:18
   ---------------------------------------- 0.2/124.9 MB 1.3 MB/s eta 0:01:3
6
   ---------------------------------------- 0.4/124.9 MB 2.2 MB/s eta 0:00:5
7
   ---------------------------------------- 0.4/124.9 MB 2.2 MB/s eta 0:00:5
6
   ---------------------------------------- 0.5/124.9 MB 2.0 MB/s eta 0:01:0
4
   ---------------------------------------- 0.7/124.9 MB 2.1 MB/s eta 0:01:0
0
   ---------------------------------------- 0.8/124.9 MB 2.3 MB/s eta 0:00:5
4
   ---------------------------------------- 0.9/124.9 MB 2.3 MB/s eta 0:00:5
5
   ---------------------------------------- 0.9/124.9 MB 2.2 MB/s eta 0:00:5
6
   ---------------------------------------- 1.0/124.9 MB 2.2 MB/s eta 0:00:5
8
   ---------------------------------------- 1.1/124.9 MB 2.2 MB/s eta 0:00:5
7
   ---------------------------------------- 1.1/124.9 MB 2.1 MB/s eta 0:00:5
9
   ---------------------------------------- 1.2/124.9 MB 2.1 MB/s eta 0:00:5
9
   ---------------------------------------- 1.2/124.9 MB 2.0 MB/s eta 0:01:0
3
   ---------------------------------------- 1.3/124.9 MB 2.0 MB/s eta 0:01:0
2
   ---------------------------------------- 1.4/124.9 MB 1.9 MB/s eta 0:01:0
5
   ---------------------------------------- 1.5/124.9 MB 2.0 MB/s eta 0:01:0
3
   ---------------------------------------- 1.5/124.9 MB 1.9 MB/s eta 0:01:0
5
    --------------------------------------- 1.6/124.9 MB 1.9 MB/s eta 0:01:0
5
    --------------------------------------- 1.7/124.9 MB 1.9 MB/s eta 0:01:0
6
    --------------------------------------- 1.7/124.9 MB 1.9 MB/s eta 0:01:0
6
    --------------------------------------- 1.9/124.9 MB 2.0 MB/s eta 0:01:0
3
```

```
    ---------------------------------------- 2.0/124.9 MB 1.9 MB/s eta 0:01:0
4
    ---------------------------------------- 2.1/124.9 MB 2.0 MB/s eta 0:01:0
3
    ---------------------------------------- 2.2/124.9 MB 2.0 MB/s eta 0:01:0
3
    ---------------------------------------- 2.3/124.9 MB 2.0 MB/s eta 0:01:0
2
    ---------------------------------------- 2.3/124.9 MB 2.0 MB/s eta 0:01:0
3
    ---------------------------------------- 2.4/124.9 MB 2.0 MB/s eta 0:01:0
3
    ---------------------------------------- 2.5/124.9 MB 1.9 MB/s eta 0:01:0
3
    ---------------------------------------- 2.6/124.9 MB 2.0 MB/s eta 0:01:0
3
    ---------------------------------------- 2.7/124.9 MB 2.0 MB/s eta 0:01:0
3
    ---------------------------------------- 2.8/124.9 MB 2.0 MB/s eta 0:01:0
1
    ---------------------------------------- 3.0/124.9 MB 2.1 MB/s eta 0:01:0
0
   - --------------------------------------- 3.2/124.9 MB 2.1 MB/s eta 0:00:5
8
   - --------------------------------------- 3.4/124.9 MB 2.2 MB/s eta 0:00:5
6
   - --------------------------------------- 3.5/124.9 MB 2.2 MB/s eta 0:00:5
5
   - --------------------------------------- 3.7/124.9 MB 2.3 MB/s eta 0:00:5
3
   - --------------------------------------- 3.8/124.9 MB 2.3 MB/s eta 0:00:5
3
   - --------------------------------------- 4.0/124.9 MB 2.4 MB/s eta 0:00:5
2
   - --------------------------------------- 4.2/124.9 MB 2.4 MB/s eta 0:00:5
1
   - --------------------------------------- 4.4/124.9 MB 2.4 MB/s eta 0:00:5
0
   - --------------------------------------- 4.6/124.9 MB 2.5 MB/s eta 0:00:4
9
   - --------------------------------------- 4.7/124.9 MB 2.5 MB/s eta 0:00:4
8
   - --------------------------------------- 4.8/124.9 MB 2.5 MB/s eta 0:00:4
8
   - --------------------------------------- 5.0/124.9 MB 2.5 MB/s eta 0:00:4
8
   - --------------------------------------- 5.2/124.9 MB 2.6 MB/s eta 0:00:4
7
   - --------------------------------------- 5.4/124.9 MB 2.6 MB/s eta 0:00:4
6
   - --------------------------------------- 5.5/124.9 MB 2.7 MB/s eta 0:00:4
5
   - --------------------------------------- 5.8/124.9 MB 2.7 MB/s eta 0:00:4
5
   - --------------------------------------- 6.0/124.9 MB 2.7 MB/s eta 0:00:4
4
```

```
   - -------------------------------------- 6.2/124.9 MB 2.8 MB/s eta 0:00:4
4
   -- ------------------------------------- 6.4/124.9 MB 2.8 MB/s eta 0:00:4
3
   -- ------------------------------------- 6.6/124.9 MB 2.8 MB/s eta 0:00:4
2
   -- ------------------------------------- 6.9/124.9 MB 2.9 MB/s eta 0:00:4
1
   -- ------------------------------------- 7.1/124.9 MB 2.9 MB/s eta 0:00:4
1
   -- ------------------------------------- 7.5/124.9 MB 3.0 MB/s eta 0:00:3
9
   -- ------------------------------------- 7.9/124.9 MB 3.1 MB/s eta 0:00:3
8
   -- ------------------------------------- 8.3/124.9 MB 3.2 MB/s eta 0:00:3
6
   -- ------------------------------------- 8.6/124.9 MB 3.3 MB/s eta 0:00:3
6
   -- ------------------------------------- 9.2/124.9 MB 3.4 MB/s eta 0:00:3
4
   --- ------------------------------------ 9.6/124.9 MB 3.5 MB/s eta 0:00:3
3
   --- ------------------------------------ 9.9/124.9 MB 3.6 MB/s eta 0:00:3
2
   --- ------------------------------------ 10.3/124.9 MB 3.8 MB/s eta 0:00:
31
   --- ------------------------------------ 10.5/124.9 MB 3.8 MB/s eta 0:00:
31
   --- ------------------------------------ 10.8/124.9 MB 3.9 MB/s eta 0:00:
29
   --- ------------------------------------ 11.1/124.9 MB 4.0 MB/s eta 0:00:
29
   --- ------------------------------------ 11.4/124.9 MB 4.2 MB/s eta 0:00:
28
   --- ------------------------------------ 12.2/124.9 MB 5.0 MB/s eta 0:00:
23
   ---- ----------------------------------- 12.8/124.9 MB 5.8 MB/s eta 0:00:
20
   ---- ----------------------------------- 13.5/124.9 MB 6.4 MB/s eta 0:00:
18
   ---- ----------------------------------- 14.2/124.9 MB 7.0 MB/s eta 0:00:
16
   ---- ----------------------------------- 14.8/124.9 MB 7.7 MB/s eta 0:00:
15
   ----- ---------------------------------- 15.6/124.9 MB 8.7 MB/s eta 0:00:
13
   ----- ---------------------------------- 16.5/124.9 MB 10.2 MB/s eta 0:0
0:11
   ----- ---------------------------------- 17.3/124.9 MB 11.5 MB/s eta 0:0
0:10
   ----- ---------------------------------- 18.1/124.9 MB 12.4 MB/s eta 0:0
0:09
   ----- ---------------------------------- 18.6/124.9 MB 12.8 MB/s eta 0:0
0:09
   ------ --------------------------------- 19.0/124.9 MB 13.4 MB/s eta 0:0
0:08
```

```
   ------ --------------------------------- 19.0/124.9 MB 12.8 MB/s eta 0:0
0:09
   ------ --------------------------------- 19.9/124.9 MB 13.1 MB/s eta 0:0
0:09
   ------ --------------------------------- 20.7/124.9 MB 14.6 MB/s eta 0:0
0:08
   ------ --------------------------------- 21.7/124.9 MB 17.2 MB/s eta 0:0
0:06
   ------- -------------------------------- 22.5/124.9 MB 17.2 MB/s eta 0:0
0:06
   ------- -------------------------------- 23.1/124.9 MB 16.4 MB/s eta 0:0
0:07
   ------- -------------------------------- 23.7/124.9 MB 16.0 MB/s eta 0:0
0:07
   ------- -------------------------------- 24.0/124.9 MB 15.2 MB/s eta 0:0
0:07
   ------- -------------------------------- 24.1/124.9 MB 14.9 MB/s eta 0:0
0:07
   ------- -------------------------------- 24.3/124.9 MB 14.2 MB/s eta 0:0
0:08
   ------- -------------------------------- 24.7/124.9 MB 13.4 MB/s eta 0:0
0:08
   -------- ------------------------------- 25.0/124.9 MB 13.4 MB/s eta 0:0
0:08
   -------- ------------------------------- 25.5/124.9 MB 12.6 MB/s eta 0:0
0:08
   -------- ------------------------------- 26.0/124.9 MB 12.1 MB/s eta 0:0
0:09
   -------- ------------------------------- 26.5/124.9 MB 11.9 MB/s eta 0:0
0:09
   -------- ------------------------------- 26.9/124.9 MB 11.5 MB/s eta 0:0
0:09
   -------- ------------------------------- 27.6/124.9 MB 11.1 MB/s eta 0:0
0:09
   -------- ------------------------------- 28.0/124.9 MB 10.6 MB/s eta 0:0
0:10
   --------- ------------------------------ 28.4/124.9 MB 10.4 MB/s eta 0:0
0:10
   --------- ------------------------------ 28.5/124.9 MB 9.9 MB/s eta 0:00:
10
   --------- ------------------------------ 28.7/124.9 MB 9.6 MB/s eta 0:00:
10
   --------- ------------------------------ 29.0/124.9 MB 9.4 MB/s eta 0:00:
11
   --------- ------------------------------ 29.2/124.9 MB 9.2 MB/s eta 0:00:
11
   --------- ------------------------------ 29.4/124.9 MB 9.6 MB/s eta 0:00:
10
   --------- ------------------------------ 29.5/124.9 MB 9.4 MB/s eta 0:00:
11
   --------- ------------------------------ 29.7/124.9 MB 9.1 MB/s eta 0:00:
11
   --------- ------------------------------ 29.9/124.9 MB 8.8 MB/s eta 0:00:
11
   --------- ------------------------------ 30.1/124.9 MB 8.6 MB/s eta 0:00:
12
```

```
   --------- ------------------------------ 30.4/124.9 MB 8.4 MB/s eta 0:00:
12
   --------- ------------------------------ 30.6/124.9 MB 8.3 MB/s eta 0:00:
12
   --------- ------------------------------ 30.8/124.9 MB 8.0 MB/s eta 0:00:
12
   --------- ------------------------------ 31.1/124.9 MB 7.9 MB/s eta 0:00:
12
   ---------- ----------------------------- 31.3/124.9 MB 7.7 MB/s eta 0:00:
13
   ---------- ----------------------------- 31.5/124.9 MB 7.5 MB/s eta 0:00:
13
   ---------- ----------------------------- 31.8/124.9 MB 7.4 MB/s eta 0:00:
13
   ---------- ----------------------------- 32.2/124.9 MB 7.4 MB/s eta 0:00:
13
   ---------- ----------------------------- 32.7/124.9 MB 7.3 MB/s eta 0:00:
13
   ---------- ----------------------------- 33.1/124.9 MB 7.3 MB/s eta 0:00:
13
   ---------- ----------------------------- 33.9/124.9 MB 7.4 MB/s eta 0:00:
13
   ----------- ---------------------------- 34.4/124.9 MB 7.7 MB/s eta 0:00:
12
   ----------- ---------------------------- 35.2/124.9 MB 8.0 MB/s eta 0:00:
12
   ----------- ---------------------------- 35.7/124.9 MB 8.2 MB/s eta 0:00:
11
   ----------- ---------------------------- 36.1/124.9 MB 8.3 MB/s eta 0:00:
11
   ----------- ---------------------------- 36.5/124.9 MB 8.2 MB/s eta 0:00:
11
   ----------- ---------------------------- 36.8/124.9 MB 8.2 MB/s eta 0:00:
11
   ----------- ---------------------------- 37.2/124.9 MB 8.4 MB/s eta 0:00:
11
   ------------ --------------------------- 38.0/124.9 MB 8.2 MB/s eta 0:00:
11
   ------------ --------------------------- 38.3/124.9 MB 8.3 MB/s eta 0:00:
11
   ------------ --------------------------- 38.8/124.9 MB 8.6 MB/s eta 0:00:
10
   ------------ --------------------------- 39.1/124.9 MB 9.0 MB/s eta 0:00:
10
   ------------ --------------------------- 39.3/124.9 MB 8.8 MB/s eta 0:00:
10
   ------------ --------------------------- 39.6/124.9 MB 8.7 MB/s eta 0:00:
10
   ------------ --------------------------- 40.3/124.9 MB 9.5 MB/s eta 0:00:
09
   ------------- -------------------------- 41.4/124.9 MB 10.6 MB/s eta 0:0
0:08
   ------------- -------------------------- 42.2/124.9 MB 11.5 MB/s eta 0:0
0:08
   ------------- -------------------------- 42.8/124.9 MB 11.5 MB/s eta 0:0
0:08
```

```
   ------------- -------------------------- 43.6/124.9 MB 11.7 MB/s eta 0:0
0:07
   -------------- ------------------------- 44.1/124.9 MB 11.7 MB/s eta 0:0
0:07
   -------------- ------------------------- 44.6/124.9 MB 11.9 MB/s eta 0:0
0:07
   -------------- ------------------------- 45.2/124.9 MB 11.9 MB/s eta 0:0
0:07
   -------------- ------------------------- 45.9/124.9 MB 11.9 MB/s eta 0:0
0:07
   -------------- ------------------------- 46.4/124.9 MB 11.9 MB/s eta 0:0
0:07
   --------------- ------------------------ 46.9/124.9 MB 12.4 MB/s eta 0:0
0:07
   --------------- ------------------------ 47.7/124.9 MB 13.1 MB/s eta 0:0
0:06
   --------------- ------------------------ 48.4/124.9 MB 13.1 MB/s eta 0:0
0:06
   --------------- ------------------------ 48.9/124.9 MB 13.4 MB/s eta 0:0
0:06
   --------------- ------------------------ 49.4/124.9 MB 13.9 MB/s eta 0:0
0:06
   ---------------- ----------------------- 50.0/124.9 MB 14.9 MB/s eta 0:0
0:06
   ---------------- ----------------------- 50.9/124.9 MB 14.9 MB/s eta 0:0
0:05
   ---------------- ----------------------- 51.6/124.9 MB 14.9 MB/s eta 0:0
0:05
   ---------------- ----------------------- 52.1/124.9 MB 14.9 MB/s eta 0:0
0:05
   ----------------- ---------------------- 53.2/124.9 MB 15.6 MB/s eta 0:0
0:05
   ----------------- ---------------------- 53.8/124.9 MB 15.6 MB/s eta 0:0
0:05
   ----------------- ---------------------- 54.4/124.9 MB 15.6 MB/s eta 0:0
0:05
   ----------------- ---------------------- 55.0/124.9 MB 15.6 MB/s eta 0:0
0:05
   ----------------- ---------------------- 55.8/124.9 MB 15.6 MB/s eta 0:0
0:05
   ------------------ --------------------- 56.3/124.9 MB 15.6 MB/s eta 0:0
0:05
   ------------------ --------------------- 56.9/124.9 MB 15.6 MB/s eta 0:0
0:05
   ------------------ --------------------- 57.5/124.9 MB 15.6 MB/s eta 0:0
0:05
   ------------------ --------------------- 58.1/124.9 MB 16.0 MB/s eta 0:0
0:05
   ------------------ --------------------- 58.8/124.9 MB 16.0 MB/s eta 0:0
0:05
   ------------------- -------------------- 59.4/124.9 MB 16.0 MB/s eta 0:0
0:05
   ------------------- -------------------- 60.0/124.9 MB 16.4 MB/s eta 0:0
0:04
   ------------------- -------------------- 60.4/124.9 MB 16.4 MB/s eta 0:0
0:04
```

```
   ------------------- -------------------- 61.1/124.9 MB 15.6 MB/s eta 0:0
0:05
   ------------------- -------------------- 61.8/124.9 MB 16.0 MB/s eta 0:0
0:04
   -------------------- ------------------- 62.8/124.9 MB 16.4 MB/s eta 0:0
0:04
   -------------------- ------------------- 63.5/124.9 MB 16.8 MB/s eta 0:0
0:04
   -------------------- ------------------- 65.0/124.9 MB 17.7 MB/s eta 0:0
0:04
   --------------------- ------------------ 65.7/124.9 MB 17.7 MB/s eta 0:0
0:04
   --------------------- ------------------ 66.2/124.9 MB 17.7 MB/s eta 0:0
0:04
   --------------------- ------------------ 66.7/124.9 MB 17.3 MB/s eta 0:0
0:04
   --------------------- ------------------ 67.6/124.9 MB 17.7 MB/s eta 0:0
0:04
   --------------------- ------------------ 68.6/124.9 MB 17.7 MB/s eta 0:0
0:04
   ---------------------- ----------------- 69.5/124.9 MB 18.7 MB/s eta 0:0
0:03
   ---------------------- ----------------- 70.7/124.9 MB 19.8 MB/s eta 0:0
0:03
   ---------------------- ----------------- 71.6/124.9 MB 20.5 MB/s eta 0:0
0:03
   ----------------------- ---------------- 72.5/124.9 MB 20.5 MB/s eta 0:0
0:03
   ----------------------- ---------------- 73.1/124.9 MB 19.9 MB/s eta 0:0
0:03
   ----------------------- ---------------- 73.7/124.9 MB 18.7 MB/s eta 0:0
0:03
   ----------------------- ---------------- 74.0/124.9 MB 18.7 MB/s eta 0:0
0:03
   ----------------------- ---------------- 74.4/124.9 MB 18.2 MB/s eta 0:0
0:03
   ----------------------- ---------------- 74.8/124.9 MB 16.8 MB/s eta 0:0
0:03
   ------------------------ --------------- 75.3/124.9 MB 16.4 MB/s eta 0:0
0:04
   ------------------------ --------------- 75.9/124.9 MB 16.0 MB/s eta 0:0
0:04
   ------------------------ --------------- 76.6/124.9 MB 16.8 MB/s eta 0:0
0:03
   ------------------------ --------------- 77.2/124.9 MB 16.8 MB/s eta 0:0
0:03
   ------------------------ --------------- 77.9/124.9 MB 16.8 MB/s eta 0:0
0:03
   ------------------------- -------------- 78.5/124.9 MB 16.4 MB/s eta 0:0
0:03
   ------------------------- -------------- 79.2/124.9 MB 16.4 MB/s eta 0:0
0:03
   ------------------------- -------------- 79.8/124.9 MB 16.4 MB/s eta 0:0
0:03
   ------------------------- -------------- 80.4/124.9 MB 15.6 MB/s eta 0:0
0:03
```

```
   ------------------------- -------------- 81.1/124.9 MB 15.6 MB/s eta 0:0
0:03
   -------------------------- ------------- 81.8/124.9 MB 15.6 MB/s eta 0:0
0:03
   -------------------------- ------------- 82.4/124.9 MB 15.2 MB/s eta 0:0
0:03
   -------------------------- ------------- 83.1/124.9 MB 14.9 MB/s eta 0:0
0:03
   -------------------------- ------------- 83.8/124.9 MB 15.6 MB/s eta 0:0
0:03
   -------------------------- ------------- 84.1/124.9 MB 15.6 MB/s eta 0:0
0:03
   --------------------------- ------------ 84.8/124.9 MB 16.0 MB/s eta 0:0
0:03
   --------------------------- ------------ 85.4/124.9 MB 16.8 MB/s eta 0:0
0:03
   --------------------------- ------------ 86.1/124.9 MB 16.8 MB/s eta 0:0
0:03
   --------------------------- ------------ 86.8/124.9 MB 16.8 MB/s eta 0:0
0:03
   --------------------------- ------------ 87.2/124.9 MB 16.4 MB/s eta 0:0
0:03
   ---------------------------- ----------- 87.8/124.9 MB 16.4 MB/s eta 0:0
0:03
   ---------------------------- ----------- 88.8/124.9 MB 16.4 MB/s eta 0:0
0:03
   ---------------------------- ----------- 89.5/124.9 MB 16.0 MB/s eta 0:0
0:03
   ---------------------------- ----------- 90.0/124.9 MB 16.0 MB/s eta 0:0
0:03
   ----------------------------- ---------- 90.7/124.9 MB 16.4 MB/s eta 0:0
0:03
   ----------------------------- ---------- 91.4/124.9 MB 16.0 MB/s eta 0:0
0:03
   ----------------------------- ---------- 92.1/124.9 MB 16.4 MB/s eta 0:0
0:03
   ----------------------------- ---------- 92.7/124.9 MB 16.4 MB/s eta 0:0
0:02
   ----------------------------- ---------- 93.4/124.9 MB 16.4 MB/s eta 0:0
0:02
   ------------------------------ --------- 94.1/124.9 MB 16.8 MB/s eta 0:0
0:02
   ------------------------------ --------- 94.8/124.9 MB 17.2 MB/s eta 0:0
0:02
   ------------------------------ --------- 95.5/124.9 MB 17.3 MB/s eta 0:0
0:02
   ------------------------------ --------- 96.2/124.9 MB 17.2 MB/s eta 0:0
0:02
   ------------------------------- -------- 96.9/124.9 MB 17.2 MB/s eta 0:0
0:02
   ------------------------------- -------- 97.6/124.9 MB 18.2 MB/s eta 0:0
0:02
   ------------------------------- -------- 98.3/124.9 MB 18.7 MB/s eta 0:0
0:02
   ------------------------------- -------- 98.9/124.9 MB 18.2 MB/s eta 0:0
0:02
```

```
   ------------------------------- -------- 99.5/124.9 MB 18.2 MB/s eta 0:0
0:02
   ------------------------------- ------- 100.2/124.9 MB 18.7 MB/s eta 0:0
0:02
   ------------------------------- ------- 100.9/124.9 MB 18.2 MB/s eta 0:0
0:02
   ------------------------------- ------- 101.7/124.9 MB 18.7 MB/s eta 0:0
0:02
   ------------------------------- ------- 102.5/124.9 MB 18.7 MB/s eta 0:0
0:02
   -------------------------------- ------ 103.2/124.9 MB 18.7 MB/s eta 0:0
0:02
   -------------------------------- ------ 103.9/124.9 MB 19.3 MB/s eta 0:0
0:02
   -------------------------------- ------ 104.6/124.9 MB 18.7 MB/s eta 0:0
0:02
   -------------------------------- ------ 105.3/124.9 MB 18.7 MB/s eta 0:0
0:02
   --------------------------------- ----- 105.8/124.9 MB 18.7 MB/s eta 0:0
0:02
   --------------------------------- ----- 106.4/124.9 MB 18.2 MB/s eta 0:0
0:02
   --------------------------------- ----- 107.0/124.9 MB 18.2 MB/s eta 0:0
0:01
   --------------------------------- ----- 107.6/124.9 MB 17.7 MB/s eta 0:0
0:01
   --------------------------------- ----- 108.2/124.9 MB 17.7 MB/s eta 0:0
0:01
   --------------------------------- ----- 108.7/124.9 MB 17.3 MB/s eta 0:0
0:01
   ---------------------------------- ---- 109.2/124.9 MB 17.2 MB/s eta 0:0
0:01
   ---------------------------------- ---- 109.6/124.9 MB 16.8 MB/s eta 0:0
0:01
   ---------------------------------- ---- 110.0/124.9 MB 16.4 MB/s eta 0:0
0:01
   ---------------------------------- ---- 110.7/124.9 MB 16.4 MB/s eta 0:0
0:01
   ---------------------------------- ---- 111.3/124.9 MB 16.4 MB/s eta 0:0
0:01
   ---------------------------------- ---- 111.3/124.9 MB 16.4 MB/s eta 0:0
0:01
   ---------------------------------- ---- 111.9/124.9 MB 15.2 MB/s eta 0:0
0:01
   ----------------------------------- --- 112.5/124.9 MB 14.9 MB/s eta 0:0
0:01
   ----------------------------------- --- 113.3/124.9 MB 15.2 MB/s eta 0:0
0:01
   ----------------------------------- --- 114.1/124.9 MB 14.9 MB/s eta 0:0
0:01
   ----------------------------------- --- 114.8/124.9 MB 14.9 MB/s eta 0:0
0:01
   ------------------------------------ -- 116.0/124.9 MB 16.4 MB/s eta 0:0
0:01
   ------------------------------------ -- 117.5/124.9 MB 17.7 MB/s eta 0:0
0:01
```

```
   ------------------------------------ -- 118.2/124.9 MB 17.7 MB/s eta 0:0
0:01
   ------------------------------------- - 118.8/124.9 MB 17.7 MB/s eta 0:0
0:01
   ------------------------------------- - 118.8/124.9 MB 17.7 MB/s eta 0:0
0:01
   ------------------------------------- - 119.1/124.9 MB 16.4 MB/s eta 0:0
0:01
   ------------------------------------- - 119.9/124.9 MB 17.3 MB/s eta 0:0
0:01
   ------------------------------------- - 121.0/124.9 MB 18.2 MB/s eta 0:0
0:01
   --------------------------------------  122.6/124.9 MB 21.9 MB/s eta 0:0
0:01
   --------------------------------------  123.4/124.9 MB 22.6 MB/s eta 0:0
0:01
   --------------------------------------  124.1/124.9 MB 22.6 MB/s eta 0:0
0:01
   --------------------------------------  124.8/124.9 MB 21.8 MB/s eta 0:0
0:01
   --------------------------------------  124.8/124.9 MB 21.8 MB/s eta 0:0
0:01
   --------------------------------------  124.8/124.9 MB 21.8 MB/s eta 0:0
0:01
   --------------------------------------  124.8/124.9 MB 21.8 MB/s eta 0:0
0:01
   --------------------------------------  124.8/124.9 MB 21.8 MB/s eta 0:0
0:01
   --------------------------------------  124.8/124.9 MB 21.8 MB/s eta 0:0
0:01
   --------------------------------------  124.8/124.9 MB 14.9 MB/s eta 0:0
0:01
   --------------------------------------  124.9/124.9 MB 14.6 MB/s eta 0:0
0:01
   --------------------------------------  124.9/124.9 MB 14.2 MB/s eta 0:0
0:01
   --------------------------------------  124.9/124.9 MB 14.2 MB/s eta 0:0
0:01
   --------------------------------------  124.9/124.9 MB 14.2 MB/s eta 0:0
0:01
   --------------------------------------  124.9/124.9 MB 14.2 MB/s eta 0:0
0:01
   --------------------------------------  124.9/124.9 MB 14.2 MB/s eta 0:0
0:01
   --------------------------------------- 124.9/124.9 MB 11.1 MB/s eta 0:0
0:00
Installing collected packages: xgboost
Successfully installed xgboost-2.1.3
```

In [119…

```python
from xgboost import XGBClassifier
```

# XGBoost Classifier

In [120…

```python
from xgboost import XGBClassifier
```

```
xgb_model = XGBClassifier(use_label_encoder=False, eval_metric='logloss')
xgb_model.fit(x_train, y_train)
y_train_predict = xgb_model.predict(x_train)
# Predict
y_test_predict = xgb_model.predict(x_test)

# Evaluate
print("train")
print(f"accuracy_score={accuracy_score(y_train,y_train_predict)},precision_s
print("test")
print(f"accuracy_score={accuracy_score(y_test,y_test_predict)},precision_sco
print("")
```

```
C:\Users\Welcome\anaconda3\Lib\site-packages\xgboost\core.py:158: UserWarnin
g: [14:07:35] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autos
caling-group-i-0c55ff5f71b100e98-1\xgboost\xgboost-ci-windows\src\learner.c
c:740:
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
train
accuracy_score=0.9129513818842041,precision_score=0.9066008846546444,recall_
score=0.9443558390926812
test
accuracy_score=0.745865268253328,precision_score=0.7478202548625084,recall_s
core=0.8144631117604091
```

## Adaboost

In [129…

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Loop over different values of n_estimators
for i in range(10, 260, 10):
    ada_model = AdaBoostClassifier(n_estimators=i, random_state=42)
    ada_model.fit(x_train, y_train)

    # Predictions
    y_train_predict = ada_model.predict(x_train)
    y_test_predict = ada_model.predict(x_test)

    # Evaluation Metrics
    print(f"n_estimators = {i}")

    print("Train:")
    print(f"Accuracy = {accuracy_score(y_train, y_train_predict):.4f}, "
          f"Precision = {precision_score(y_train, y_train_predict):.4f}, "
          f"Recall = {recall_score(y_train, y_train_predict):.4f}")

    print("Test:")
    print(f"Accuracy = {accuracy_score(y_test, y_test_predict):.4f}, "
          f"Precision = {precision_score(y_test, y_test_predict):.4f}, "
          f"Recall = {recall_score(y_test, y_test_predict):.4f}")
```

```
    print("-" * 50)  # Separator for readability
```

```
n_estimators = 10
Train:
Accuracy = 0.6587, Precision = 0.6767, Recall = 0.7666
Test:
Accuracy = 0.6571, Precision = 0.6615, Recall = 0.7765
--------------------------------------------------
n_estimators = 20
Train:
Accuracy = 0.6751, Precision = 0.6927, Recall = 0.7716
Test:
Accuracy = 0.6789, Precision = 0.6810, Recall = 0.7874
--------------------------------------------------
n_estimators = 30
Train:
Accuracy = 0.6808, Precision = 0.7013, Recall = 0.7648
Test:
Accuracy = 0.6801, Precision = 0.6858, Recall = 0.7765
--------------------------------------------------
n_estimators = 40
Train:
Accuracy = 0.6838, Precision = 0.7025, Recall = 0.7709
Test:
Accuracy = 0.6757, Precision = 0.6817, Recall = 0.7743
--------------------------------------------------
n_estimators = 50
Train:
Accuracy = 0.6852, Precision = 0.7042, Recall = 0.7705
Test:
Accuracy = 0.6765, Precision = 0.6812, Recall = 0.7787
--------------------------------------------------
n_estimators = 60
Train:
Accuracy = 0.6867, Precision = 0.7064, Recall = 0.7693
Test:
Accuracy = 0.6785, Precision = 0.6838, Recall = 0.7772
--------------------------------------------------
n_estimators = 70
Train:
Accuracy = 0.6868, Precision = 0.7061, Recall = 0.7705
Test:
Accuracy = 0.6773, Precision = 0.6837, Recall = 0.7736
--------------------------------------------------
n_estimators = 80
Train:
Accuracy = 0.6871, Precision = 0.7062, Recall = 0.7710
Test:
Accuracy = 0.6789, Precision = 0.6842, Recall = 0.7772
--------------------------------------------------
n_estimators = 90
Train:
Accuracy = 0.6886, Precision = 0.7075, Recall = 0.7723
Test:
Accuracy = 0.6785, Precision = 0.6829, Recall = 0.7801
--------------------------------------------------
n_estimators = 100
Train:
```

```
Accuracy = 0.6891, Precision = 0.7077, Recall = 0.7732
Test:
Accuracy = 0.6801, Precision = 0.6839, Recall = 0.7823
--------------------------------------------------
n_estimators = 110
Train:
Accuracy = 0.6882, Precision = 0.7073, Recall = 0.7714
Test:
Accuracy = 0.6805, Precision = 0.6851, Recall = 0.7801
--------------------------------------------------
n_estimators = 120
Train:
Accuracy = 0.6906, Precision = 0.7100, Recall = 0.7718
Test:
Accuracy = 0.6813, Precision = 0.6864, Recall = 0.7787
--------------------------------------------------
n_estimators = 130
Train:
Accuracy = 0.6908, Precision = 0.7103, Recall = 0.7714
Test:
Accuracy = 0.6801, Precision = 0.6863, Recall = 0.7750
--------------------------------------------------
n_estimators = 140
Train:
Accuracy = 0.6904, Precision = 0.7101, Recall = 0.7709
Test:
Accuracy = 0.6801, Precision = 0.6863, Recall = 0.7750
--------------------------------------------------
n_estimators = 150
Train:
Accuracy = 0.6905, Precision = 0.7103, Recall = 0.7707
Test:
Accuracy = 0.6813, Precision = 0.6862, Recall = 0.7794
--------------------------------------------------
n_estimators = 160
Train:
Accuracy = 0.6907, Precision = 0.7105, Recall = 0.7707
Test:
Accuracy = 0.6850, Precision = 0.6885, Recall = 0.7845
--------------------------------------------------
n_estimators = 170
Train:
Accuracy = 0.6907, Precision = 0.7108, Recall = 0.7700
Test:
Accuracy = 0.6813, Precision = 0.6862, Recall = 0.7794
--------------------------------------------------
n_estimators = 180
Train:
Accuracy = 0.6899, Precision = 0.7101, Recall = 0.7694
Test:
Accuracy = 0.6825, Precision = 0.6880, Recall = 0.7779
--------------------------------------------------
n_estimators = 190
Train:
Accuracy = 0.6896, Precision = 0.7100, Recall = 0.7687
Test:
```

```
Accuracy = 0.6813, Precision = 0.6869, Recall = 0.7772
--------------------------------------------------
n_estimators = 200
Train:
Accuracy = 0.6907, Precision = 0.7103, Recall = 0.7712
Test:
Accuracy = 0.6837, Precision = 0.6881, Recall = 0.7816
--------------------------------------------------
n_estimators = 210
Train:
Accuracy = 0.6902, Precision = 0.7099, Recall = 0.7709
Test:
Accuracy = 0.6833, Precision = 0.6877, Recall = 0.7816
--------------------------------------------------
n_estimators = 220
Train:
Accuracy = 0.6899, Precision = 0.7095, Recall = 0.7709
Test:
Accuracy = 0.6833, Precision = 0.6881, Recall = 0.7801
--------------------------------------------------
n_estimators = 230
Train:
Accuracy = 0.6908, Precision = 0.7103, Recall = 0.7714
Test:
Accuracy = 0.6833, Precision = 0.6879, Recall = 0.7809
--------------------------------------------------
n_estimators = 240
Train:
Accuracy = 0.6898, Precision = 0.7094, Recall = 0.7709
Test:
Accuracy = 0.6846, Precision = 0.6887, Recall = 0.7823
--------------------------------------------------
n_estimators = 250
Train:
Accuracy = 0.6903, Precision = 0.7098, Recall = 0.7714
Test:
Accuracy = 0.6817, Precision = 0.6866, Recall = 0.7794
--------------------------------------------------
```

In [132…

```
#CATAGORICAL BOOSTING

!pip install catboost
install.packages("catboost")
from catboost import CatBoostClassifier
from sklearn.model_selection import GridSearchCV
```

```
Collecting catboost
  Downloading catboost-1.2.7-cp311-cp311-win_amd64.whl.metadata (1.2 kB)
Collecting graphviz (from catboost)
  Downloading graphviz-0.20.3-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: matplotlib in c:\users\welcome\anaconda3\lib
\site-packages (from catboost) (3.8.0)
Requirement already satisfied: numpy<2.0,>=1.16.0 in c:\users\welcome\anacon
da3\lib\site-packages (from catboost) (1.26.4)
Requirement already satisfied: pandas>=0.24 in c:\users\welcome\anaconda3\li
b\site-packages (from catboost) (2.1.4)
Requirement already satisfied: scipy in c:\users\welcome\anaconda3\lib\site-
packages (from catboost) (1.11.4)
Requirement already satisfied: plotly in c:\users\welcome\anaconda3\lib\site
-packages (from catboost) (5.9.0)
Requirement already satisfied: six in c:\users\welcome\anaconda3\lib\site-pa
ckages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\welcome\an
aconda3\lib\site-packages (from pandas>=0.24->catboost) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\welcome\anaconda3\li
b\site-packages (from pandas>=0.24->catboost) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\welcome\anaconda3
\lib\site-packages (from pandas>=0.24->catboost) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\welcome\anaconda
3\lib\site-packages (from matplotlib->catboost) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\welcome\anaconda3\li
b\site-packages (from matplotlib->catboost) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\welcome\anacond
a3\lib\site-packages (from matplotlib->catboost) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\welcome\anacond
a3\lib\site-packages (from matplotlib->catboost) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\welcome\anaconda3
\lib\site-packages (from matplotlib->catboost) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\welcome\anaconda3\l
ib\site-packages (from matplotlib->catboost) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\welcome\anaconda
3\lib\site-packages (from matplotlib->catboost) (3.0.9)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\welcome\anaconda3
\lib\site-packages (from plotly->catboost) (8.2.2)
Downloading catboost-1.2.7-cp311-cp311-win_amd64.whl (101.7 MB)
   ---------------------------------------- 0.0/101.7 MB ? eta -:--:--
   ---------------------------------------- 0.0/101.7 MB ? eta -:--:--
   ---------------------------------------- 0.0/101.7 MB 222.6 kB/s eta 0:0
7:37
   ---------------------------------------- 0.0/101.7 MB 330.3 kB/s eta 0:0
5:08
   ---------------------------------------- 0.1/101.7 MB 751.6 kB/s eta 0:0
2:16
   ---------------------------------------- 0.2/101.7 MB 1.1 MB/s eta 0:01:3
2
   ---------------------------------------- 0.2/101.7 MB 1.1 MB/s eta 0:01:3
6
   ---------------------------------------- 0.4/101.7 MB 1.7 MB/s eta 0:01:0
1
   ---------------------------------------- 0.5/101.7 MB 1.9 MB/s eta 0:00:5
5
   ---------------------------------------- 0.5/101.7 MB 1.9 MB/s eta 0:00:5
```

```
5
----------------------------------------  0.7/101.7 MB 1.9 MB/s eta 0:00:5
3
----------------------------------------  0.8/101.7 MB 2.0 MB/s eta 0:00:5
1
----------------------------------------  0.8/101.7 MB 1.9 MB/s eta 0:00:5
2
----------------------------------------  0.9/101.7 MB 2.0 MB/s eta 0:00:5
1
----------------------------------------  1.0/101.7 MB 1.9 MB/s eta 0:00:5
3
----------------------------------------  1.1/101.7 MB 2.0 MB/s eta 0:00:5
1
----------------------------------------  1.1/101.7 MB 2.0 MB/s eta 0:00:5
0
----------------------------------------  1.3/101.7 MB 2.1 MB/s eta 0:00:4
9
 ---------------------------------------  1.3/101.7 MB 2.1 MB/s eta 0:00:4
9
 ---------------------------------------  1.5/101.7 MB 2.1 MB/s eta 0:00:4
8
 ---------------------------------------  1.6/101.7 MB 2.2 MB/s eta 0:00:4
7
 ---------------------------------------  1.7/101.7 MB 2.3 MB/s eta 0:00:4
5
 ---------------------------------------  1.9/101.7 MB 2.3 MB/s eta 0:00:4
3
 ---------------------------------------  2.0/101.7 MB 2.3 MB/s eta 0:00:4
3
 ---------------------------------------  2.1/101.7 MB 2.4 MB/s eta 0:00:4
2
 ---------------------------------------  2.3/101.7 MB 2.5 MB/s eta 0:00:4
0
 ---------------------------------------  2.4/101.7 MB 2.5 MB/s eta 0:00:3
9
- --------------------------------------  2.6/101.7 MB 2.6 MB/s eta 0:00:3
8
- --------------------------------------  2.7/101.7 MB 2.6 MB/s eta 0:00:3
8
- --------------------------------------  2.7/101.7 MB 2.6 MB/s eta 0:00:3
9
- --------------------------------------  2.9/101.7 MB 2.7 MB/s eta 0:00:3
8
- --------------------------------------  3.1/101.7 MB 2.7 MB/s eta 0:00:3
7
- --------------------------------------  3.2/101.7 MB 2.7 MB/s eta 0:00:3
6
- --------------------------------------  3.4/101.7 MB 2.8 MB/s eta 0:00:3
6
- --------------------------------------  3.5/101.7 MB 2.8 MB/s eta 0:00:3
5
- --------------------------------------  3.6/101.7 MB 2.8 MB/s eta 0:00:3
5
- --------------------------------------  3.8/101.7 MB 2.9 MB/s eta 0:00:3
5
- --------------------------------------  3.9/101.7 MB 2.9 MB/s eta 0:00:3
```

```
4
   - -------------------------------------- 3.9/101.7 MB 2.9 MB/s eta 0:00:3
4
   - -------------------------------------- 4.3/101.7 MB 3.0 MB/s eta 0:00:3
3
   - -------------------------------------- 4.5/101.7 MB 3.0 MB/s eta 0:00:3
2
   - -------------------------------------- 4.7/101.7 MB 3.1 MB/s eta 0:00:3
2
   - -------------------------------------- 4.8/101.7 MB 3.1 MB/s eta 0:00:3
2
   - -------------------------------------- 5.0/101.7 MB 3.1 MB/s eta 0:00:3
1
   -- ------------------------------------- 5.2/101.7 MB 3.2 MB/s eta 0:00:3
1
   -- ------------------------------------- 5.4/101.7 MB 3.2 MB/s eta 0:00:3
1
   -- ------------------------------------- 5.6/101.7 MB 3.3 MB/s eta 0:00:3
0
   -- ------------------------------------- 5.7/101.7 MB 3.3 MB/s eta 0:00:3
0
   -- ------------------------------------- 5.9/101.7 MB 3.3 MB/s eta 0:00:2
9
   -- ------------------------------------- 6.2/101.7 MB 3.4 MB/s eta 0:00:2
9
   -- ------------------------------------- 6.4/101.7 MB 3.4 MB/s eta 0:00:2
8
   -- ------------------------------------- 6.6/101.7 MB 3.4 MB/s eta 0:00:2
8
   -- ------------------------------------- 6.8/101.7 MB 3.5 MB/s eta 0:00:2
8
   -- ------------------------------------- 7.0/101.7 MB 3.5 MB/s eta 0:00:2
7
   -- ------------------------------------- 7.2/101.7 MB 3.6 MB/s eta 0:00:2
7
   -- ------------------------------------- 7.4/101.7 MB 3.6 MB/s eta 0:00:2
7
   -- ------------------------------------- 7.6/101.7 MB 3.6 MB/s eta 0:00:2
7
   --- ------------------------------------ 7.8/101.7 MB 3.6 MB/s eta 0:00:2
6
   --- ------------------------------------ 8.0/101.7 MB 3.7 MB/s eta 0:00:2
6
   --- ------------------------------------ 8.2/101.7 MB 3.7 MB/s eta 0:00:2
6
   --- ------------------------------------ 8.4/101.7 MB 3.7 MB/s eta 0:00:2
6
   --- ------------------------------------ 8.6/101.7 MB 3.7 MB/s eta 0:00:2
5
   --- ------------------------------------ 8.9/101.7 MB 3.8 MB/s eta 0:00:2
5
   --- ------------------------------------ 9.2/101.7 MB 3.9 MB/s eta 0:00:2
4
   --- ------------------------------------ 9.5/101.7 MB 4.0 MB/s eta 0:00:2
4
   --- ------------------------------------ 9.9/101.7 MB 4.0 MB/s eta 0:00:2
```

```
3
   --- ------------------------------------ 10.1/101.7 MB 4.1 MB/s eta 0:00:
23
   ---- ----------------------------------- 10.3/101.7 MB 4.3 MB/s eta 0:00:
22
   ---- ----------------------------------- 10.4/101.7 MB 4.3 MB/s eta 0:00:
22
   ---- ----------------------------------- 10.7/101.7 MB 4.4 MB/s eta 0:00:
21
   ---- ----------------------------------- 11.0/101.7 MB 4.6 MB/s eta 0:00:
20
   ---- ----------------------------------- 11.3/101.7 MB 4.8 MB/s eta 0:00:
19
   ---- ----------------------------------- 11.6/101.7 MB 5.0 MB/s eta 0:00:
18
   ---- ----------------------------------- 11.9/101.7 MB 5.1 MB/s eta 0:00:
18
   ---- ----------------------------------- 12.2/101.7 MB 5.3 MB/s eta 0:00:
17
   ---- ----------------------------------- 12.5/101.7 MB 5.4 MB/s eta 0:00:
17
   ----- ---------------------------------- 12.8/101.7 MB 5.4 MB/s eta 0:00:
17
   ----- ---------------------------------- 13.0/101.7 MB 5.6 MB/s eta 0:00:
16
   ----- ---------------------------------- 13.4/101.7 MB 5.7 MB/s eta 0:00:
16
   ----- ---------------------------------- 13.7/101.7 MB 5.8 MB/s eta 0:00:
16
   ----- ---------------------------------- 14.1/101.7 MB 6.1 MB/s eta 0:00:
15
   ----- ---------------------------------- 14.3/101.7 MB 6.3 MB/s eta 0:00:
14
   ----- ---------------------------------- 14.6/101.7 MB 6.2 MB/s eta 0:00:
15
   ----- ---------------------------------- 14.9/101.7 MB 6.3 MB/s eta 0:00:
14
   ----- ---------------------------------- 15.2/101.7 MB 6.4 MB/s eta 0:00:
14
   ------ --------------------------------- 15.5/101.7 MB 6.5 MB/s eta 0:00:
14
   ------ --------------------------------- 15.7/101.7 MB 6.5 MB/s eta 0:00:
14
   ------ --------------------------------- 15.8/101.7 MB 6.4 MB/s eta 0:00:
14
   ------ --------------------------------- 16.0/101.7 MB 6.5 MB/s eta 0:00:
14
   ------ --------------------------------- 16.3/101.7 MB 6.5 MB/s eta 0:00:
14
   ------ --------------------------------- 16.5/101.7 MB 6.5 MB/s eta 0:00:
14
   ------ --------------------------------- 16.7/101.7 MB 6.5 MB/s eta 0:00:
13
   ------ --------------------------------- 16.9/101.7 MB 6.5 MB/s eta 0:00:
13
   ------ --------------------------------- 17.1/101.7 MB 6.5 MB/s eta 0:00:
```

```
13
   ------ --------------------------------- 17.5/101.7 MB 6.6 MB/s eta 0:00:
13
   ------ --------------------------------- 17.8/101.7 MB 6.7 MB/s eta 0:00:
13
   ------- -------------------------------- 18.1/101.7 MB 6.9 MB/s eta 0:00:
13
   ------- -------------------------------- 18.3/101.7 MB 6.8 MB/s eta 0:00:
13
   ------- -------------------------------- 18.5/101.7 MB 6.9 MB/s eta 0:00:
13
   ------- -------------------------------- 18.7/101.7 MB 6.8 MB/s eta 0:00:
13
   ------- -------------------------------- 18.9/101.7 MB 6.8 MB/s eta 0:00:
13
   ------- -------------------------------- 19.2/101.7 MB 6.8 MB/s eta 0:00:
13
   ------- -------------------------------- 19.5/101.7 MB 6.7 MB/s eta 0:00:
13
   ------- -------------------------------- 19.6/101.7 MB 6.7 MB/s eta 0:00:
13
   ------- -------------------------------- 19.9/101.7 MB 6.6 MB/s eta 0:00:
13
   ------- -------------------------------- 20.1/101.7 MB 6.5 MB/s eta 0:00:
13
   -------- ------------------------------- 20.4/101.7 MB 6.6 MB/s eta 0:00:
13
   -------- ------------------------------- 20.8/101.7 MB 6.8 MB/s eta 0:00:
12
   -------- ------------------------------- 21.3/101.7 MB 6.9 MB/s eta 0:00:
12
   -------- ------------------------------- 21.6/101.7 MB 6.9 MB/s eta 0:00:
12
   -------- ------------------------------- 22.2/101.7 MB 7.0 MB/s eta 0:00:
12
   -------- ------------------------------- 22.8/101.7 MB 7.3 MB/s eta 0:00:
11
   --------- ------------------------------ 23.3/101.7 MB 7.4 MB/s eta 0:00:
11
   --------- ------------------------------ 24.0/101.7 MB 7.8 MB/s eta 0:00:
10
   --------- ------------------------------ 24.7/101.7 MB 8.1 MB/s eta 0:00:
10
   --------- ------------------------------ 25.3/101.7 MB 8.2 MB/s eta 0:00:
10
   ---------- ----------------------------- 25.8/101.7 MB 8.5 MB/s eta 0:00:
09
   ---------- ----------------------------- 26.3/101.7 MB 9.0 MB/s eta 0:00:
09
   ---------- ----------------------------- 26.8/101.7 MB 9.4 MB/s eta 0:00:
09
   ---------- ----------------------------- 27.3/101.7 MB 9.8 MB/s eta 0:00:
08
   ---------- ----------------------------- 27.8/101.7 MB 10.2 MB/s eta 0:0
0:08
   ----------- ---------------------------- 28.2/101.7 MB 10.4 MB/s eta 0:0
```

```
0:08
   ----------- ---------------------------- 28.6/101.7 MB 10.6 MB/s eta 0:0
0:07
   ----------- ---------------------------- 29.1/101.7 MB 11.1 MB/s eta 0:0
0:07
   ----------- ---------------------------- 29.8/101.7 MB 12.1 MB/s eta 0:0
0:06
   ----------- ---------------------------- 30.2/101.7 MB 12.6 MB/s eta 0:0
0:06
   ------------ --------------------------- 30.7/101.7 MB 13.1 MB/s eta 0:0
0:06
   ------------ --------------------------- 31.5/101.7 MB 13.6 MB/s eta 0:0
0:06
   ------------ --------------------------- 32.0/101.7 MB 13.6 MB/s eta 0:0
0:06
   ------------ --------------------------- 32.6/101.7 MB 13.6 MB/s eta 0:0
0:06
   ------------- -------------------------- 33.1/101.7 MB 13.4 MB/s eta 0:0
0:06
   ------------- -------------------------- 33.8/101.7 MB 13.4 MB/s eta 0:0
0:06
   ------------- -------------------------- 34.3/101.7 MB 13.6 MB/s eta 0:0
0:05
   ------------- -------------------------- 34.8/101.7 MB 13.1 MB/s eta 0:0
0:06
   ------------- -------------------------- 35.4/101.7 MB 13.4 MB/s eta 0:0
0:05
   -------------- ------------------------- 35.9/101.7 MB 13.4 MB/s eta 0:0
0:05
   -------------- ------------------------- 36.4/101.7 MB 13.6 MB/s eta 0:0
0:05
   -------------- ------------------------- 37.1/101.7 MB 13.6 MB/s eta 0:0
0:05
   -------------- ------------------------- 37.8/101.7 MB 13.6 MB/s eta 0:0
0:05
   --------------- ------------------------ 38.6/101.7 MB 13.9 MB/s eta 0:0
0:05
   --------------- ------------------------ 39.2/101.7 MB 14.2 MB/s eta 0:0
0:05
   --------------- ------------------------ 39.9/101.7 MB 13.9 MB/s eta 0:0
0:05
   --------------- ------------------------ 40.4/101.7 MB 14.2 MB/s eta 0:0
0:05
   ---------------- ----------------------- 40.9/101.7 MB 14.2 MB/s eta 0:0
0:05
   ---------------- ----------------------- 41.5/101.7 MB 13.9 MB/s eta 0:0
0:05
   ---------------- ----------------------- 41.9/101.7 MB 13.6 MB/s eta 0:0
0:05
   ---------------- ----------------------- 42.6/101.7 MB 13.9 MB/s eta 0:0
0:05
   ---------------- ----------------------- 43.1/101.7 MB 13.9 MB/s eta 0:0
0:05
   ----------------- ---------------------- 43.8/101.7 MB 13.9 MB/s eta 0:0
0:05
   ----------------- ---------------------- 44.3/101.7 MB 13.9 MB/s eta 0:0
```

```
0:05
   ----------------- ---------------------- 45.1/101.7 MB 14.2 MB/s eta 0:0
0:04
   ----------------- ---------------------- 45.6/101.7 MB 13.9 MB/s eta 0:0
0:05
   ------------------ --------------------- 46.1/101.7 MB 14.2 MB/s eta 0:0
0:04
   ------------------ --------------------- 46.6/101.7 MB 14.2 MB/s eta 0:0
0:04
   ------------------ --------------------- 47.0/101.7 MB 13.9 MB/s eta 0:0
0:04
   ------------------ --------------------- 47.6/101.7 MB 13.9 MB/s eta 0:0
0:04
   ------------------ --------------------- 48.3/101.7 MB 13.9 MB/s eta 0:0
0:04
   ------------------- -------------------- 49.0/101.7 MB 13.9 MB/s eta 0:0
0:04
   ------------------- -------------------- 49.7/101.7 MB 13.9 MB/s eta 0:0
0:04
   ------------------- -------------------- 50.2/101.7 MB 14.2 MB/s eta 0:0
0:04
   ------------------- -------------------- 50.6/101.7 MB 13.9 MB/s eta 0:0
0:04
   -------------------- ------------------- 51.2/101.7 MB 14.2 MB/s eta 0:0
0:04
   -------------------- ------------------- 52.0/101.7 MB 14.6 MB/s eta 0:0
0:04
   -------------------- ------------------- 52.5/101.7 MB 14.6 MB/s eta 0:0
0:04
   -------------------- ------------------- 53.2/101.7 MB 14.9 MB/s eta 0:0
0:04
   --------------------- ------------------ 53.7/101.7 MB 14.9 MB/s eta 0:0
0:04
   --------------------- ------------------ 54.3/101.7 MB 14.9 MB/s eta 0:0
0:04
   --------------------- ------------------ 54.7/101.7 MB 14.6 MB/s eta 0:0
0:04
   --------------------- ------------------ 55.3/101.7 MB 14.2 MB/s eta 0:0
0:04
   ---------------------- ----------------- 56.0/101.7 MB 14.9 MB/s eta 0:0
0:04
   ---------------------- ----------------- 56.5/101.7 MB 14.9 MB/s eta 0:0
0:04
   ---------------------- ----------------- 57.1/101.7 MB 15.2 MB/s eta 0:0
0:03
   ---------------------- ----------------- 57.7/101.7 MB 14.9 MB/s eta 0:0
0:03
   ---------------------- ----------------- 58.3/101.7 MB 15.2 MB/s eta 0:0
0:03
   ----------------------- ---------------- 59.0/101.7 MB 15.2 MB/s eta 0:0
0:03
   ----------------------- ---------------- 59.6/101.7 MB 14.9 MB/s eta 0:0
0:03
   ----------------------- ---------------- 60.1/101.7 MB 14.9 MB/s eta 0:0
0:03
   ----------------------- ---------------- 60.7/101.7 MB 14.9 MB/s eta 0:0
```

```
0:03
   ------------------------ --------------- 61.3/101.7 MB 15.6 MB/s eta 0:0
0:03
   ------------------------ --------------- 61.8/101.7 MB 14.9 MB/s eta 0:0
0:03
   ------------------------ --------------- 62.2/101.7 MB 14.6 MB/s eta 0:0
0:03
   ------------------------ --------------- 62.7/101.7 MB 14.6 MB/s eta 0:0
0:03
   ------------------------ --------------- 63.3/101.7 MB 14.2 MB/s eta 0:0
0:03
   ------------------------- -------------- 63.7/101.7 MB 14.2 MB/s eta 0:0
0:03
   ------------------------- -------------- 64.2/101.7 MB 13.9 MB/s eta 0:0
0:03
   ------------------------- -------------- 64.7/101.7 MB 14.2 MB/s eta 0:0
0:03
   ------------------------- -------------- 65.0/101.7 MB 13.9 MB/s eta 0:0
0:03
   ------------------------- -------------- 65.5/101.7 MB 13.9 MB/s eta 0:0
0:03
   ------------------------- -------------- 65.8/101.7 MB 13.9 MB/s eta 0:0
0:03
   -------------------------- ------------- 66.3/101.7 MB 13.4 MB/s eta 0:0
0:03
   -------------------------- ------------- 66.8/101.7 MB 12.9 MB/s eta 0:0
0:03
   -------------------------- ------------- 67.5/101.7 MB 13.1 MB/s eta 0:0
0:03
   -------------------------- ------------- 68.0/101.7 MB 13.1 MB/s eta 0:0
0:03
   -------------------------- ------------- 68.4/101.7 MB 13.1 MB/s eta 0:0
0:03
   --------------------------- ------------ 68.9/101.7 MB 12.8 MB/s eta 0:0
0:03
   --------------------------- ------------ 69.3/101.7 MB 12.6 MB/s eta 0:0
0:03
   --------------------------- ------------ 69.9/101.7 MB 12.6 MB/s eta 0:0
0:03
   --------------------------- ------------ 70.5/101.7 MB 12.6 MB/s eta 0:0
0:03
   --------------------------- ------------ 71.1/101.7 MB 12.6 MB/s eta 0:0
0:03
   ---------------------------- ----------- 71.4/101.7 MB 12.4 MB/s eta 0:0
0:03
   ---------------------------- ----------- 72.2/101.7 MB 12.6 MB/s eta 0:0
0:03
   ---------------------------- ----------- 72.6/101.7 MB 12.8 MB/s eta 0:0
0:03
   ---------------------------- ----------- 73.1/101.7 MB 12.6 MB/s eta 0:0
0:03
   ---------------------------- ----------- 73.7/101.7 MB 12.9 MB/s eta 0:0
0:03
   ----------------------------- ---------- 74.2/101.7 MB 12.8 MB/s eta 0:0
0:03
   ----------------------------- ---------- 74.8/101.7 MB 13.1 MB/s eta 0:0
```

```
0:03
   ----------------------------- ---------- 75.4/101.7 MB 13.1 MB/s eta 0:0
0:03
   ----------------------------- ---------- 75.9/101.7 MB 13.4 MB/s eta 0:0
0:02
   ------------------------------ --------- 76.4/101.7 MB 13.6 MB/s eta 0:0
0:02
   ------------------------------ --------- 76.9/101.7 MB 13.9 MB/s eta 0:0
0:02
   ------------------------------ --------- 77.5/101.7 MB 13.6 MB/s eta 0:0
0:02
   ------------------------------ --------- 77.9/101.7 MB 13.6 MB/s eta 0:0
0:02
   ------------------------------ --------- 78.3/101.7 MB 13.4 MB/s eta 0:0
0:02
   ------------------------------ --------- 78.8/101.7 MB 13.4 MB/s eta 0:0
0:02
   ------------------------------- -------- 79.3/101.7 MB 13.6 MB/s eta 0:0
0:02
   ------------------------------- -------- 80.0/101.7 MB 13.6 MB/s eta 0:0
0:02
   ------------------------------- -------- 80.6/101.7 MB 13.6 MB/s eta 0:0
0:02
   ------------------------------- -------- 81.1/101.7 MB 13.4 MB/s eta 0:0
0:02
   -------------------------------- ------- 81.6/101.7 MB 13.6 MB/s eta 0:0
0:02
   -------------------------------- ------- 82.1/101.7 MB 13.4 MB/s eta 0:0
0:02
   -------------------------------- ------- 82.6/101.7 MB 13.1 MB/s eta 0:0
0:02
   -------------------------------- ------- 82.9/101.7 MB 13.1 MB/s eta 0:0
0:02
   -------------------------------- ------- 83.4/101.7 MB 12.8 MB/s eta 0:0
0:02
   -------------------------------- ------- 83.8/101.7 MB 12.6 MB/s eta 0:0
0:02
   --------------------------------- ------ 84.3/101.7 MB 12.4 MB/s eta 0:0
0:02
   --------------------------------- ------ 84.7/101.7 MB 12.1 MB/s eta 0:0
0:02
   --------------------------------- ------ 85.3/101.7 MB 12.1 MB/s eta 0:0
0:02
   --------------------------------- ------ 85.7/101.7 MB 11.9 MB/s eta 0:0
0:02
   --------------------------------- ------ 85.9/101.7 MB 11.5 MB/s eta 0:0
0:02
   --------------------------------- ------ 86.2/101.7 MB 11.3 MB/s eta 0:0
0:02
   ---------------------------------- ----- 86.5/101.7 MB 11.1 MB/s eta 0:0
0:02
   ---------------------------------- ----- 86.8/101.7 MB 11.1 MB/s eta 0:0
0:02
   ---------------------------------- ----- 87.0/101.7 MB 10.7 MB/s eta 0:0
0:02
   ---------------------------------- ----- 87.4/101.7 MB 10.6 MB/s eta 0:0
```

```
0:02
   ---------------------------------- ----- 87.4/101.7 MB 10.1 MB/s eta 0:0
0:02
   ---------------------------------- ----- 87.7/101.7 MB 10.1 MB/s eta 0:0
0:02
   ---------------------------------- ----- 88.0/101.7 MB 9.9 MB/s eta 0:00:
02
   ---------------------------------- ----- 88.2/101.7 MB 9.8 MB/s eta 0:00:
02
   ---------------------------------- ----- 88.4/101.7 MB 9.5 MB/s eta 0:00:
02
   ---------------------------------- ----- 88.7/101.7 MB 9.4 MB/s eta 0:00:
02
   ---------------------------------- ----- 88.9/101.7 MB 9.2 MB/s eta 0:00:
02
   ----------------------------------- ---- 89.0/101.7 MB 9.1 MB/s eta 0:00:
02
   ----------------------------------- ---- 89.3/101.7 MB 8.8 MB/s eta 0:00:
02
   ----------------------------------- ---- 89.4/101.7 MB 8.7 MB/s eta 0:00:
02
   ----------------------------------- ---- 89.6/101.7 MB 8.6 MB/s eta 0:00:
02
   ----------------------------------- ---- 89.8/101.7 MB 8.3 MB/s eta 0:00:
02
   ----------------------------------- ---- 89.9/101.7 MB 8.2 MB/s eta 0:00:
02
   ----------------------------------- ---- 90.1/101.7 MB 8.0 MB/s eta 0:00:
02
   ----------------------------------- ---- 90.1/101.7 MB 7.8 MB/s eta 0:00:
02
   ----------------------------------- ---- 90.2/101.7 MB 7.6 MB/s eta 0:00:
02
   ----------------------------------- ---- 90.5/101.7 MB 7.4 MB/s eta 0:00:
02
   ----------------------------------- ---- 90.6/101.7 MB 7.4 MB/s eta 0:00:
02
   ----------------------------------- ---- 90.8/101.7 MB 7.2 MB/s eta 0:00:
02
   ----------------------------------- ---- 91.1/101.7 MB 7.1 MB/s eta 0:00:
02
   ----------------------------------- ---- 91.4/101.7 MB 7.0 MB/s eta 0:00:
02
   ------------------------------------ --- 91.6/101.7 MB 7.0 MB/s eta 0:00:
02
   ------------------------------------ --- 91.9/101.7 MB 6.8 MB/s eta 0:00:
02
   ------------------------------------ --- 92.1/101.7 MB 6.7 MB/s eta 0:00:
02
   ------------------------------------ --- 92.4/101.7 MB 6.7 MB/s eta 0:00:
02
   ------------------------------------ --- 92.7/101.7 MB 6.6 MB/s eta 0:00:
02
   ------------------------------------ --- 93.0/101.7 MB 6.7 MB/s eta 0:00:
02
   ------------------------------------ --- 93.3/101.7 MB 6.6 MB/s eta 0:00:
```

```
02
   ------------------------------------ --- 93.6/101.7 MB 6.6 MB/s eta 0:00:
02
   ------------------------------------ --- 94.0/101.7 MB 6.5 MB/s eta 0:00:
02
   ------------------------------------- -- 94.3/101.7 MB 6.5 MB/s eta 0:00:
02
   ------------------------------------- -- 94.7/101.7 MB 6.5 MB/s eta 0:00:
02
   ------------------------------------- -- 95.0/101.7 MB 6.5 MB/s eta 0:00:
02
   ------------------------------------- -- 95.3/101.7 MB 6.5 MB/s eta 0:00:
01
   ------------------------------------- -- 95.7/101.7 MB 6.5 MB/s eta 0:00:
01
   ------------------------------------- -- 96.2/101.7 MB 6.5 MB/s eta 0:00:
01
   ------------------------------------- -- 96.5/101.7 MB 6.5 MB/s eta 0:00:
01
   -------------------------------------- - 96.9/101.7 MB 6.7 MB/s eta 0:00:
01
   -------------------------------------- - 97.3/101.7 MB 6.7 MB/s eta 0:00:
01
   -------------------------------------- - 97.6/101.7 MB 6.9 MB/s eta 0:00:
01
   -------------------------------------- - 98.2/101.7 MB 7.0 MB/s eta 0:00:
01
   -------------------------------------- - 98.5/101.7 MB 7.1 MB/s eta 0:00:
01
   -------------------------------------- - 98.9/101.7 MB 7.3 MB/s eta 0:00:
01
   ---------------------------------------  99.3/101.7 MB 7.4 MB/s eta 0:00:
01
   ---------------------------------------  99.6/101.7 MB 7.5 MB/s eta 0:00:
01
   ---------------------------------------  100.0/101.7 MB 7.7 MB/s eta 0:0
0:01
   ---------------------------------------  100.4/101.7 MB 8.3 MB/s eta 0:0
0:01
   ---------------------------------------  100.7/101.7 MB 8.5 MB/s eta 0:0
0:01
   ---------------------------------------  101.0/101.7 MB 8.7 MB/s eta 0:0
0:01
   ---------------------------------------  101.2/101.7 MB 8.7 MB/s eta 0:0
0:01
   ---------------------------------------  101.4/101.7 MB 8.7 MB/s eta 0:0
0:01
   ---------------------------------------  101.7/101.7 MB 8.6 MB/s eta 0:0
0:01
   ---------------------------------------  101.7/101.7 MB 8.6 MB/s eta 0:0
0:01
   ---------------------------------------  101.7/101.7 MB 8.6 MB/s eta 0:0
0:01
   ---------------------------------------  101.7/101.7 MB 8.6 MB/s eta 0:0
0:01
   ---------------------------------------  101.7/101.7 MB 8.6 MB/s eta 0:0
```

```
0:01
   ---------------------------------------- 101.7/101.7 MB 7.1 MB/s eta 0:0
0:00
Downloading graphviz-0.20.3-py3-none-any.whl (47 kB)
   ---------------------------------------- 0.0/47.1 kB ? eta -:--:--
   ---------------------------------------- 47.1/47.1 kB 1.2 MB/s eta 0:00:0
0
Installing collected packages: graphviz, catboost
Successfully installed catboost-1.2.7 graphviz-0.20.3
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[132], line 4
      1 #CATAGORICAL BOOSTING
      3 get_ipython().system('pip install catboost')
----> 4 install.packages("catboost")
      5 from catboost import CatBoostClassifier
      6 from sklearn.model_selection import GridSearchCV

NameError: name 'install' is not defined
```

## Create a CatBoostClassifier

model = CatBoostClassifier(iterations=500, # Number of boosting iterations

depth=6, # Depth of the tree

learning_rate=0.1,

loss_function='MultiClass', # For multiclass classification

verbose=200) # Print progress every 200 iterations

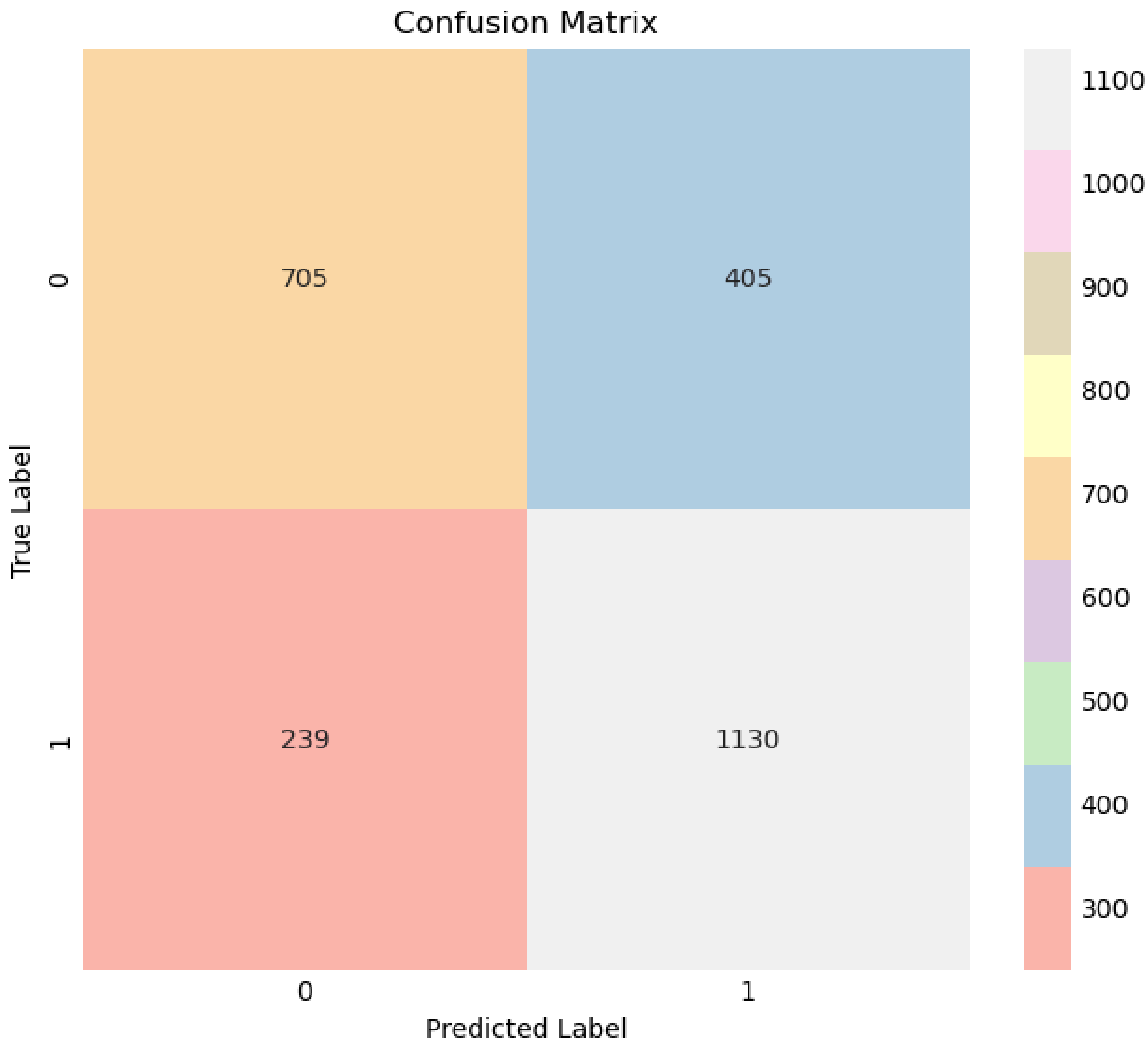
## CATAGORICAL BOOSTING

In [135…

```
# Import Libraries
from catboost import CatBoostClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score, c
import seaborn as sns
import matplotlib.pyplot as plt

# Define the base CatBoost model
cat_model = CatBoostClassifier(
    iterations=50,  # Number of boosting iterations
    depth=6,         # Depth of the tree
    learning_rate=0.1,
    loss_function='MultiClass',  # For multiclass classification
    verbose=200  # Print progress every 200 iterations
)

# Define hyperparameter grid for GridSearchCV
param_grid = {
    'iterations': [50, 100],  # We are taking here the iterations is (50,100
    'depth': [4, 6, 8],
    'learning_rate': [0.01, 0.1, 0.2],
```

```
    'l2_leaf_reg': [1, 3, 5],
}

# Perform Grid Search with Cross Validation
grid_search = GridSearchCV(estimator=cat_model, param_grid=param_grid, cv=5,
grid_search.fit(x_train, y_train)  # Fit the model

# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Train the best model
best_model = grid_search.best_estimator_

# Model Evaluation
train_pred = best_model.predict(x_train)
test_pred = best_model.predict(x_test)

print("\n--- Train Performance ---")
print(f"Accuracy: {accuracy_score(y_train, train_pred):.4f}")
print(f"Precision: {precision_score(y_train, train_pred, average='weighted')
print(f"Recall: {recall_score(y_train, train_pred, average='weighted'):.4f}"

print("\n--- Test Performance ---")
print(f"Accuracy: {accuracy_score(y_test, test_pred):.4f}")
print(f"Precision: {precision_score(y_test, test_pred, average='weighted'):.
print(f"Recall: {recall_score(y_test, test_pred, average='weighted'):.4f}")

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, test_pred)

# Plot Confusion Matrix
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Pastel1',
            xticklabels=best_model.classes_, yticklabels=best_model.classes_
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

# Classification Report
print("\nClassification Report:\n", classification_report(y_test, test_pred)
```

```
0:      learn: 0.6589250        total: 77.8ms   remaining: 7.71s
99:     learn: 0.3665393        total: 1.52s    remaining: 0us
Best Parameters: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 3, 'learning
_rate': 0.2}

--- Train Performance ---
Accuracy: 0.8652
Precision: 0.8667
Recall: 0.8652

--- Test Performance ---
Accuracy: 0.7402
Precision: 0.7409
Recall: 0.7402
```



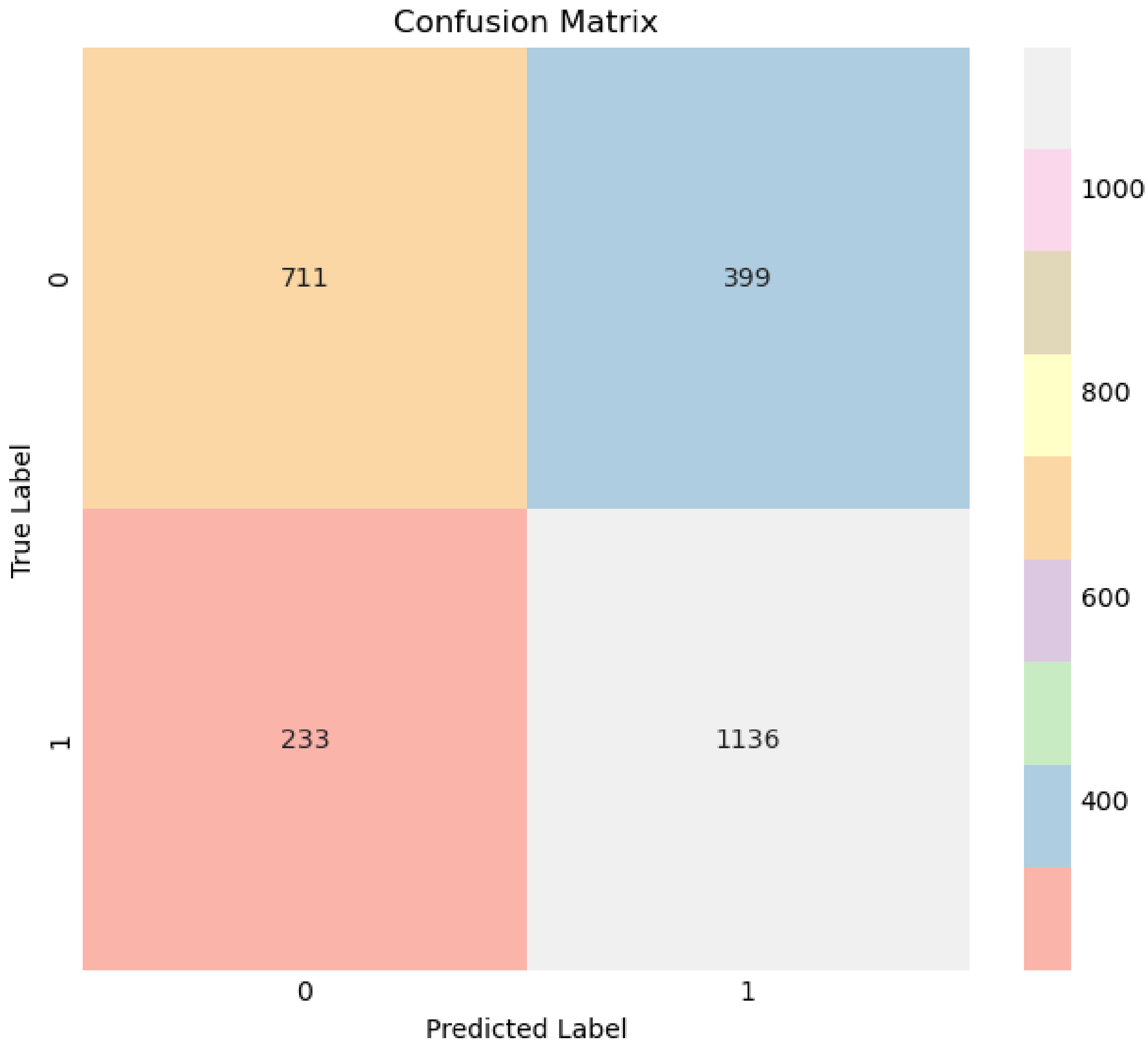
```
Classification Report:
               precision    recall  f1-score   support

           0       0.75      0.64      0.69      1110
           1       0.74      0.83      0.78      1369

    accuracy                           0.74      2479
   macro avg       0.74      0.73      0.73      2479
weighted avg       0.74      0.74      0.74      2479
```

In [136…

```python
# Import Libraries
from catboost import CatBoostClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score, c
import seaborn as sns
import matplotlib.pyplot as plt

# Define the base CatBoost model
cat_model = CatBoostClassifier(
    iterations=500,  # Number of boosting iterations
    depth=6,         # Depth of the tree
    learning_rate=0.1,
    loss_function='MultiClass',  # For multiclass classification
    verbose=200  # Print progress every 200 iterations
)

# Define hyperparameter grid for GridSearchCV
param_grid = {
    'iterations': [500, 1000],  #Now we are taking iterations (500,1000) and
    'depth': [4, 6, 8],
    'learning_rate': [0.01, 0.1, 0.2],
    'l2_leaf_reg': [1, 3, 5],
}

# Perform Grid Search with Cross Validation
grid_search = GridSearchCV(estimator=cat_model, param_grid=param_grid, cv=5,
grid_search.fit(x_train, y_train)  # Fit the model

# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Train the best model
best_model = grid_search.best_estimator_

# Model Evaluation
train_pred = best_model.predict(x_train)
test_pred = best_model.predict(x_test)

print("\n--- Train Performance ---")
print(f"Accuracy: {accuracy_score(y_train, train_pred):.4f}")
print(f"Precision: {precision_score(y_train, train_pred, average='weighted')
print(f"Recall: {recall_score(y_train, train_pred, average='weighted'):.4f}"

print("\n--- Test Performance ---")
```

```python
print(f"Accuracy: {accuracy_score(y_test, test_pred):.4f}")
print(f"Precision: {precision_score(y_test, test_pred, average='weighted'):.
print(f"Recall: {recall_score(y_test, test_pred, average='weighted'):.4f}")

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, test_pred)

# Plot Confusion Matrix
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Pastel1',
            xticklabels=best_model.classes_, yticklabels=best_model.classes_
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

# Classification Report
print("\nClassification Report:\n", classification_report(y_test, test_pred)
```

```
0:      learn: 0.6781240        total: 16.9ms   remaining: 8.43s
200:    learn: 0.4670769        total: 1.3s     remaining: 1.94s
400:    learn: 0.4041547        total: 2.5s     remaining: 617ms
499:    learn: 0.3827739        total: 3.1s     remaining: 0us
Best Parameters: {'depth': 6, 'iterations': 500, 'l2_leaf_reg': 5, 'learning
_rate': 0.1}

--- Train Performance ---
Accuracy: 0.8466
Precision: 0.8477
Recall: 0.8466

--- Test Performance ---
Accuracy: 0.7451
Precision: 0.7459
Recall: 0.7451
```




```
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.64      0.69      1110
           1       0.74      0.83      0.78      1369

    accuracy                           0.75      2479
   macro avg       0.75      0.74      0.74      2479
weighted avg       0.75      0.75      0.74      2479
```

Catboost for iterations 500,100 for checking the accuracy

Note: As comapare to iterations[50, 100] & iterations[500, 1000], iterations[50, 100] Gives better accuracy

Note: CATBOOST gives Better accuracy as compared to other models

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: