st125066_KaungNyoLwin_A3_Assignment

October 7, 2024

1 Car Price Prediction

The goal is to predict car price

The followings describe the features in the dataset

- **name** : Brand and series of the car
- **Year** : The year when the car is produced
- **selling_price** : Selling price of the car
- **km_driven** : The amount of kilometer that the car was driven before selling the car
- **fuel** : The type of fuel used in the car
- **seller_type** : The channel through which the deal of car is organized
- **transmission** : The gear transmission type that is used in the car
- **owner** : The number of times that the car was traded before
- **mileage** : The kilometers that can be traveled by using the fuel of 1 liter or 1 kg
- **engine** : The size of the engine
- **max_power** : The maximum power of engine measured in bhp. (BHP - the brake horsepower is the horse power after taking account of losses due to friction)
- **torque** : Torque
- **seats** : The number of seats in the car

1.1 Importing libraries

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import os
import mlflow

warnings.filterwarnings('ignore')
```

[2]:
```
import matplotlib
np.__version__, pd.__version__, sns.__version__, matplotlib.__version__
```

[2]:
```
('1.26.4', '2.2.2', '0.13.2', '3.9.2')
```

1.2 1. Load data

[3]:
```
# Load the data
df = pd.read_csv(os.path.join(os.getcwd(),"data/Cars.csv"))
```

[4]:
```
# print the first rows of data
df.head()
```

[4]:
```
                          name  year  selling_price  km_driven    fuel  \
0       Maruti Swift Dzire VDI  2014         450000     145500  Diesel
1  Skoda Rapid 1.5 TDI Ambition  2014         370000     120000  Diesel
2     Honda City 2017-2020 EXi  2006         158000     140000  Petrol
3    Hyundai i20 Sportz Diesel  2010         225000     127000  Diesel
4       Maruti Swift VXI BSIII  2007         130000     120000  Petrol

  seller_type transmission         owner     mileage   engine   max_power  \
0  Individual       Manual   First Owner   23.4 kmpl  1248 CC      74 bhp
1  Individual       Manual  Second Owner  21.14 kmpl  1498 CC  103.52 bhp
2  Individual       Manual   Third Owner   17.7 kmpl  1497 CC      78 bhp
3  Individual       Manual   First Owner   23.0 kmpl  1396 CC      90 bhp
4  Individual       Manual   First Owner   16.1 kmpl  1298 CC    88.2 bhp

                    torque  seats
0           190Nm@ 2000rpm    5.0
1      250Nm@ 1500-2500rpm    5.0
2    12.7@ 2,700(kgm@ rpm)    5.0
3  22.4 kgm at 1750-2750rpm    5.0
4    11.5@ 4,500(kgm@ rpm)    5.0
```

[5]:
```
# Check the shape of your data
df.shape
```

[5]: (8128, 13)

[6]:
```
# Statistical info
df.describe()
```

[6]:
```
              year  selling_price     km_driven        seats
count  8128.000000   8.128000e+03  8.128000e+03  7907.000000
mean   2013.804011   6.382718e+05  6.981951e+04     5.416719
std       4.044249   8.062534e+05  5.655055e+04     0.959588
min    1983.000000   2.999900e+04  1.000000e+00     2.000000
25%    2011.000000   2.549990e+05  3.500000e+04     5.000000
50%    2015.000000   4.500000e+05  6.000000e+04     5.000000
75%    2017.000000   6.750000e+05  9.800000e+04     5.000000
max    2020.000000   1.000000e+07  2.360457e+06    14.000000
```

[7]:
```python
# Check Dtypes of data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           8128 non-null   object
 5   seller_type    8128 non-null   object
 6   transmission   8128 non-null   object
 7   owner          8128 non-null   object
 8   mileage        7907 non-null   object
 9   engine         7907 non-null   object
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
```

[8]:
```python
# Check the column names
df.columns
```

[8]:
```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner', 'mileage', 'engine', 'max_power', 'torque',
       'seats'],
      dtype='object')
```

1.3 2. Exploratory Data Analysis

EDA is an essential step to inspect the data, so to better understand nature of the given data.

1.3.1 2.1 Data Cleansing

We need to perform data cleansing as the data is not tidy yet.

[9]:
```python
df.columns
```

[9]:
```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner', 'mileage', 'engine', 'max_power', 'torque',
       'seats'],
      dtype='object')
```

[10]:
```python
df.head(10)
```

[10]:

```
                             name  year  selling_price  km_driven  \
0          Maruti Swift Dzire VDI  2014         450000     145500
1        Skoda Rapid 1.5 TDI Ambition  2014         370000     120000
2        Honda City 2017-2020 EXi  2006         158000     140000
3       Hyundai i20 Sportz Diesel  2010         225000     127000
4          Maruti Swift VXI BSIII  2007         130000     120000
5      Hyundai Xcent 1.2 VTVT E Plus  2017         440000      45000
6     Maruti Wagon R LXI DUO BSIII  2007          96000     175000
7              Maruti 800 DX BSII  2001          45000       5000
8                Toyota Etios VXD  2011         350000      90000
9  Ford Figo Diesel Celebration Edition  2013         200000     169000

     fuel seller_type transmission         owner     mileage   engine  \
0  Diesel  Individual       Manual   First Owner   23.4 kmpl  1248 CC
1  Diesel  Individual       Manual  Second Owner  21.14 kmpl  1498 CC
2  Petrol  Individual       Manual   Third Owner   17.7 kmpl  1497 CC
3  Diesel  Individual       Manual   First Owner   23.0 kmpl  1396 CC
4  Petrol  Individual       Manual   First Owner   16.1 kmpl  1298 CC
5  Petrol  Individual       Manual   First Owner  20.14 kmpl  1197 CC
6     LPG  Individual       Manual   First Owner  17.3 km/kg  1061 CC
7  Petrol  Individual       Manual  Second Owner   16.1 kmpl   796 CC
8  Diesel  Individual       Manual   First Owner  23.59 kmpl  1364 CC
9  Diesel  Individual       Manual   First Owner   20.0 kmpl  1399 CC

    max_power                    torque  seats
0      74 bhp            190Nm@ 2000rpm    5.0
1  103.52 bhp       250Nm@ 1500-2500rpm    5.0
2      78 bhp      12.7@ 2,700(kgm@ rpm)    5.0
3      90 bhp  22.4 kgm at 1750-2750rpm    5.0
4    88.2 bhp      11.5@ 4,500(kgm@ rpm)    5.0
5   81.86 bhp          113.75nm@ 4000rpm    5.0
6    57.5 bhp       7.8@ 4,500(kgm@ rpm)    5.0
7      37 bhp             59Nm@ 2500rpm    4.0
8    67.1 bhp       170Nm@ 1800-2400rpm    5.0
9    68.1 bhp            160Nm@ 2000rpm    5.0
```

2.1.1 - We will remove torque feature as we don't understand the feature

[11]:

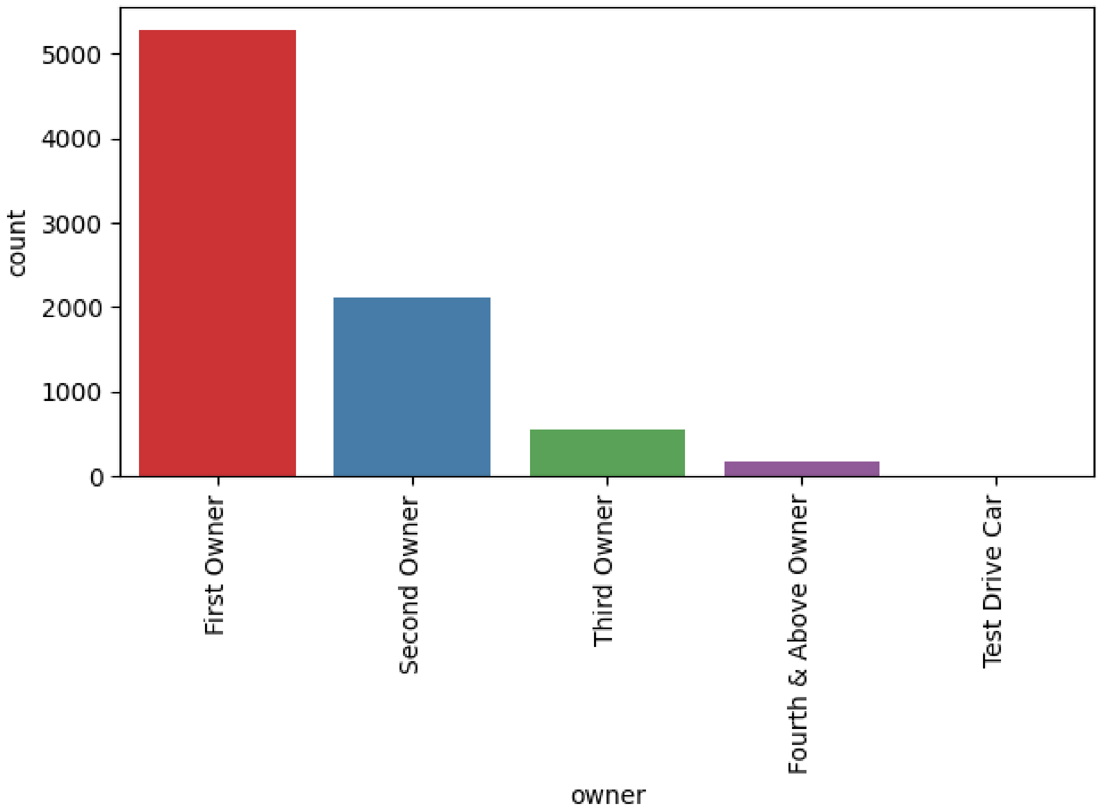
```python
# Remove torque feature
df.drop('torque',axis=1,inplace=True)
```

2.1.2 - We will remove the rows with owner "Test Drive Car" as the price is too high which makes them outliers and this only has 5 rows

[12]:

```python
# Checking counts of owner types
ax = sns.countplot(data = df, x = 'owner',palette= 'Set1')
ax.set_xticklabels(df['owner'].unique(),rotation=90)
plt.tight_layout()
```

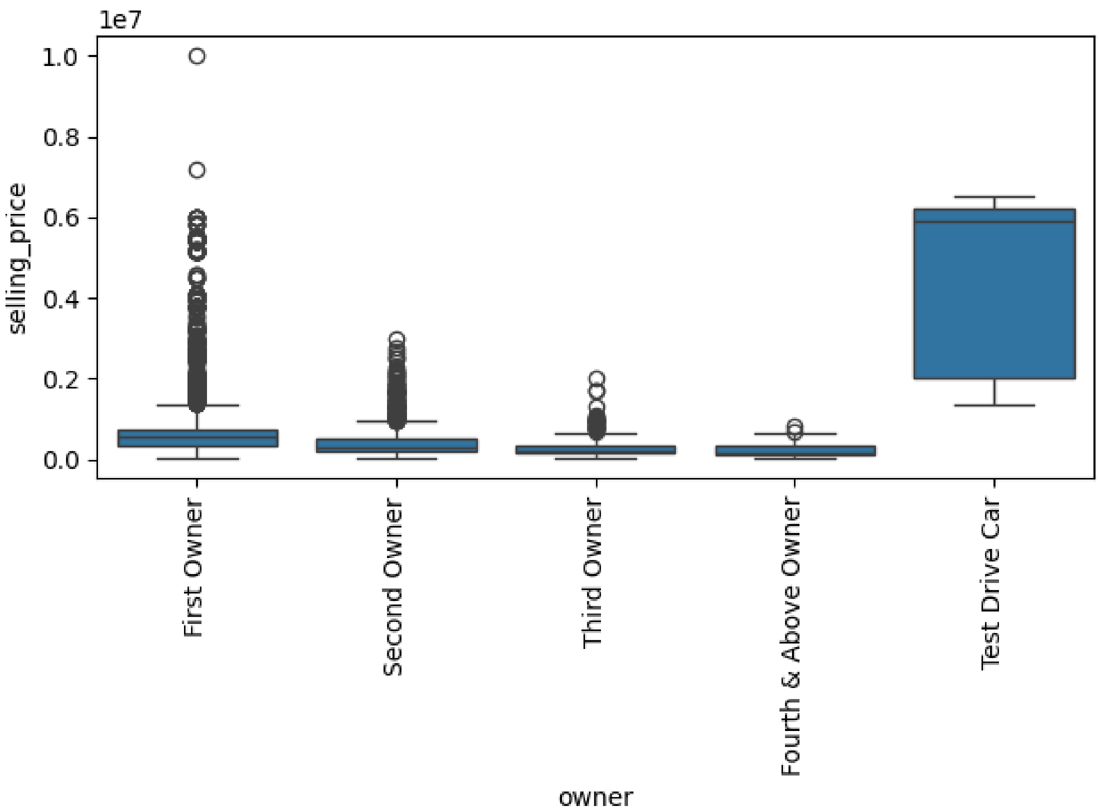
```
plt.show()
```



[13]:
```
# checking the count of test drive car
df.loc[df['owner'] == 'Test Drive Car','selling_price'].count()
```

[13]: 5

[14]:
```
# checking the distribution of owner types
ax = sns.boxplot(x = df["owner"], y = df["selling_price"])
ax.set_xticklabels(df['owner'].unique(),rotation=90)
plt.tight_layout()
plt.show()
```



[15]:
```
# Remove rows with "Test Drive Car"
df = df[df['owner'] != 'Test Drive Car']
```

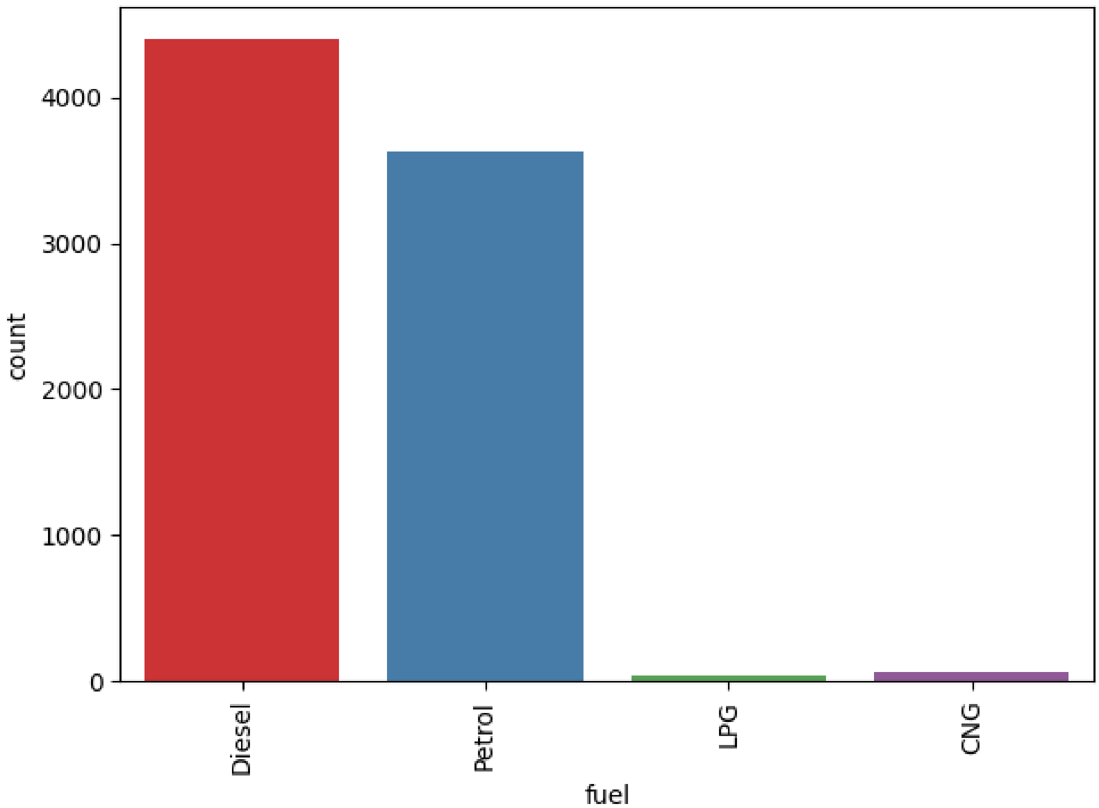
2.1.3 - We will map the owner types to 1 to 4

[16]:
```
owner = {'First Owner' : 1,
         'Second Owner' : 2,
         'Third Owner' : 3,
         'Fourth & Above Owner' : 4
         }

for k,v in owner.items():
    df.replace(k,v,inplace=True)
```

2.1.4 - We will remove the rows with fuel types "LPG" and "CNG" as they use different mileage units and it has only 38 rows

[17]:
```
# Checking counts of fuel types
ax = sns.countplot(data = df, x = 'fuel',palette= 'Set1')
ax.set_xticklabels(df['fuel'].unique(),rotation=90)
plt.tight_layout()
```

```
plt.show()
```




```
[18]: # Checking fuel milage units
df[df['fuel'].isin(['LPG','CND'])]['mileage'].head()
```

```
[18]: 6       17.3 km/kg
90      26.2 km/kg
870     26.2 km/kg
1511    26.2 km/kg
1658    17.3 km/kg
Name: mileage, dtype: object
```

```
[19]: # Checking the number of fuel types "LPG,CNd"
df[df['fuel'].isin(['LPG','CND'])].shape
```

```
[19]: (38, 12)
```

```
[20]: # Remove the rows with fuel types "LPG,CNd"
df = df[~df['fuel'].isin(['LPG', 'CNG'])]
df['fuel'].unique()
```

[20]: array(['Diesel', 'Petrol'], dtype=object)

2.1.5 - removing unit 'kmpl' in 'milage' feature and convert it into numerical feature

[21]:
```
# Replace 'kmpl' with blank
df.loc[df['mileage'].str.split(" ").str[1] == 'kmpl', 'mileage'] = df.
 ↪loc[df['mileage'].str.split(" ").str[1] == 'kmpl', 'mileage'].str.replace("␣
 ↪kmpl", "")
```

[22]:
```
# Convert mileage feature to float
df['mileage'] = df['mileage'].astype(float)
```

2.1.6 - removing unit 'CC' in 'engine' feature and convert it into numerical feature

[23]:
```
# Replace 'CC' with blank
df.loc[df['engine'].str.split(" ").str[1] == 'CC', 'engine'] = df.
 ↪loc[df['engine'].str.split(" ").str[1] == 'CC', 'engine'].str.replace(" CC",␣
 ↪"")
```

[24]:
```
# Convert engine feature to float
df['engine'] = df['engine'].astype(float)
```

2.1.7 - removing unit 'bhp' in 'max_power' feature and convert it into numerical feature

[25]:
```
# Replace 'bhp' with blank
df.loc[df['max_power'].str.split(" ").str[1] == 'bhp', 'max_power'] = df.
 ↪loc[df['max_power'].str.split(" ").str[1] == 'bhp', 'max_power'].str.
 ↪replace(" bhp", "")
```

[26]:
```
# Convert max_power feature to float
df['max_power'] = df['max_power'].astype(float)
```

2.1.8 - We will transform the name feature to brand feature by taking the first word of names

[27]:
```
# rename the column
df.rename(columns = {'name':'brand'}, inplace = True)
```

[28]:
```
# Taking the first word of name
df['brand'] = df['brand'].str.split(" ").str[0]
```

[29]:
```
# Checking clean data
df.head()
```

[29]:
```
     brand  year  selling_price  km_driven    fuel seller_type transmission  \
0   Maruti  2014         450000     145500  Diesel  Individual       Manual
1    Skoda  2014         370000     120000  Diesel  Individual       Manual
```

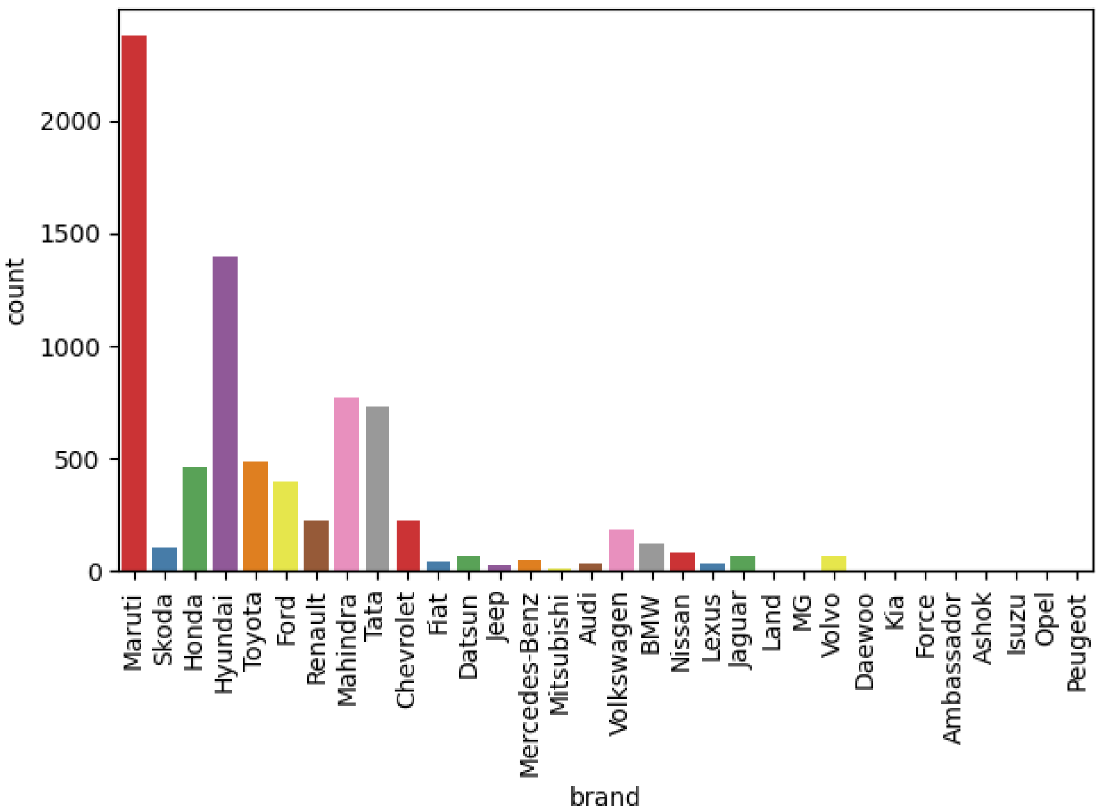
```
2    Honda  2006         158000     140000  Petrol  Individual        Manual
3  Hyundai  2010         225000     127000  Diesel  Individual        Manual
4   Maruti  2007         130000     120000  Petrol  Individual        Manual

   owner  mileage  engine  max_power  seats
0      1    23.40  1248.0      74.00    5.0
1      2    21.14  1498.0     103.52    5.0
2      3    17.70  1497.0      78.00    5.0
3      1    23.00  1396.0      90.00    5.0
4      1    16.10  1298.0      88.20    5.0
```

1.3.2 2.2 Univariate analysis

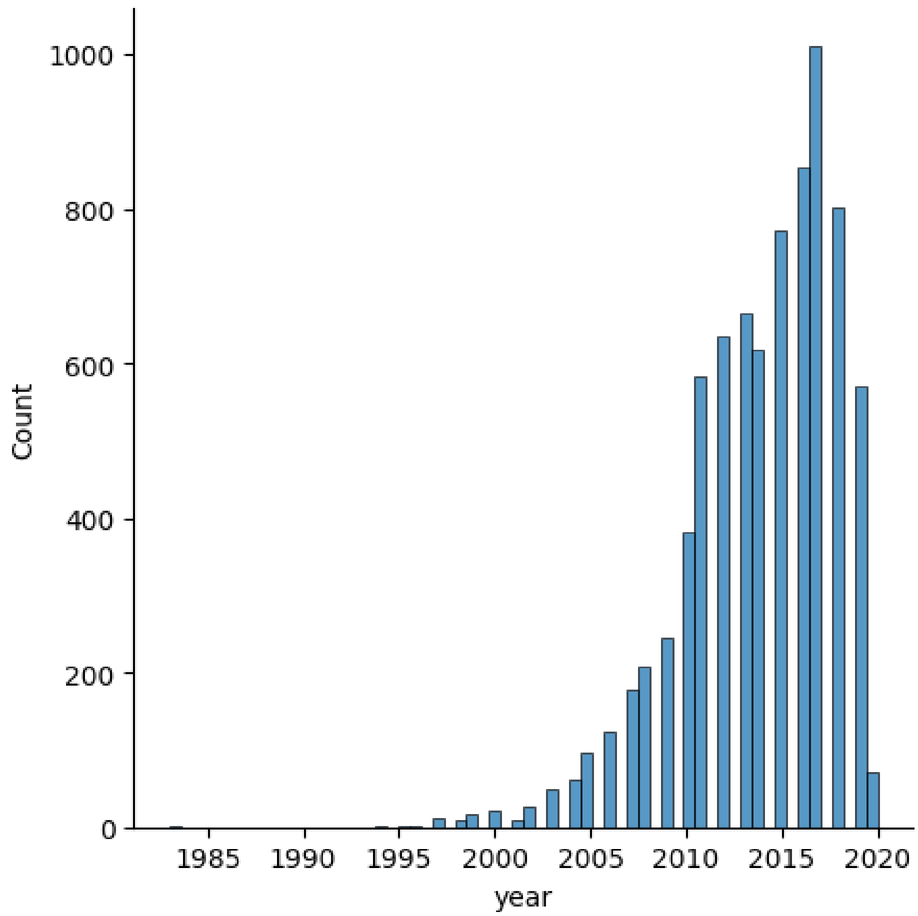
Single variable exploratory data anlaysis

Checking distributions

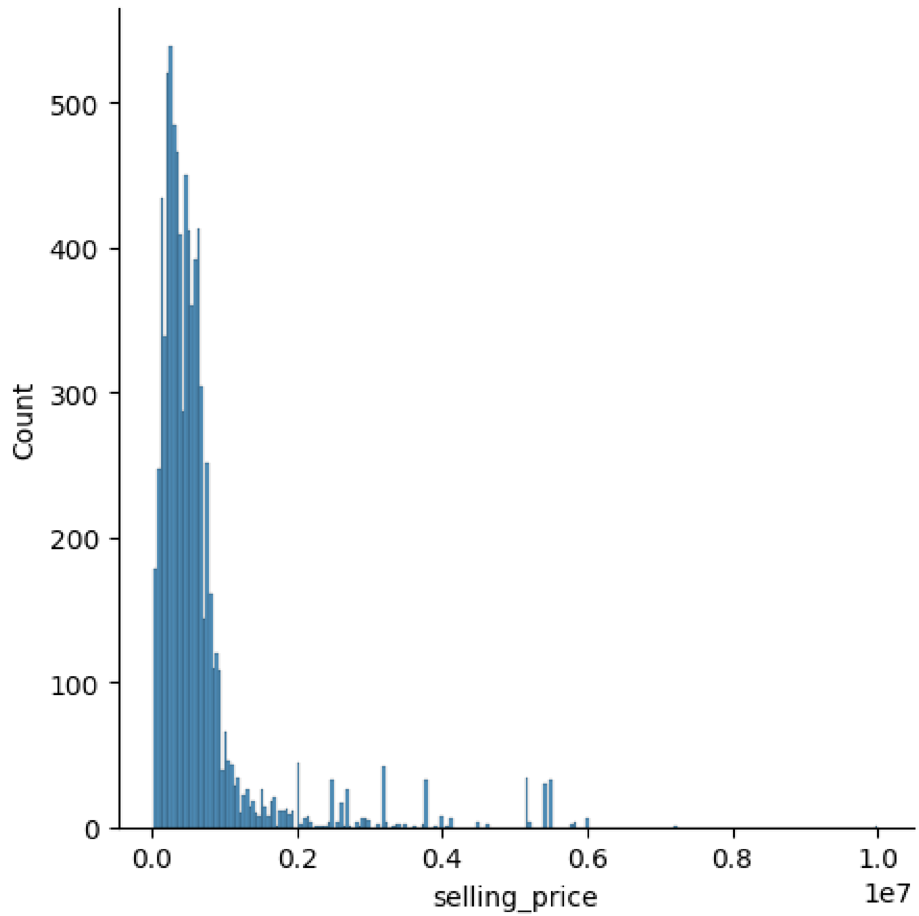
[30]:
```python
for j,i in enumerate(df.columns):
    if i == 'brand':
        ax = sns.countplot(data = df, x = i,palette= 'Set1')
        ax.set_xticklabels(df[i].unique(),rotation=90)
        plt.tight_layout()
        plt.show()
    else:
        sns.displot(data=df,x=df[i])
        plt.show()
```



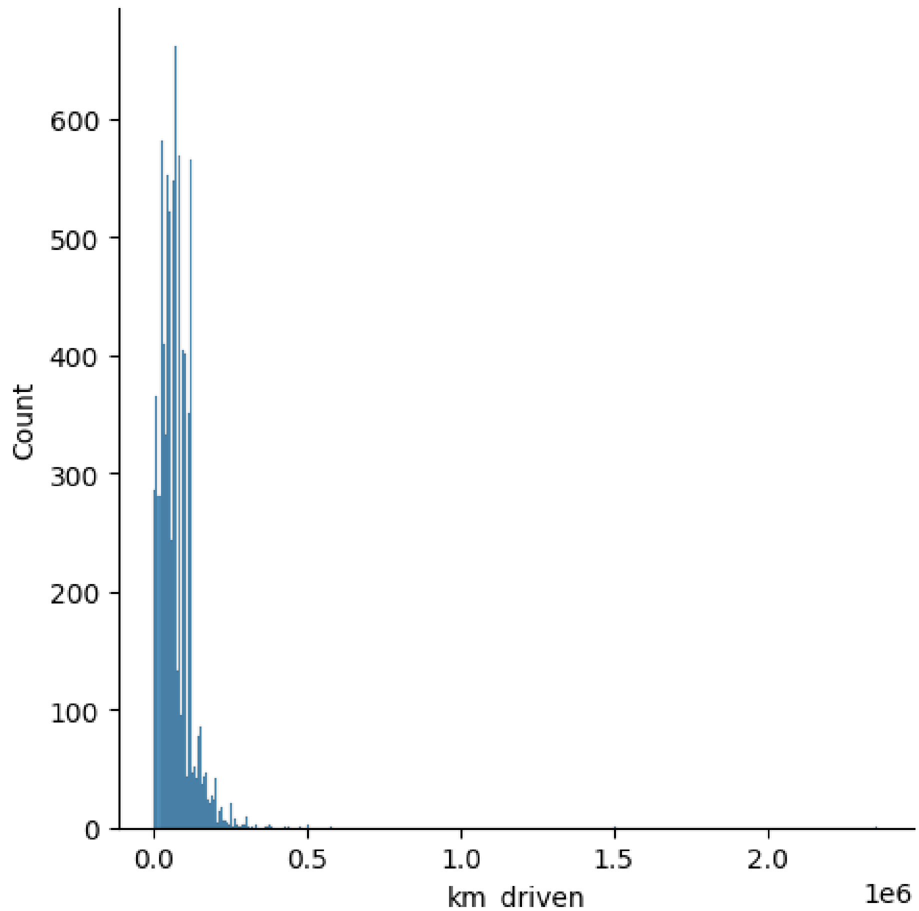
count
2000
1500
1000
500
0
Maruti
Skoda
Honda
Hyundai
Toyota
Ford
Renault
Mahindra
Tata
Chevrolet
Fiat
Datsun
Jeep
Mercedes-Benz
Mitsubishi
Audi
Volkswagen
BMW
Nissan
Lexus
Jaguar
Land
MG
Volvo
Daewoo
Kia
Force
Ambassador
Ashok
Isuzu
Opel
Peugeot
brand

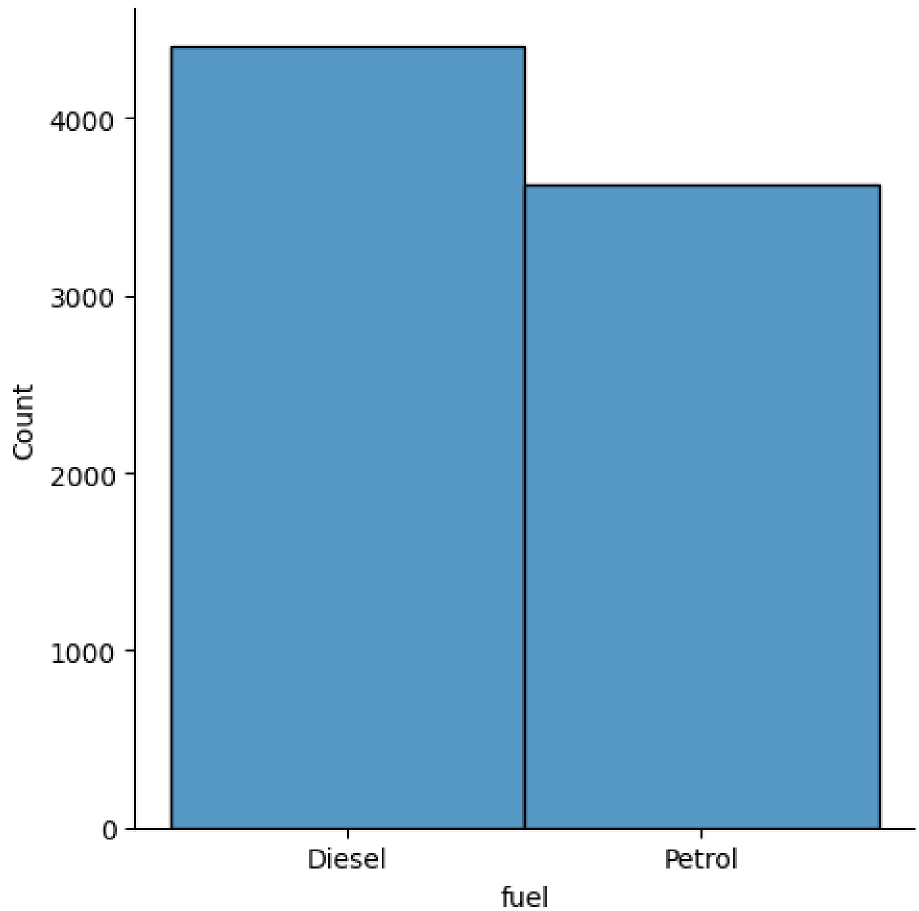


1000
800
600
400
200
0
Count
1985
1990
1995
2000
2005
2010
2015
2020
year

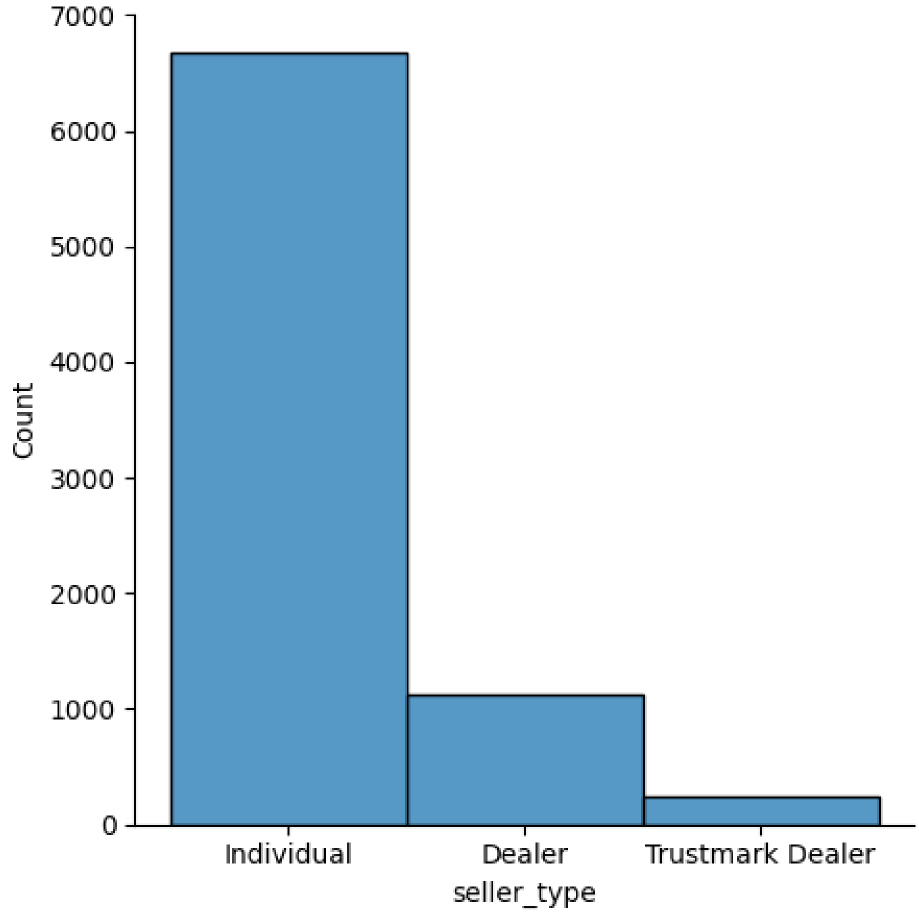


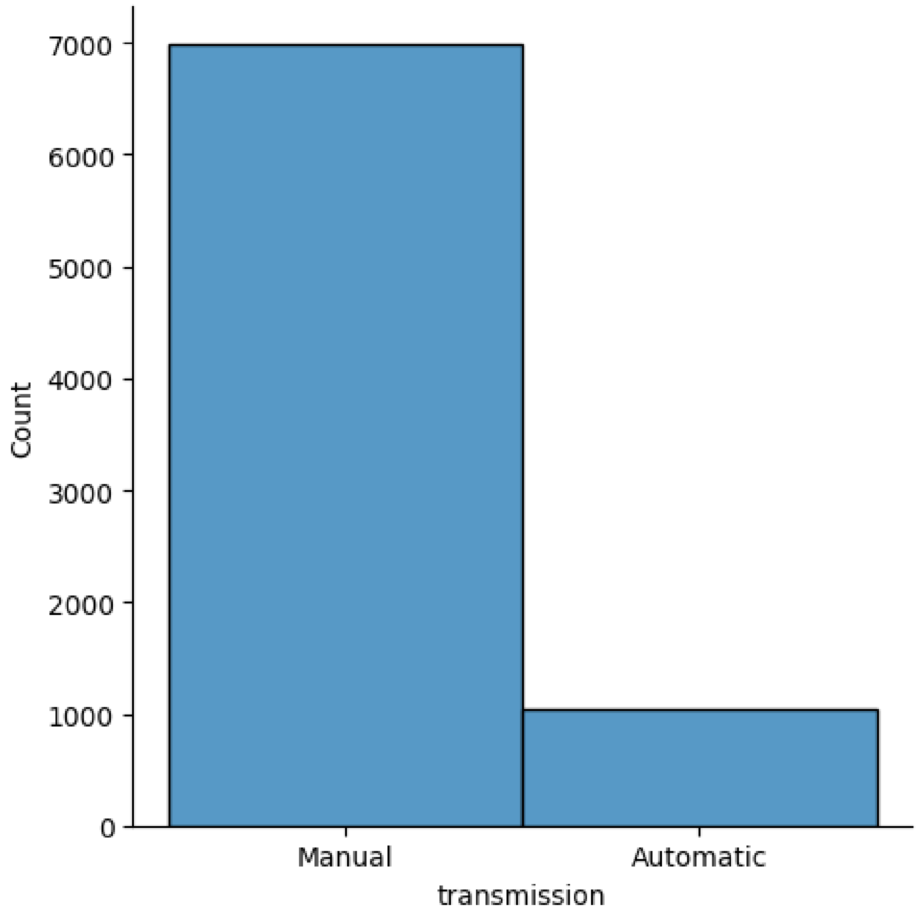
500
400
300
200
100
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
1e7
selling_price


Count
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
1e6
km_driven

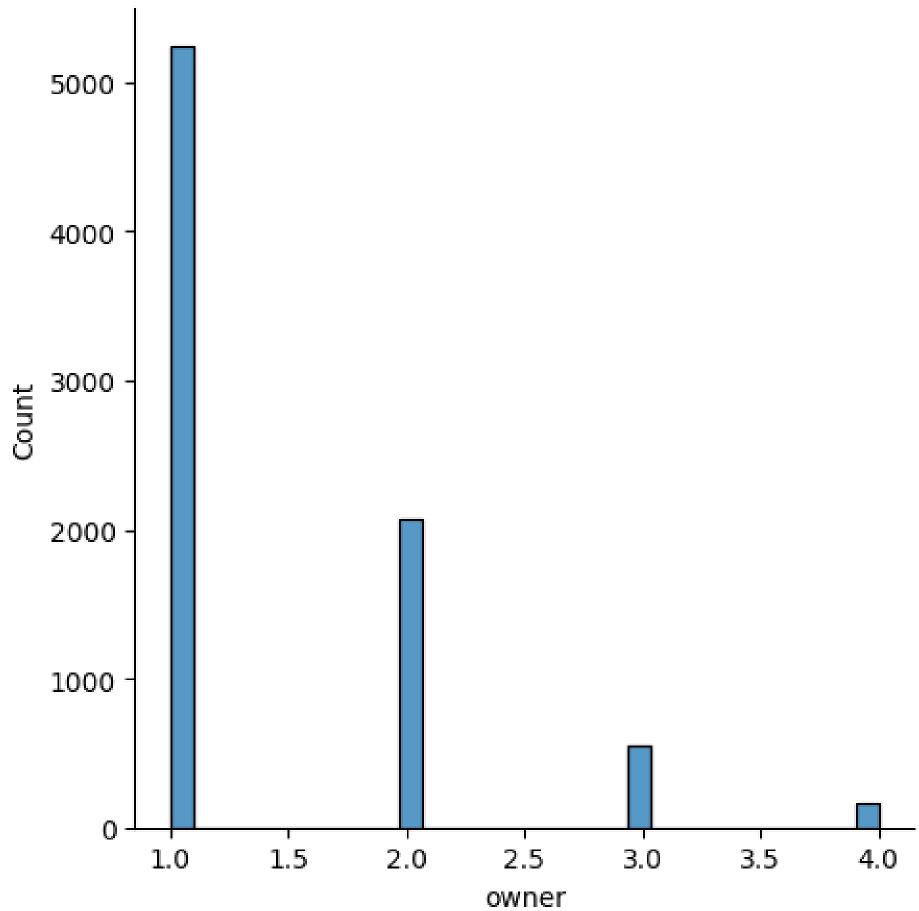


4000
3000
2000
1000
0
Count
Diesel
Petrol
fuel

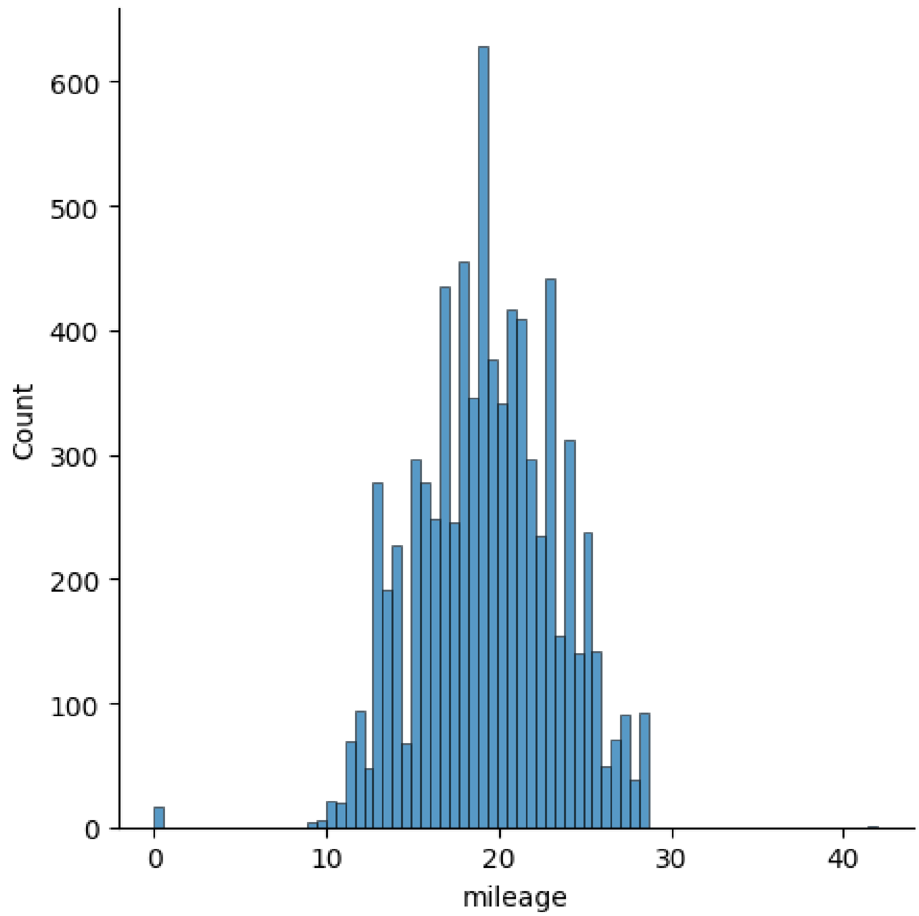

7000
6000
5000
4000
3000
2000
1000
0
Count
Individual
Dealer
Trustmark Dealer
seller_type

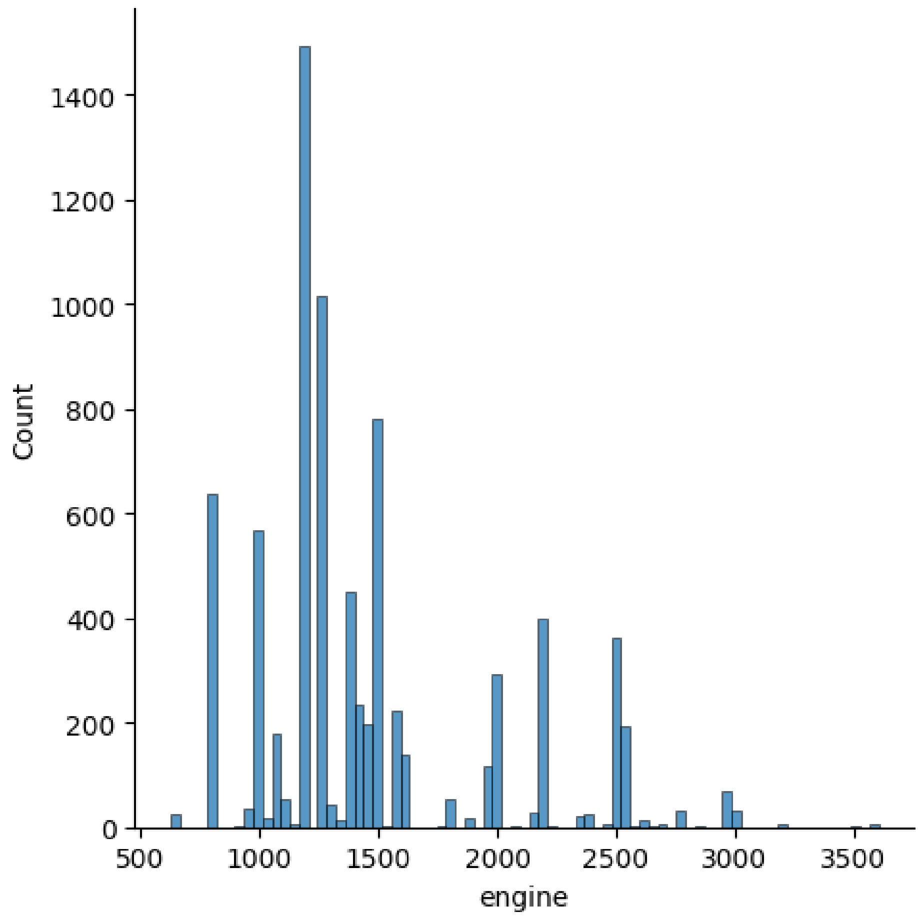


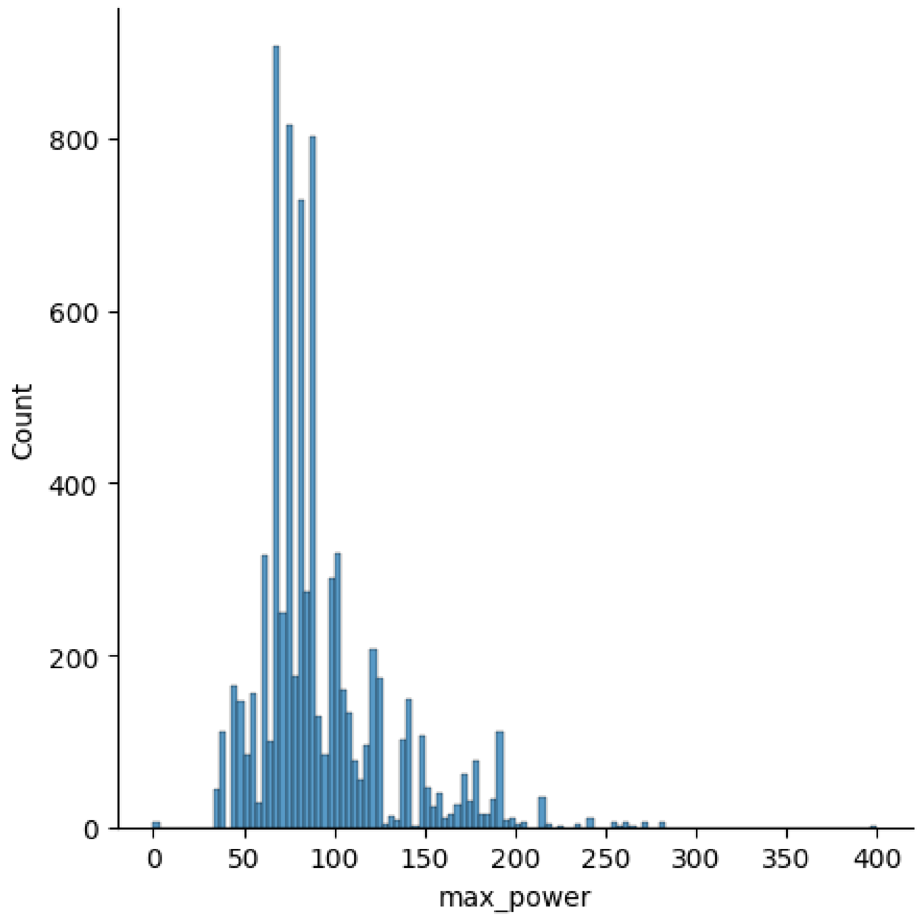
7000
6000
5000
4000
3000
2000
1000
0
Count
Manual
Automatic
transmission

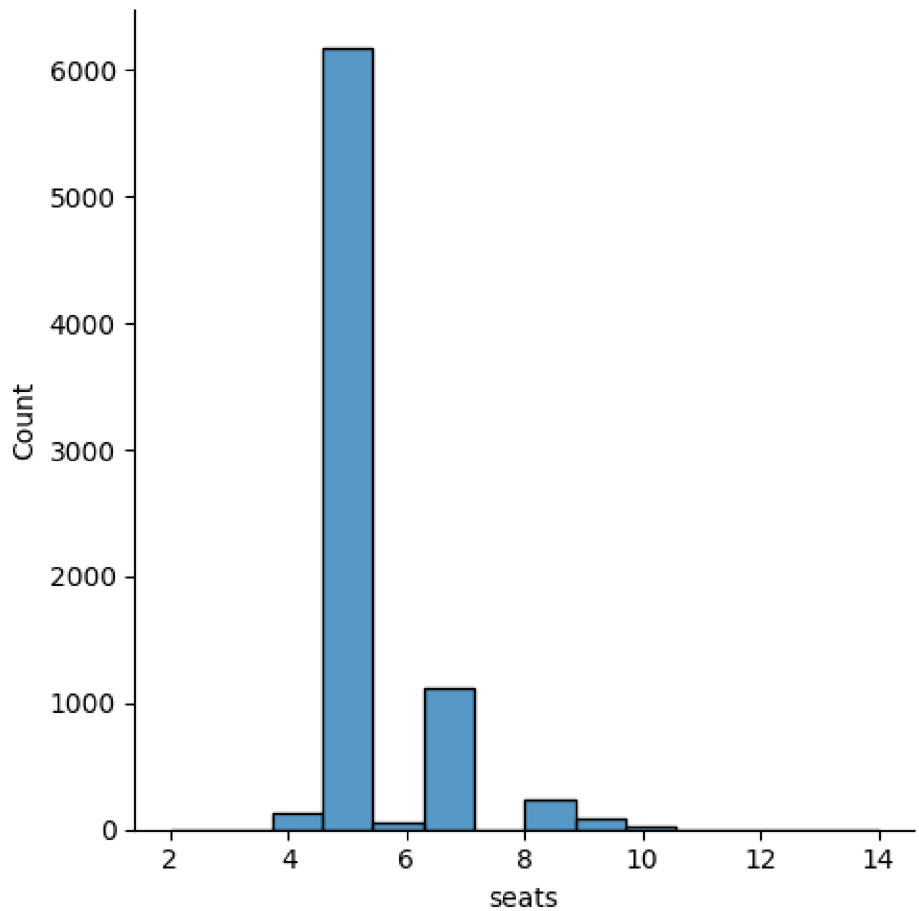


5000
4000
3000
2000
1000
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
owner


600
500
400
300
200
100
0
Count
0
10
20
30
40
mileage


1400
1200
1000
800
600
400
200
0
Count
500
1000
1500
2000
2500
3000
3500
engine


Count
800
600
400
200
0
0
50
100
150
200
250
300
350
400
max_power



1.3.3 2.3 Multivariate analysis

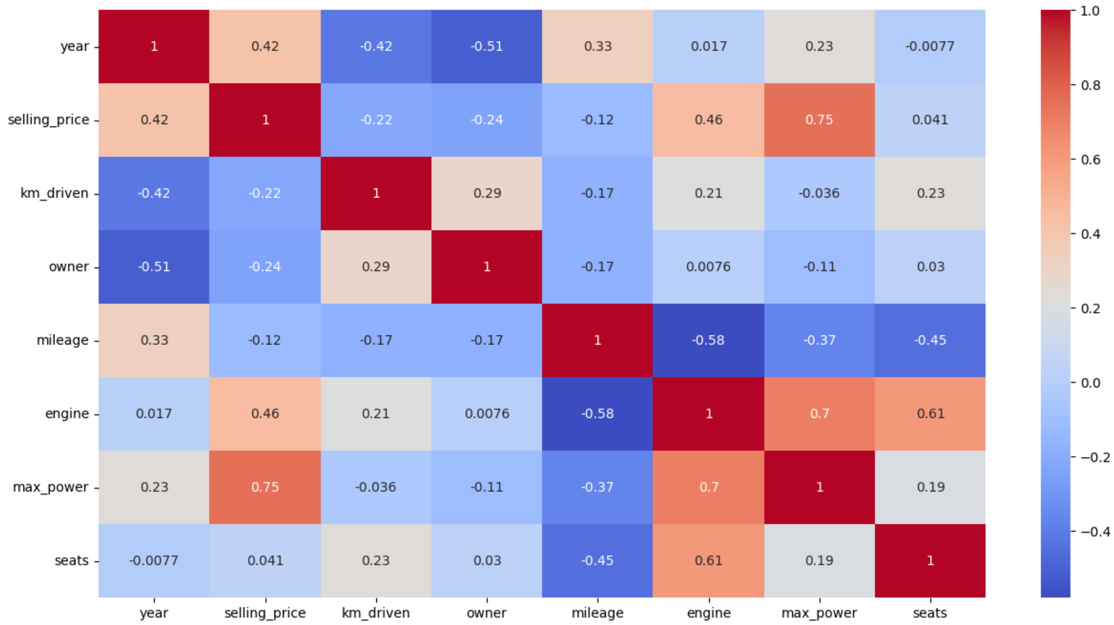
Multiple variable exploratory data analysis

Correlation Matrix Let's use correlation matrix to find strong factors predicting car price. It's also for checking whether certain features are too correlated.

[31]:
```
df_corr = df[[ 'year', 'selling_price', 'km_driven', 'owner', 'mileage',␣
↪'engine', 'max_power', 'seats']]
```

[32]:
```
# Let's check out heatmap
plt.figure(figsize = (15,8))
sns.heatmap(df_corr.corr(), annot=True, cmap="coolwarm")  #don't forget these␣
↪are not all variables! categorical is not here...
```

[32]: <Axes: >



Label encoding Apply Label encoding to categorical features to see the correlations

[33]:
```
# check unique values of four categorical features

print(f"brand : {len(df['brand'].unique())}")
print(f"fuel : {len(df['fuel'].unique())}")
print(f"transmission : {len(df['transmission'].unique())}")
print(f"seller_type : {len(df['seller_type'].unique())}")
```

```
brand : 32
fuel : 2
transmission : 2
seller_type : 3
```

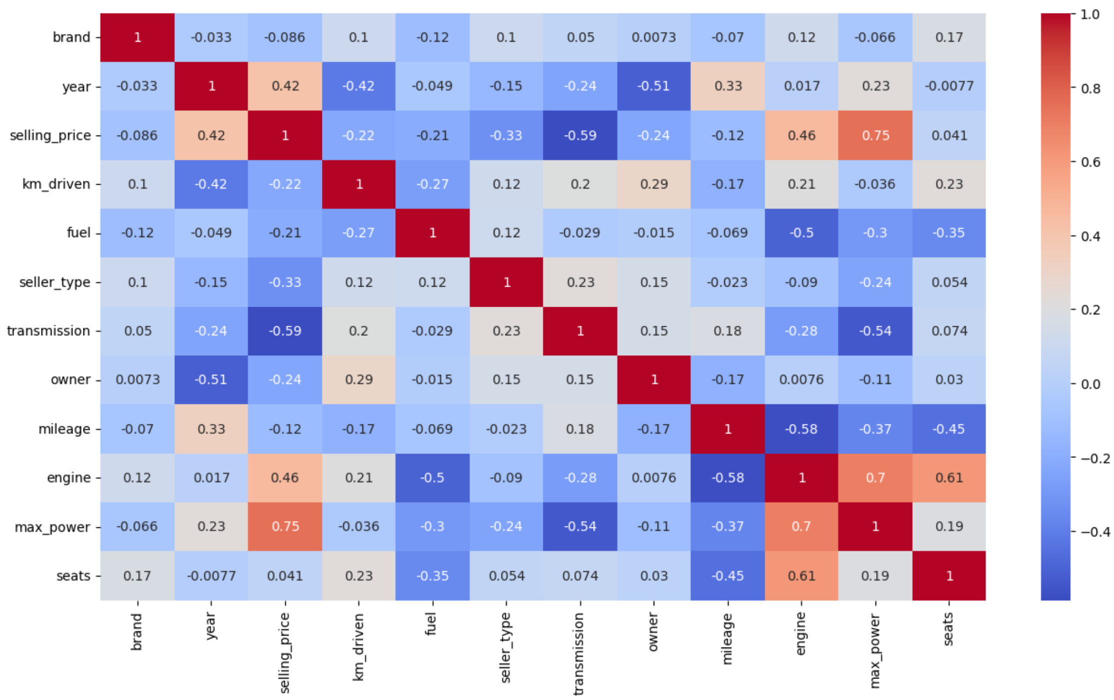
[34]:
```
# apply label encoding to check corelation
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df["fuel"] = le.fit_transform(df["fuel"])
df["transmission"] = le.fit_transform(df["transmission"])
df["brand"] = le.fit_transform(df["brand"])
df["seller_type"] = le.fit_transform(df["seller_type"])

print(df["fuel"].unique(),df["transmission"].unique())
print(df["brand"].unique(),df["seller_type"].unique())
```

```
[0 1] [1 0]
```

```
[20 27 10 11 29  9 26 19 28  4  7  6 14 21 22  2 30  3 23 17 13 16 18 31
  5 15  8  0  1 12 24 25] [1 0 2]
```

[35]:
```python
# Let's check out heatmap
plt.figure(figsize = (15,8))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")  #don't forget these are␣
↪not all variables! categorical is not here...
```

[35]: `<Axes: >`

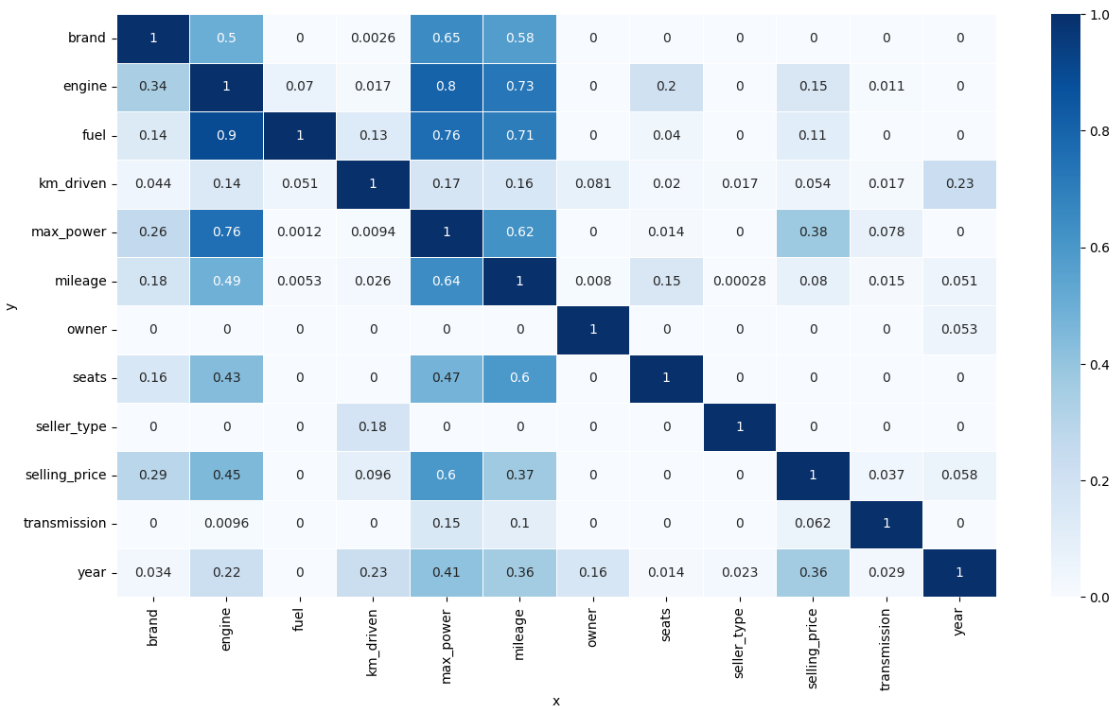


Predictive Power Score

[36]:
```python
import ppscore as pps

# before using pps, let's drop country and year
dfcopy = df.copy()
# dfcopy.drop(['country', 'year'], axis='columns', inplace=True)
# t = pps.matrix(dfcopy)[['x', 'y', 'ppscore']]
# t.pivot(index='x',columns='y',values='ppscore')
#this needs some minor preprocessing because seaborn.heatmap unfortunately does␣
↪not accept tidy data
matrix_df = pps.matrix(dfcopy)[['x', 'y', 'ppscore']].pivot(columns='x',␣
↪index='y', values='ppscore')

#plot
```

```python
plt.figure(figsize = (15,8))
sns.heatmap(matrix_df, vmin=0, vmax=1, cmap="Blues", linewidths=0.5, annot=True)
```

[36]: <Axes: xlabel='x', ylabel='y'>



The brand feature will not be chosen to train model as it does not explain much to selling price. The seller_type has three unique categories. So, this feature will be one-hot encoded.

[37]:
```python
df['seller_type'] = le.inverse_transform(df['seller_type'] )
```

[38]:
```python
df.head()
```

[38]:
```
   brand  year  selling_price  km_driven  fuel seller_type  transmission  \
0     20  2014         450000     145500     0  Individual             1
1     27  2014         370000     120000     0  Individual             1
2     10  2006         158000     140000     1  Individual             1
3     11  2010         225000     127000     0  Individual             1
4     20  2007         130000     120000     1  Individual             1

   owner  mileage  engine  max_power  seats
0      1    23.40  1248.0      74.00    5.0
1      2    21.14  1498.0     103.52    5.0
2      3    17.70  1497.0      78.00    5.0
```

```
3       1    23.00  1396.0       90.00    5.0
4       1    16.10  1298.0       88.20    5.0
```

[39]:
```python
df = pd.get_dummies(data=df,columns=['seller_type'],drop_first=True,dtype=int)
```

[40]:
```python
df.head()
```

[40]:
```
   brand  year  selling_price  km_driven  fuel  transmission  owner  mileage  \
0     20  2014         450000     145500     0             1      1    23.40
1     27  2014         370000     120000     0             1      2    21.14
2     10  2006         158000     140000     1             1      3    17.70
3     11  2010         225000     127000     0             1      1    23.00
4     20  2007         130000     120000     1             1      1    16.10

   engine  max_power  seats  seller_type_Individual  \
0  1248.0      74.00    5.0                       1
1  1498.0     103.52    5.0                       1
2  1497.0      78.00    5.0                       1
3  1396.0      90.00    5.0                       1
4  1298.0      88.20    5.0                       1

   seller_type_Trustmark Dealer
0                             0
1                             0
2                             0
3                             0
4                             0
```

[41]:
```python
# normalizing the feature
df['year'] = df['year'].apply(lambda x : (x - 1886) / (2024-1886) )
```

[42]:
```python
# binning the target
df['selling_price'] = pd.cut(df['selling_price'],4,labels=False)
```

[43]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8028 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   brand                         8028 non-null   int64
 1   year                          8028 non-null   float64
 2   selling_price                 8028 non-null   int64
 3   km_driven                     8028 non-null   int64
 4   fuel                          8028 non-null   int64
 5   transmission                  8028 non-null   int64
```

```
 6   owner                         8028 non-null   int64
 7   mileage                       7814 non-null   float64
 8   engine                        7814 non-null   float64
 9   max_power                     7820 non-null   float64
 10  seats                         7814 non-null   float64
 11  seller_type_Individual        8028 non-null   int64
 12  seller_type_Trustmark Dealer  8028 non-null   int64
dtypes: float64(5), int64(8)
memory usage: 878.1 KB
```

[44]:
```
df.head()
```

[44]:
```
   brand      year  selling_price  km_driven  fuel  transmission  owner  \
0     20  0.927536              0     145500     0             1      1
1     27  0.927536              0     120000     0             1      2
2     10  0.869565              0     140000     1             1      3
3     11  0.898551              0     127000     0             1      1
4     20  0.876812              0     120000     1             1      1

   mileage  engine  max_power  seats  seller_type_Individual  \
0    23.40  1248.0      74.00    5.0                       1
1    21.14  1498.0     103.52    5.0                       1
2    17.70  1497.0      78.00    5.0                       1
3    23.00  1396.0      90.00    5.0                       1
4    16.10  1298.0      88.20    5.0                       1

   seller_type_Trustmark Dealer
0                             0
1                             0
2                             0
3                             0
4                             0
```

1.4 3. Feature Engineering

We gonna skip Feature Engineering for now

1.5 4. Feature selection

According to the correlation matrix

- seats and brand have less than 0.2 of correlation scores with selling_price(target). So, these features will be removed.
- max_power and engine are highly correlated with 0.7. But, I will keep them both as high max_power with small engine might explain built quality of car (in terms of power losses due to friction)

According to the predictive power score

- transmission, owner, seller_type and km_driven has less than 0.1 score.

- But, only km_driven will be removed as the others have fair correlation scores with selling_price and also with assumption of they are somewhat important and nuanced features.

[45]:
```
df.columns
```

[45]:
```
Index(['brand', 'year', 'selling_price', 'km_driven', 'fuel', 'transmission',
       'owner', 'mileage', 'engine', 'max_power', 'seats',
       'seller_type_Individual', 'seller_type_Trustmark Dealer'],
      dtype='object')
```

[46]:
```
#x is our strong features
X = df[['year', 'fuel', 'seller_type_Individual', 'seller_type_Trustmark␣
 ↪Dealer',
        'transmission', 'owner', 'engine', 'max_power']]

#y is simply selling price
y = df["selling_price"]
```

[47]:
```
# check if y has missing values to remove
df["selling_price"].isna().sum()
```

[47]:
```
0
```

1.5.1 Train test split

[48]:
```
df.shape
```

[48]:
```
(8028, 13)
```

[49]:
```
# I assume only 10 percent of test data will be enough as it is around 800.
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1,␣
 ↪random_state = 42)
```

1.6 5. Preprocessing

1.6.1 Null values

[50]:
```
#check for null values
X_train.isna().sum()
```

[50]:
```
year                            0
fuel                            0
seller_type_Individual          0
seller_type_Trustmark Dealer    0
transmission                    0
```

```
owner                          0
engine                       199
max_power                    193
dtype: int64
```

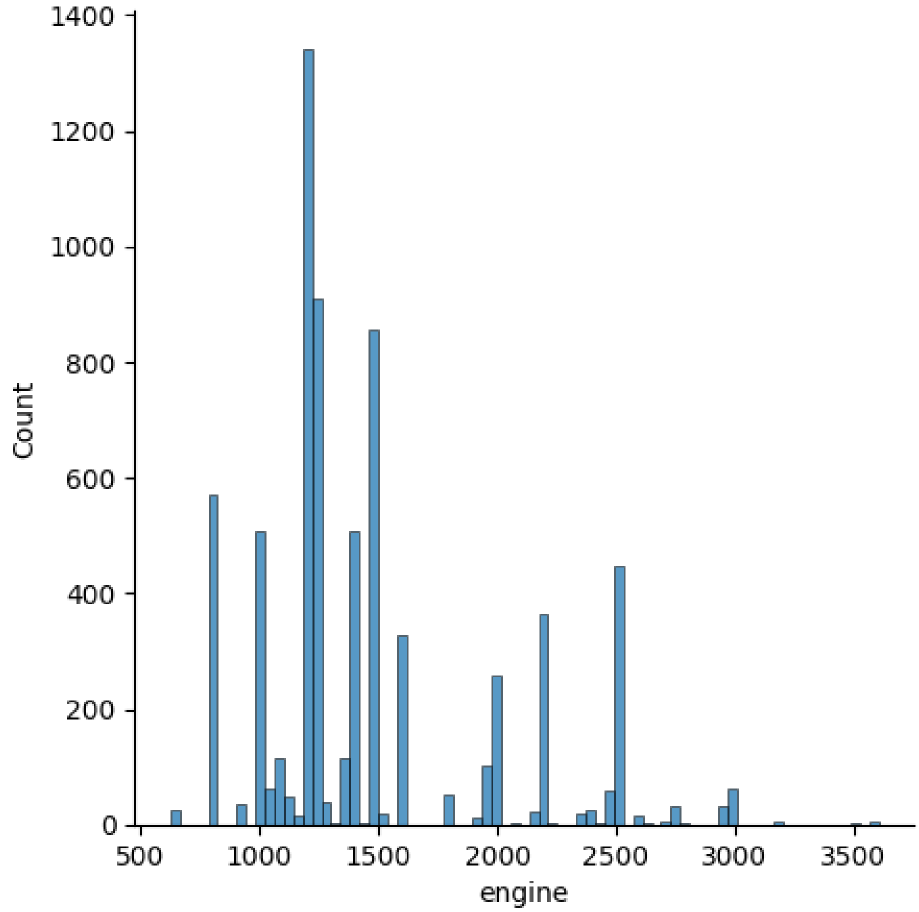
[51]:
```
X_test.isna().sum()
```

[51]:
```
year                         0
fuel                         0
seller_type_Individual       0
seller_type_Trustmark Dealer 0
transmission                 0
owner                        0
engine                      15
max_power                   15
dtype: int64
```

As the distributions of engine and max_power has right skewness, median values will be used to replace

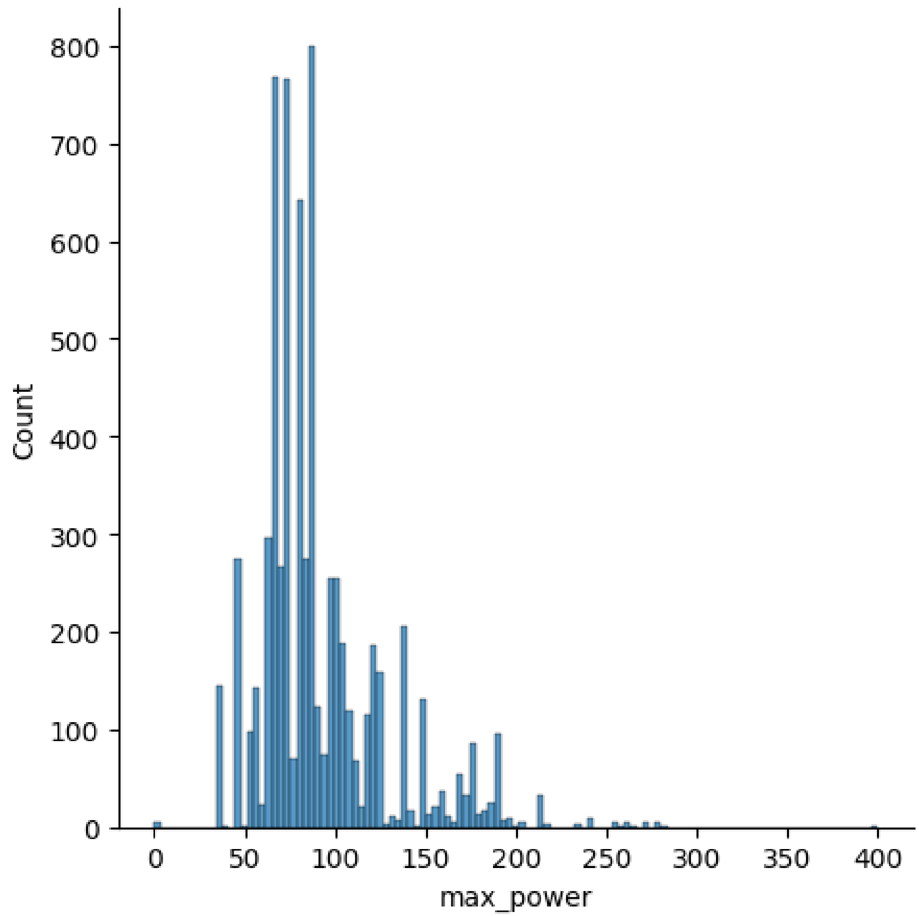
[52]:
```
sns.displot(data=X_train, x='engine')
```

[52]:
```
<seaborn.axisgrid.FacetGrid at 0x7f92c783f010>
```



[53]:
```python
sns.displot(data=X_train, x='max_power')
```

[53]: <seaborn.axisgrid.FacetGrid at 0x7f92c51b6bc0>



[54]:
```
#let's fill the testing set with the training distribution first!
# X_test['school'].fillna(X_train['school'].mean(), inplace=True)
X_train['engine'].fillna(X_train['engine'].median(), inplace=True)
X_train['max_power'].fillna(X_train['max_power'].median(), inplace=True)
```

[55]:
```
#let's fill the testing set with the training distribution first!
# X_test['school'].fillna(X_train['school'].mean(), inplace=True)
X_test['engine'].fillna(X_train['engine'].median(), inplace=True)
X_test['max_power'].fillna(X_train['max_power'].median(), inplace=True)
```

[56]:
```
#check again
X_train.isna().sum()
```

[56]:
```
year                            0
fuel                            0
seller_type_Individual          0
seller_type_Trustmark Dealer    0
transmission                    0
```

```
owner                         0
engine                        0
max_power                     0
dtype: int64
```

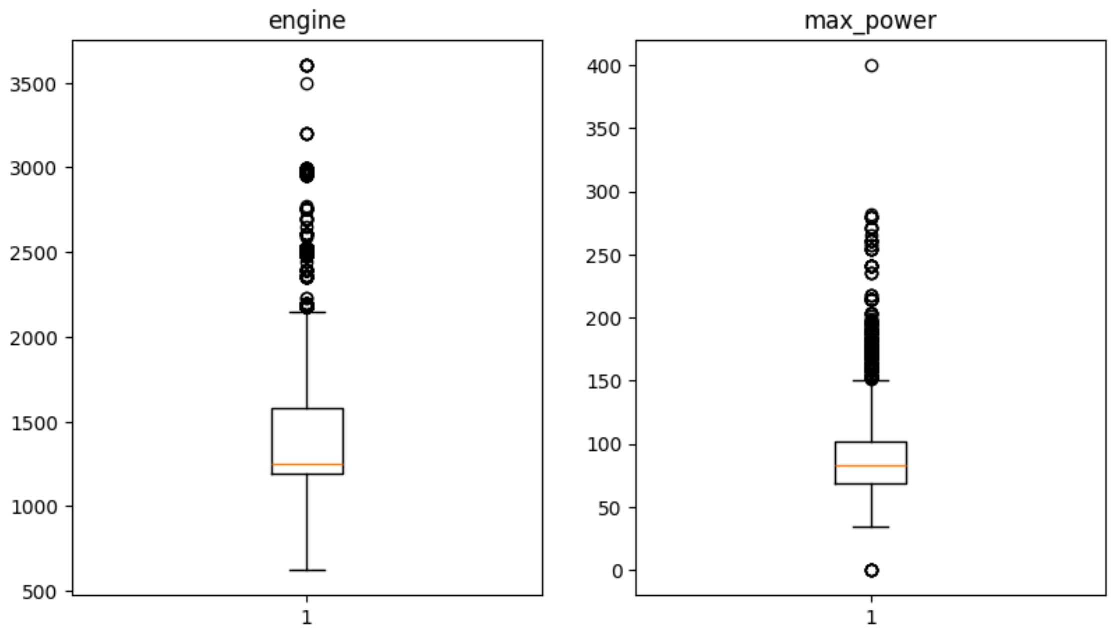
[57]:
```
X_test.isna().sum()
```

[57]:
```
year                          0
fuel                          0
seller_type_Individual        0
seller_type_Trustmark Dealer  0
transmission                  0
owner                         0
engine                        0
max_power                     0
dtype: int64
```

1.6.2 Checking Outliers

For two numerical features of engine and max__power, outliers need to be checked for scaling

[58]:
```
# Create a dictionary of columns.
col_dict = {'engine':1,'max_power':2}

# Detect outliers in each variable using box plots.
plt.figure(figsize=(20,30))

for variable,i in col_dict.items():
                     plt.subplot(5,4,i)
                     plt.boxplot(X_train[variable])
                     plt.title(variable)

plt.show()
```



[59]:
```
def outlier_count(col, data = X_train):

    # calculate your 25% quatile and 75% quatile
    q75, q25 = np.percentile(data[col], [75, 25])

    # calculate your inter quatile
    iqr = q75 - q25

    # min_val and max_val
    min_val = q25 - (iqr*1.5)
    max_val = q75 + (iqr*1.5)

    # count number of outliers, which are the data that are less than min_val␣
 ↪or more than max_val calculated above
    outlier_count = len(np.where((data[col] > max_val) | (data[col] <␣
 ↪min_val))[0])

    # calculate the percentage of the outliers
    outlier_percent = round(outlier_count/len(data[col])*100, 2)

    if(outlier_count > 0):
        print("\n"+15*'-' + col + 15*'-'+"\n")
        print('Number of outliers: {}'.format(outlier_count))
        print('Percent of data that is outlier: {}%'.format(outlier_percent))
```

[60]:
```
for col in ['engine','max_power']:
    outlier_count(col)
```

```
---------------engine---------------

Number of outliers: 1079
Percent of data that is outlier: 14.93%

---------------max_power---------------

Number of outliers: 519
Percent of data that is outlier: 7.18%
```

1.6.3 Scaling

As the features of engine and max_power has a considerable amount of outliers and skewness and they are not in a bounded range, standardization can be used to scale.

[61]:
```
from sklearn.preprocessing import StandardScaler

# feature scaling helps improve reach convergence faster
scaler = StandardScaler()
X_train[['engine','max_power']] = scaler.
 ↪fit_transform(X_train[['engine','max_power']])
X_test[['engine','max_power']]  = scaler.
 ↪transform(X_test[['engine','max_power']])
```

For selling_price(y), np.log will be used to scale

[62]:
```
from sklearn.preprocessing import OneHotEncoder
oh = OneHotEncoder(sparse_output=False)
oh.fit(y_train.to_numpy().reshape(-1,1))
Y_train = oh.transform(y_train.to_numpy().reshape(-1,1))
Y_test = oh.transform(y_test.to_numpy().reshape(-1,1))
```

[63]:
```
# Let's check shapes of all X_train, X_test, y_train, y_test
print("Shape of X_train: ", X_train.shape)
print("Shape of X_test: ", X_test.shape)
print("Shape of y_train: ", Y_train.shape)
print("Shape of y_test: ", Y_test.shape)
```

```
Shape of X_train:  (7225, 8)
Shape of X_test:  (803, 8)
Shape of y_train:  (7225, 4)
Shape of y_test:  (803, 4)
```

2 6. Modeling

Modifying the Multinomial Linear Regression class - add accuracy score function - add functions of precision, recall and f1-score for each class - add functions macro precision, macro recall and macro f1 - add functions weighted precision, weighted recall and weighted f1 - add ridge L2 penalty - compare the results of the implementation with scikit-learn classification report

[180]:
```
from sklearn.model_selection import KFold
import time
# from mlflow.pyfunc import PythonModel

class MultinomialLogisticRegression(object):

    kfold = KFold(n_splits=5)

    def __init__(self, k, n, method,class_weights = None,penalty = 0,cv=kfold,␣
↪alpha = 0.001, max_iter=5000):
        self.k = k
        self.n = n
        self.alpha = alpha
        self.max_iter = max_iter
        self.method = method
        self.cv = cv
        self.penalty = penalty
        self.class_weights =class_weights

    def fit_CV(self,X,Y):
        self.cv_metric = {}
        for fold, (train_idx, val_idx) in enumerate(self.cv.split(X)):
            X_cross_train = X[train_idx]
            Y_cross_train = Y[train_idx]
            X_cross_val   = X[val_idx]
            Y_cross_val   = np.argmax(Y[val_idx],axis=1)
            self.fit(X_cross_train,Y_cross_train)
            Y_hat = self.predict(X_cross_val)
            metric = {}
            metric['accuracy'] = self.accuracy(Y_cross_val,Y_hat)
            metric['precision'] = self.
↪macro_metric("precision",Y_cross_val,Y_hat,self.class_weights)
            metric['recall'] = self.
↪macro_metric("recall",Y_cross_val,Y_hat,self.class_weights)
            metric['f1_score'] = self.
↪macro_metric("f1_score",Y_cross_val,Y_hat,self.class_weights)
            self.cv_metric[fold] = metric
        print(self.cv_metric)

    def fit(self, X, Y):
```

```python
        self.W = np.random.rand(self.n, self.k)
        self.losses = []
        # Y = np.asarray(Y).copy()
        # print(Y.shape)
        if self.method == "batch":
            start_time = time.time()
            for i in range(self.max_iter):
                loss, grad =  self.gradient(X, Y)
                self.losses.append(loss)
                self.W = self.W - self.alpha * grad
                if i % 500 == 0:
                    print(f"Loss at iteration {i}", loss)
            print(f"time taken: {time.time() - start_time}")

        elif self.method == "minibatch":
            start_time = time.time()
            batch_size = int(0.3 * X.shape[0])
            for i in range(self.max_iter):
                ix = np.random.randint(0, X.shape[0]) #<----with replacement
                batch_X = X[ix:ix+batch_size]
                batch_Y = Y[ix:ix+batch_size]
                loss, grad = self.gradient(batch_X, batch_Y)
                self.losses.append(loss)
                self.W = self.W - self.alpha * grad
                if i % 500 == 0:
                    print(f"Loss at iteration {i}", loss)
            print(f"time taken: {time.time() - start_time}")

        elif self.method == "sto":
            start_time = time.time()
            list_of_used_ix = []
            for i in range(self.max_iter):
                idx = np.random.randint(X.shape[0])
                while i in list_of_used_ix:
                    idx = np.random.randint(X.shape[0])
                X_train = X[idx, :].reshape(1, -1)
                Y_train = Y[idx]
                loss, grad = self.gradient(X_train, Y_train)
                self.losses.append(loss)
                self.W = self.W - self.alpha * grad

                list_of_used_ix.append(i)
                if len(list_of_used_ix) == X.shape[0]:
                    list_of_used_ix = []
                if i % 500 == 0:
                    print(f"Loss at iteration {i}", loss)
            print(f"time taken: {time.time() - start_time}")
```

```python
        else:
            raise ValueError('Method must be one of the followings: "batch", "minibatch" or "sto".')


    def gradient(self, X, Y):
        m = X.shape[0]
        h = self.h_theta(X, self.W)
        loss = (- np.sum(np.multiply(Y,np.log(h)) / m)) + (self.penalty * np.sum(np.square(self.W)))
        print(loss)
        # print(type(h),type(Y))
        error = h - Y
        grad = self.softmax_grad(X, error)
        return loss, grad

    def softmax(self, theta_t_x):
        # print( np.sum(theta_t_x, axis=1, keepdims=True))
        return np.exp(theta_t_x) / np.sum(np.exp(theta_t_x), axis=1, keepdims=True)

    def softmax_grad(self, X, error):
        return  (X.T @ error) + (2*self.penalty*self.W)

    def h_theta(self, X, W):
        '''
        Input:
            X shape: (m, n)
            w shape: (n, k)
        Returns:
            yhat shape: (m, k)
        '''
        return self.softmax(X @ W)

    def predict(self, X_test):
        return np.argmax(self.h_theta(X_test, self.W), axis=1)

    def plot(self):
        plt.plot(np.arange(len(self.losses)) , self.losses, label = "Train Losses")
        plt.title("Losses")
        plt.xlabel("epoch")
        plt.ylabel("losses")
        plt.legend()

    def accuracy(self,Y_true,Y_pred):
```

```
        Y_true = Y_true.reshape(1,-1)[0]
        Y_pred = Y_pred.reshape(1,-1)[0]
        return len(Y_pred[Y_true == Y_pred]) /  len(Y_pred)

    def _confusion_class(self,Y_true,Y_pred,c):
        Y_true = Y_true.reshape(1,-1)[0]
        Y_pred = Y_pred.reshape(1,-1)[0]
        class_filter = (Y_true == c) | (Y_pred == c)
        Y_true_class = Y_true[class_filter]
        Y_pred_class = Y_pred[class_filter]
        TP  = len(Y_true_class[Y_true_class == Y_pred_class])
        FP  = len(Y_true_class[(Y_true_class != Y_pred_class)  & (Y_pred_class␣
↪== c)])
        FN  = len(Y_true_class[(Y_true_class != Y_pred_class)  & (Y_pred_class !
↪= c)])
        return (TP,FP,FN)

    def precision(self,Y_true,Y_pred,c):
        TP,FP,_ = self._confusion_class(Y_true,Y_pred,c)
        if  TP + FP == 0:
            p = 0
        else:
            p =  TP / (TP + FP)
        return p

    def recall(self,Y_true,Y_pred,c):
        TP,_,FN = self._confusion_class(Y_true,Y_pred,c)
        if  TP + FN == 0:
            r = 0
        else:
            r =  TP / (TP+FN)
        return r

    def f1_score(self,Y_true,Y_pred,c):
        p = self.precision(Y_true,Y_pred,c)
        r = self.recall(Y_true,Y_pred,c)
        if p + r == 0:
            f1 = 0
        else:
            f1 = 2 * p * r / (p + r)
        return f1

    def macro_metric(self,metric,Y_true,Y_pred,weighted=False):
        if metric == 'precision':
            macro_ = np.array([self.precision(Y_true,Y_pred,c) for c in np.
↪unique(Y_true)])
        elif metric == 'recall':
```

```
            macro_ =  np.array([self.recall(Y_true,Y_pred,c) for c in np.
↪unique(Y_true)])
        else:
            macro_  = np.array([self.f1_score(Y_true,Y_pred,c) for c in np.
↪unique(Y_true)])
        if  weighted == True:
            weights =  np.array([len(Y_true[Y_true == c]) / len(Y_true) for c␣
↪in np.unique(Y_true)])
            macro_m = np.sum(macro_ * weights)
        else:
            macro_m = np.mean(macro_)
        return macro_m
```

[181]:
```
lg = MultinomialLogisticRegression(k=4,n=8,method='batch',penalty=0.
↪9,max_iter=100)
```

[182]:
```
lg.fit(X=X_train.to_numpy(),Y=Y_train)
```

```
8.87144212951591
Loss at iteration 0 8.87144212951591
155.87475443443776
149.81715214429624
146.1302361616937
144.10900011274924
142.68564385989583
141.5641057571601
140.531489355333
139.5279114312616
138.54563921816782
137.57848900183723
136.6255591809052
135.68534182268783
134.75751056890186
133.84147332146648
132.93697233213496
132.0436556259019
131.16127768673687
130.28956408678266
129.42827939231364
128.57718280315194
127.73605127406994
126.90466451243033
126.08281280874756
125.27029176631186
124.46690486723817
123.67246150115216
```

122.88677771296975
122.10967537443146
121.34098230832615
120.58053186754942
119.82816285235158
119.08371924267195
118.34705005377513
117.61800913264881
116.89645500215045
116.18225068928768
115.4752635743434
114.77536523862328
114.08243132311745
113.39634139019232
112.71697879220277
112.04423054409395
111.37798720105083
110.71814273999232
110.06459444505694
109.41724279650695
108.77599136294361
108.14074669651659
107.51141823098338
106.88791818242899
106.27016145253049
105.65806553425408
105.05155041991155
104.45053851151688
103.85495453340805
103.26472544711226
102.67978036844754
102.1000504868638
101.52546898703407
100.95597097271255
100.39149339287776
99.83197497018232
99.27735613172716
98.72757894217767
98.18258703923414
97.64232557146389
97.10674113849637
96.57578173357598
96.04939668845924
95.52753662063608
95.01015338284672
94.49720001485794
93.98863069745464
93.4844007085963

```
92.9844663816798
92.48878506584505
91.99731508825323
91.51001571826376
91.02684713343054
90.5477703872362
90.07274737847885
89.60174082222589
89.13471422224684
88.67163184483815
88.21245869395213
87.75716048754515
87.30570363505971
86.85805521595942
86.41418295923711
85.97405522382039
85.53764097980229
85.10490979042798
84.67583179477334
84.25037769105467
83.8285187205133
83.41022665182298
82.99547376597278
82.58423284158104
82.1764771406019
time taken: 0.29651308059692383
```

[183]:
```python
y_hat = lg.predict(X_test=X_test.to_numpy())
```

[184]:
```python
lg.accuracy(Y_true=y_test.to_numpy(),Y_pred=y_hat)
```

[184]:
```
0.9713574097135741
```

[142]:
```python
for i in range(lg.k):
    print (f" Metric for the class {i}")
    print("="*30)
    print(f"Precision  : {lg.precision(Y_true=y_test.
↪to_numpy(),Y_pred=y_hat,c=i)}")
    print(f"Recall  : {lg.recall(Y_true=y_test.to_numpy(),Y_pred=y_hat,c=i)}")
    print(f"f1 score : {lg.f1_score(Y_true=y_test.to_numpy(),Y_pred=y_hat,c=i)}␣
↪\n")
```

```
 Metric for the class 0
==============================
Precision  : 0.9960681520314548
Recall  : 0.9921671018276762
f1 score : 0.9941137998691956
```

```
 Metric for the class 1
==============================
Precision  : 0.5142857142857142
Recall  : 0.782608695652174
f1 score : 0.6206896551724138

 Metric for the class 2
==============================
Precision  : 0.4
Recall  : 0.14285714285714285
f1 score : 0.21052631578947364

 Metric for the class 3
==============================
Precision  : 0
Recall  : 0
f1 score : 0
```

[143]:

```
print (f" Macro Metric")
print("="*30)
# print(lg.macro_metric(metric="precision",Y_true=y_test.
 ↪to_numpy(),Y_pred=y_hat))
p =  lg.macro_metric(metric="precision",Y_true=y_test.to_numpy(),Y_pred=y_hat)
r =   lg.macro_metric(metric="recall",Y_true=y_test.to_numpy(),Y_pred=y_hat)
f1  =  lg.macro_metric(metric="f1",Y_true=y_test.to_numpy(),Y_pred=y_hat)
print(f"Precision : {p}" )
print(f"Recall : {r}")
print(f"F1_score : {f1}")
```

```
 Macro Metric
==============================
Precision : 0.636784622105723
Recall : 0.639210980112331
F1_score : 0.6084432569436944
```

[144]:

```
print (f" Macro Metric with class weights")
print("="*30)
# print(lg.macro_metric(metric="precision",Y_true=y_test.
 ↪to_numpy(),Y_pred=y_hat))
p =  lg.macro_metric(metric="precision",Y_true=y_test.
 ↪to_numpy(),Y_pred=y_hat,weighted=True)
r =   lg.macro_metric(metric="recall",Y_true=y_test.
 ↪to_numpy(),Y_pred=y_hat,weighted=True)
f1  =  lg.macro_metric(metric="f1",Y_true=y_test.
 ↪to_numpy(),Y_pred=y_hat,weighted=True)
print(f"Precision : {p}" )
```

```
print(f"Recall : {r}")
print(f"F1_score : {f1}")
```

```
 Macro Metric with class weights
==============================
Precision : 0.9718764332312152
Recall : 0.9713574097135741
F1_score : 0.9697564149312852
```

[145]:
```
from sklearn.metrics import classification_report
print(classification_report(y_true=y_test,y_pred=y_hat))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       766
           1       0.51      0.78      0.62        23
           2       0.40      0.14      0.21        14

    accuracy                           0.97       803
   macro avg       0.64      0.64      0.61       803
weighted avg       0.97      0.97      0.97       803
```

The resluts of the implementations are almost the same with scikit-learn classification report, Here, support in the classification report is the number of samples that are used the calculating the metrics.

[198]:
```
# Creating experiment in MLflow
import mlflow
import os
mlflow.set_tracking_uri("http://mlflow.ml.brain.cs.ait.ac.th/")
os.environ['MLFLOW_TRACKING_USERNAME'] = 'admin'
os.environ['MLFLOW_TRACKING_PASSWORD'] = 'password'
os.environ['LOGNAME'] = 'st125066'
mlflow.set_experiment(experiment_name="st125066-a3")
```

[198]:
```
<Experiment: artifact_location='mlflow-artifacts:/881703814181867940',
creation_time=1728266589594, experiment_id='881703814181867940',
last_update_time=1728266589594, lifecycle_stage='active', name='st125066-a3',
tags={}>
```

[185]:
```
lg_cv = MultinomialLogisticRegression(k=4,n=8,method='batch',penalty=0.
 ↪9,max_iter=100)
```

[186]:
```
lg_cv.fit_CV(X=X_train.to_numpy(),Y=Y_train)
```

```
8.338207698921847
Loss at iteration 0 8.338207698921847
```

42.234760894393595
40.482614711123084
39.92354706282143
39.65463773733321
39.43841178788658
39.23088181611018
39.03307208154957
38.84207333011642
38.658160842817765
38.48064672057007
38.30930698715787
38.14376782032171
37.98374762447671
37.82895432109455
37.679126048986696
37.53401235497755
37.39338050908552
37.25701130540758
37.12469943991267
36.99625203097063
36.87148803219444
36.750237349469195
36.63234016867562
36.5176462687246
36.40601441165748
36.29731176422165
36.19141336540753
36.08820162868749
35.987565879322055
35.88940192243722
35.793611639991916
35.700102614061564
35.60878777449563
35.51958506903189
35.43241715420981
35.34721110555462
35.26389814566305
35.18241338894374
35.1026956018816
35.02468697779489
34.94833292514277
34.87358186851962
34.80038506154202
34.7286964108961
34.65847231086796
34.589671487729106
34.52225485339311
34.456185367799186

34.391427909514576
34.327949154080244
34.26571745965417
34.20470275953398
34.1448764611656
34.08621135126821
34.02868150672683
33.97226221092454
33.916929875204474
33.86266196516947
33.80943693154383
33.757234145336525
33.70603383706037
33.655817039774554
33.606565535731555
33.55826180642083
33.510888985813864
33.464430816625274
33.41887160941555
33.37419620437012
33.3303899355989
33.28743859780898
33.24532841521123
33.204046012529524
33.163578387988096
33.1239128881601
33.085037184566204
33.04693925191878
33.00960734791271
32.97302999446953
32.937195960346436
32.90209424502709
32.86771406381524
32.834044834057025
32.80107616242132
32.768797833172236
32.737199797370444
32.70627216294464
32.67600518557676
32.64638926034819
32.61741491409707
32.58907279843937
32.56135368340927
32.53424845167651
32.50774809330131
32.48184370098897
32.45652646580891
32.431787673344566

```
32.407618700242786
32.38401101113269
32.360956155886086
time taken: 0.20027804374694824
10.174126152719026
Loss at iteration 0 10.174126152719026
72.52681379272344
68.00874630703092
65.15172686005431
63.826419773979424
63.21177111079366
62.69805488573495
62.219553655426566
61.75540627045165
61.30176172060481
60.8560854104384
60.41775652121225
59.98624127111143
59.561345958398206
59.14287753949872
58.73070876184156
58.324709458159006
57.92476290519895
57.530751867375336
57.14256283917132
56.76008335486252
56.383203180368255
56.011813891694835
55.64580925684027
55.28508522461948
54.92954006219341
54.57907438518792
54.23359120353874
53.89299593088073
53.55719638691317
53.22610278373376
52.899627704219085
52.57768607198911
52.260195115831785
51.94707432944561
51.63824542786799
51.333632301389784
51.033160967736286
50.73675952307832
50.44435809234838
50.15588877922631
49.871285616084535
49.59048451411621
```

49.313423213817934
49.040041235956416
48.770279833114195
48.504081941883165
48.241392135752726
47.98215657872325
47.726322979662385
47.47384054741233
47.224659946648615
46.9787332544862
46.73601391782479
46.496456711423164
46.26001769669098
46.026654181185876
45.79632467880373
45.56898887065056
45.34460756658478
45.12314266741979
44.90455712777768
44.68881491958568
44.47588099620831
44.265721257208725
44.05830251373388
43.853592454518946
43.651559612506894
43.45217333208019
43.25540373690154
43.06122169836189
42.869598804633455
42.680507330326485
42.493920206748655
42.30981099276615
42.128153846265846
41.948923496218136
41.77209521534047
41.59764479336131
41.42554851088519
41.25578311385899
41.088325788640645
40.9231541376708
40.76024615574913
40.59958020691659
40.441135001945604
40.28488957644024
40.13082326954888
39.9789157032921
39.82914676250878
39.68149657542381

```
39.53594549484092
39.39247407996446
39.2510630788543
39.11169341151776
38.974346153643204
38.8390025209795
38.705643854365775
38.57425160541604
38.444807322862474
time taken: 0.22548866271972656
14.51083847786982
Loss at iteration 0 14.51083847786982
137.73163356366368
133.34773970506453
130.3346331186276
128.38166201221077
126.93105361019836
125.7284081668116
124.72975418641957
123.83230166016867
122.97300988635021
122.13520565241583
121.31272489235306
120.50272820839459
119.7037375678585
118.9148943508587
118.13565045345342
117.36562342565685
116.60452359743672
115.85211542592073
115.10819629236408
114.3725845935907
113.64511293018337
112.92562413937483
112.21396893397635
111.51000445560052
110.81359335248294
110.12460316123389
109.44290586715047
108.76837757166155
108.10089822606594
107.44035140808448
106.78662412760178
106.13960665356966
105.49919235722655
104.86527756860092
104.23776144430411
103.61654584521939
```

103.001535223045
102.39263651486176
101.78975904502714
101.19281443378453
100.60171651203775
100.016381241785
99.43672664174389
98.86267271772962
98.29414139737585
97.73105646881291
97.17334352294203
96.62092989896671
96.07374463286389
95.5317184085001
94.99478351111671
94.46287378293074
93.9359245806164
93.41387273445278
92.89665650894197
92.38421556472109
91.87649092160949
91.3734249226502
90.87496119902198
90.38104463571436
89.89162133787397
89.40663859774514
88.92604486214167
88.44978970040023
87.97782377277734
87.51009879926406
87.04656752880196
86.58718370889402
86.13190205561183
85.68067822400913
85.233468778956
84.7902311664163
84.3509236851933
83.91550545917318
83.48393641009912
83.05617723090997
82.63218935967903
82.21193495418935
81.79537686718056
81.38247862230264
80.97320439080885
80.5675189690194
80.16538775658346
79.76677673556456

```
79.37165245037056
78.97998198854545
78.591732962436
78.20687349174146
77.82537218695045
77.44719813366363
77.07232087779671
76.70071041165295
76.33233716084996
75.96717197208089
75.60518610168623
75.24635120500754
74.89063932649127
74.53802289050765
74.18847469284574
time taken: 0.21161460876464844
12.598088290297563
Loss at iteration 0 12.598088290297563
164.26551540996982
159.2526420458399
155.7718491087778
153.57992446304064
152.1970965694884
151.17593742260766
150.2217162560347
149.29823930818233
148.39083585572277
147.49739002918864
146.61575701718422
145.74528850905222
144.885350436701
144.0355965771647
143.19571145344233
142.3654504241166
141.54458846007876
140.7329239943066
139.93026757257954
139.1364412160713
138.3512754359532
137.5746084312498
136.8062850007432
136.04615593938018
135.29407744418745
134.5499106530534
133.813521222207
133.08477895895467
132.36355748936654
131.64973396117287
```

130.9431887761825
130.24380535039785
129.55146989913658
128.86607124523198
128.18750064831477
127.5156516533751
126.85041995685899
126.1917032886582
125.5394013084372
124.89341551483818
124.25364916620279
123.62000721155181
122.99239623066529
122.37072438220908
121.75490135895355
121.14483834922758
120.54044800384371
119.94164440781678
119.34834305627852
118.7604608340639
118.17791599850965
117.60062816506222
117.02851829534258
116.46150868735496
115.89952296756064
115.34248608456576
114.79032430419096
114.24296520570634
113.70033767902645
113.16237192266644
112.62899944226511
112.10015304948446
111.57576686109724
111.05577629807688
110.54011808450831
110.0287302461431
109.52155210842916
109.01852429385517
108.51958871846051
108.02468858737616
107.53376838927649
107.0467738896401
106.56365212273613
106.08435138227223
105.6088212106602
105.1370123868756
104.66887691290715
104.20436799881054

```
103.7434400463986
103.28604863161644
102.83215048566369
102.3817034749391
101.93466657989272
101.49099987287966
101.05066449511526
100.61362263283687
100.17983749277869
99.7492732770683
99.32189515765236
98.89766925035636
98.47656258868139
98.05854309743545
97.64357956629296
97.23164162336944
96.82269970889344
96.41672504905038
96.01368963006713
95.61356617259922
95.21632810647604
time taken: 0.23977112770080566
11.546784674510658
Loss at iteration 0 11.546784674510658
66.01070369476062
64.61128308043831
63.94178930918882
63.38495362550321
62.85364865880673
62.33661339658394
61.83061602640976
61.334663234160196
60.84823537429717
60.370997698265086
59.90267080446651
59.44300286765948
58.9917572740384
58.54870903449753
58.11364287133266
57.68635218407776
57.266638265807316
56.85430967412306
56.44918169202625
56.051075861350746
55.65981957608604
55.275245727839724
54.897192396782074
54.52550258239563
```

54.16002396893964
53.80060872106672
53.44711330548453
53.09939833496911
52.7573284314174
52.42077210497503
52.089601646599526
51.76369303171761
51.44292583291295
51.12718313983669
50.81635148477035
50.51032077248873
50.20898421327042
49.91223825808667
49.619982535164795
49.33211978727205
49.048555809199826
48.769199385046804
48.49396222500427
48.22275890143831
47.95550678414146
47.69212597469382
47.43253923992886
47.17667194454594
46.924451982947716
46.675809710410384
46.43067787371584
46.1889915413908
45.95068803370778
45.71570685260819
45.483989611708964
45.25547996655264
45.030123545255485
44.807867879702016
44.58866233742512
44.372458054301795
44.15920786818329
43.94886625356765
43.74138925741087
43.536734436161446
43.334860794091846
43.135728722989214
42.939299943257076
42.74553744646961
42.55440543941107
42.365869289623404
42.17989547247778
41.99645151977776

```
41.81550596989563
41.63702831943721
41.46098897642516
41.287359214986616
41.11611113152627
40.94721760236357
40.7806522428093
40.61638936765454
40.454403953043844
40.29467159970223
40.137168497485646
39.98187139122316
39.82875754781949
39.677804724586295
39.52899113877096
39.382295438252406
39.237696673373755
39.09517426988293
38.954708002953225
38.81627797225667
38.679864578064745
38.545448498351604
38.41301066687682
38.282532252225394
38.1539946377844
38.0273794026367
37.902668303353266
time taken: 0.2046196460723877
{0: {'accuracy': 0.9750865051903114, 'precision': 0.5450400430715391, 'recall':
0.4829939020780297, 'f1_score': 0.492434481137619}, 1: {'accuracy':
0.9847750865051903, 'precision': 0.8444830002335592, 'recall':
0.7653162689903219, 'f1_score': 0.7416574615327075}, 2: {'accuracy':
0.9771626297577855, 'precision': 0.8131162450700394, 'recall':
0.7396497821914187, 'f1_score': 0.7299068646378135}, 3: {'accuracy':
0.9854671280276817, 'precision': 0.7999532128564386, 'recall':
0.7704828987945053, 'f1_score': 0.7846770461199254}, 4: {'accuracy':
0.9840830449826989, 'precision': 0.8569027181688126, 'recall':
0.7193151342800513, 'f1_score': 0.7337308686145896}}
```

2.0.1 6.1 Experiment

Since the performance on the test runs in model selection is good enough, I will experiment only gradient method and learning rate.

[166]:
```python
# generating all possible experiments
import itertools
method = ['batch','minibatch','sto']
lr = [0.01,0.001,0.0001]
```

```
experiments = pd.DataFrame(list(itertools.product(method,lr)),
                           columns=['GD_methods','Learning_Rates'])
experiments
```

[166]:

	GD_methods	Learning_Rates
0	batch	0.0100
1	batch	0.0010
2	batch	0.0001
3	minibatch	0.0100
4	minibatch	0.0010
5	minibatch	0.0001
6	sto	0.0100
7	sto	0.0010
8	sto	0.0001

[205]:

```
# running the experiments
model = MultinomialLogisticRegression(k=4,n=8,method='batch',penalty=0.
↪9,max_iter=100)
for i,(method,lr) in experiments.iterrows():
    model.method = method
    model.alpha = lr
    # print("Logistic Regression with L2 regularizqtion")
    exp_name = f"method-{method}-lr-{lr}"
    print(exp_name)
    mlflow.start_run(run_name=exp_name, nested=True)
    model.fit_CV(X_train.to_numpy(), Y_train)

    params = {"method": method, "lr": lr}
    mlflow.log_params(params=params)

    y_hat = model.predict(X_test.to_numpy())

    accuracy = model.accuracy(Y_true=y_test.to_numpy(),Y_pred=y_hat)
    p =  model.macro_metric(metric="precision",Y_true=y_test.
↪to_numpy(),Y_pred=y_hat,weighted=True)
    r =   model.macro_metric(metric="recall",Y_true=y_test.
↪to_numpy(),Y_pred=y_hat,weighted=True)
    f2  =  model.macro_metric(metric="f1",Y_true=y_test.
↪to_numpy(),Y_pred=y_hat,weighted=True)

    mlflow.log_metric(key="accuracy", value=accuracy, step=i)
    mlflow.log_metric(key="weighted_precision", value=p, step=i)
    mlflow.log_metric(key="weighted_recall", value=r, step=i)
    mlflow.log_metric(key="weighted_f1_score", value=f2, step=i)

    signature = mlflow.models.infer_signature(X_train.to_numpy(), model.
↪predict(X_train.to_numpy()))
```

```python
    mlflow.sklearn.log_model(model, artifact_path='model', signature=signature)

    mlflow.end_run()
```

```
method-batch-lr-0.01
10.107689350717147
Loss at iteration 0 10.107689350717147
13527.557807277653
12615.538320014006
11892.367675046635
11309.219686279655
10770.675248204214
10280.409573548966
9836.388955345254
9433.113809712937
9058.479053939735
8701.192779643698
8356.838863985507
8026.5748618695925
7709.22726854489
7405.075383231195
7112.522721760343
6833.511838522128
6568.549154830945
6325.679307745537
6080.62202302363
5853.041978581116
5626.885201390507
5416.4664683194915
5207.957844946908
5012.891546853536
4820.652287023875
4640.129352656445
4462.8046669100595
4295.515066837718
4132.0523891486055
3977.1002532378843
3826.3460275544644
3682.5824418908724
3543.7558222800003
3410.452088762822
3282.561102279912
3158.6871113931784
3041.1901677390947
2926.1253263394988
2818.0727390853635
2710.89340782634
```

2612.0039234447668
2512.1600623784393
2421.4336890408126
2328.2009899505088
2245.5775366780276
2158.4034503003786
2082.763830678931
2001.1874527550117
1932.678834516721
1856.0971642351626
1793.4717260222924
1721.7589777235398
1665.327296669034
1597.7262825542462
1546.311409392513
1482.9875477276914
1436.818327039927
1376.9375075159287
1335.2079315369106
1278.9457513032053
1241.5708954266402
1188.3106440825934
1154.922689205052
1104.5773622972908
1074.8251479463088
1027.1397919223277
1000.9131881332235
955.53058889901
932.3354607777596
889.3030116229504
869.3208901489968
827.9686617231508
810.0546818122473
771.1912240517265
757.6742445340855
718.4708969042781
700.9252798500767
669.9007570457364
676.5786564231776
626.4024291904458
598.4470995516119
583.0206618652389
597.0027228824514
570.3231254568476
566.6186184587441
535.1107316040016
514.4201268825489
512.4449258745252

```
480.8652856166469
474.98776144075407
450.6304488208289
455.16721045389806
421.0396720310852
412.13308679394066
396.7878480794661
458.93581712648995
415.4733636652874
440.9757458365966
402.8193475455206
time taken: 0.21941924095153809
10.693447595028648
Loss at iteration 0 10.693447595028648
8576.61282667072
7880.175701478209
7377.040945451093
7034.256032734079
6732.699443784227
6462.792827669751
6203.549203299993
5962.303627524736
5708.436229981745
5500.956427809921
5261.671521850855
5047.497984530023
4849.04246186358
4655.753565579819
4459.953276793917
4299.206789488985
4110.530473125022
3950.713312320668
3790.706803298583
3647.5502341079164
3489.650146600589
3363.1309552754574
3218.5540493841872
3103.2031772499163
2967.5421082489415
2860.5153161813323
2740.676968865772
2640.5479428037606
2526.116421108973
2434.1643980270537
2336.2334824954833
2247.165066168055
2152.2988603637755
2072.0017141464023
```

1990.4471248143486
1912.5941344093794
1834.4458570147042
1764.2001827367067
1696.6156606386105
1628.1056603037555
1565.7998689217425
1502.9697319741877
1446.3144415519782
1386.9775946930292
1339.297035751354
1280.9416977349908
1236.1543887280259
1183.4197764152748
1143.7369794521937
1092.825933777506
1062.069137496418
1010.6027078693342
979.8729415519009
935.0374614324459
908.8533274813485
864.2104268082419
849.7256923141437
801.0154196650515
779.107663809232
744.6040214870882
728.9943909697907
692.7414280387848
702.0457655380211
657.0922935350144
660.4495037702999
617.1745889028748
635.9575671577009
854.1574729194826
744.0979651143073
713.6899163395317
689.0314483167886
732.6870263843347
655.9854787688065
647.0293920254165
691.1077609767123
617.1065771424164
632.4937278531193
665.3996698503753
599.6218270033407
606.1851185934636
625.1778535905403
569.3091351467351

```
552.3421940911933
546.1316281847248
511.6128928745734
506.7827676688725
558.9352294960125
487.23206100410783
478.6120052097662
504.83948575498266
451.52028225324045
442.18357674495184
431.6703960732278
432.08027282970653
445.16938636738365
444.58481871214155
521.803262662453
442.5079190930819
426.5715207158016
time taken: 0.20878219604492188
16.080822208434398
Loss at iteration 0 16.080822208434398
14267.66982468298
13411.166337781598
12712.200329928837
12129.530074651415
11608.538928718233
11129.159293107492
10688.494784073479
10277.24917536489
9854.199117034255
9462.254145026016
9088.990772987354
8731.99449806038
8373.817928724473
8055.147802366612
7717.751459463903
7407.4360988121325
7115.723381176984
6833.320737161471
6551.127448673934
6303.987631157961
6034.922839614586
5790.129620934978
5561.3049939443135
5341.243122039245
5117.737654206576
4926.251493915615
4712.92216909407
4522.427651349388
```

4342.324031758161
4172.615991818666
3995.940448430873
3846.758924434152
3681.5797130945866
3540.652129503989
3392.8392346939117
3266.7204395157905
3125.4232282054327
3006.4891492509882
2880.7707842335094
2774.539030361342
2653.891730305556
2554.4372554467336
2446.9248583228823
2357.954689480218
2254.633318922417
2171.424485561485
2080.5711697363313
2005.1730200092616
1917.364840181631
1847.0898303291576
1771.2454162308156
1705.880679891489
1632.6139925402542
1572.067835823164
1509.6674621378322
1452.0654479516857
1391.9163396085426
1338.7152949449137
1287.8879867029566
1236.8627786404047
1187.97184112512
1140.7102642333462
1100.0950531328647
1054.4473517679526
1015.0127403726873
972.8341896361519
941.300125795916
899.9166331636776
867.9134771812578
831.3485988836386
807.0934610810581
769.0593592040279
744.1447050906888
712.2768479434766
692.2640904556371
658.5794602509079

```
643.8570077148155
611.1753448426905
592.9399100507433
566.6088399900159
560.8517158357668
525.3558444103379
508.5059400725552
490.6750652972926
492.39874480443996
474.86594366504374
490.3358346828228
483.23472292982154
643.2771891392337
553.8406103281055
526.2635195208526
528.8438954845807
609.8971608816463
531.6131660984618
505.55347580338304
482.1135962402592
468.8345508706342
459.4577955081797
447.0299537226354
time taken: 0.20360970497131348
14.215608414243299
Loss at iteration 0 14.215608414243299
16476.655262376866
15362.629128023467
14447.037861150324
13696.992103481816
13057.759451773745
12488.929245155774
11979.706862351939
11496.263011839326
11037.00059630457
10585.009086879916
10163.254060617073
9744.234542653956
9355.367717871723
8968.432225600167
8610.736543733035
8252.78665273233
7923.648295074459
7592.844589403258
7290.7995432840535
6984.4764038719295
6706.852717504874
6423.775109328328
```

6169.352194894228
5907.20295759715
5673.283004542081
5431.461391856219
5218.384366380688
4993.653340692392
4795.981257605315
4590.757366389425
4414.658848701739
4221.2087671158115
4052.242734293123
3880.8036622109753
3732.2005125176365
3569.2895237853704
3431.17971548975
3283.887965595267
3159.2835831215266
3022.830139455222
2905.7168727132444
2786.1475727607217
2675.3156026661204
2561.8103515285143
2463.9768392612127
2361.4549740205894
2265.245544516624
2180.363769737071
2087.486881455171
2001.3823068114998
1924.780059196441
1847.239759415265
1768.4298529514997
1708.0577481989062
1631.490314802694
1566.8628522814956
1506.5259757758554
1448.4935306061632
1383.8629430314788
1340.5996149723185
1278.7272080760763
1228.4216651773509
1182.8356287855531
1138.370123320654
1086.2462804589863
1054.8778147344674
1005.2429650063245
965.7102179761778
931.0373640111034
898.1723046025998

```
854.9106970096028
832.4919086363249
791.6678607447207
763.6278984047404
733.4835207841263
712.6355948199391
674.0927253198784
655.4687191311347
623.4526574787571
610.000739987904
574.9257906735043
561.6136120926969
532.9868908957919
523.661976958875
492.04462793978905
482.35168695176856
456.7469489460432
450.92587448565575
422.4185460370198
415.74931541900764
393.78848770822856
388.64595971032577
364.3265474000582
364.860712233183
338.5578868402639
332.0503057636578
324.9044522032422
316.8098128827581
316.59885714750936
time taken: 0.20841431617736816
12.129029971885837
Loss at iteration 0 12.129029971885837
7460.667475169481
6826.308419356316
6374.77398212286
6060.232552401779
5779.099294416102
5537.111656570907
5321.858678900053
5086.048222278195
4884.565408618422
4671.3328556377555
4495.220064822363
4293.912025617892
4122.8730130147915
3944.689208745654
3796.3349204562483
3628.9708405570946
```

3488.688153111802
3341.3197930741876
3211.268568153488
3067.606461858747
2951.5631185700727
2824.927069207144
2714.1577333395844
2599.442577376452
2495.344543294486
2391.9686246410024
2294.860165170372
2198.736819322641
2109.682110165183
2023.830717686677
1939.4964960922239
1861.3105282390165
1783.1360379394355
1712.251268663903
1639.1123486078795
1575.8323993110153
1507.0510719047506
1449.6598533177519
1385.6156357149514
1334.6965125175634
1274.0996515758861
1228.8605279150618
1171.956688264345
1131.5050627443454
1078.1219561050439
1042.9785321835416
992.0937128522644
961.2591001003453
913.4537823257388
886.3246778052043
841.1512074463598
818.6348248130812
775.0089805843357
755.391579578001
714.673762153033
698.2206862798965
658.978568508705
646.2026989113815
608.2368036310307
596.6244221951207
561.7376468772195
555.1735298756892
518.67578465609
507.6654832266156

```
479.4529044990818
493.7787234912114
446.11400927881823
429.65381649456447
423.24900785197246
396.86822335427627
418.3644381808977
375.41247724553017
375.24039000748354
367.8745766951728
372.47051394394424
477.9345897669432
408.72857543327723
386.89146511677774
376.06561180867806
389.6804854557975
361.91017398573393
389.48620699608136
353.3212625251254
372.80448736392765
614.1408325874377
513.1961078322767
483.82754102167576
457.1091424508222
480.8183122970802
430.1171995457717
437.41103586726854
461.219736909813
411.98631696952407
398.0683285047285
393.5639080291411
375.07076347937783
373.1946103467428
404.5752909478855
354.0491634848776
time taken: 0.20508170127868652
{0: {'accuracy': 0.8934256055363322, 'precision': 0.2886740331491713, 'recall':
0.47602004294917677, 'f1_score': 0.3043916609706083}, 1: {'accuracy':
0.9820069204152249, 'precision': 0.6081690140845071, 'recall':
0.6664284012389802, 'f1_score': 0.6349615188858214}, 2: {'accuracy':
0.9557093425605536, 'precision': 0.43033387308196464, 'recall':
0.34499672425088795, 'f1_score': 0.34751529838519163}, 3: {'accuracy':
0.9633217993079585, 'precision': 0.32110726643598614, 'recall':
0.3333333333333333, 'f1_score': 0.3271060979908354}, 4: {'accuracy':
0.9688581314878892, 'precision': 0.46396396396396394, 'recall':
0.6604628966833691, 'f1_score': 0.5179045775108966}}

2024/10/07 15:24:36 INFO mlflow.tracking._tracking_service.client:   View run
```

```
method-batch-lr-0.01 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/88170
3814181867940/runs/d8fb9c661adb487b8e20dd80ed0300ac.
2024/10/07 15:24:36 INFO mlflow.tracking._tracking_service.client:   View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.

method-batch-lr-0.001
10.623075076701932
Loss at iteration 0 10.623075076701932
132.70367041849815
128.37834762567883
125.58584017065255
124.01441310655117
122.99957352504536
122.10024859445515
121.24460881750134
120.41456670208728
119.6025199423912
118.80482404503566
118.01948188324633
117.2453207584766
116.4816012910819
115.72782529257569
114.98363233206375
114.24874253922853
113.5229239301521
112.80597352182974
112.09770631108631
111.39794882525702
110.70653534528276
110.02330569904302
109.348103978212
108.68077779832366
108.02117787771819
107.36915780296691
106.7245739025608
106.08728518264022
105.45715329743523
104.83404253825081
104.21781983143693
103.60835473970204
103.00551946346162
102.40918884031132
101.81924034155873
101.2355540652646
100.65801272556325
100.086501638232
99.5209087026062
```

98.96112438001717
98.407041668978
97.85855607737453
97.3155655919313
96.77797064522957
96.24567408055226
95.7185811148232
95.19659929989889
94.67963848245687
94.16761076271072
93.66043045216679
93.1580140306207
92.66028010257708
92.16714935325923
91.67854450436138
91.19439026968061
90.71461331075255
90.23914219260138
89.76790733970253
89.30084099224518
88.83787716277034
88.37895159325103
87.92400171267143
87.4729665951536
87.02578691867231
86.582404924392
86.1427643766528
85.70681052362778
85.27449005866707
84.84575108234176
84.42054306519503
83.99881681120539
83.58052442196363
83.16561926156332
82.75405592220164
82.34579019048653
81.94077901444481
81.53898047122458
81.14035373548437
80.7448590484623
80.35245768771588
79.96311193752565
79.57678505995335
79.19344126654701
78.81304569068493
78.43556436055025
78.06096417272873
77.68921286642204

```
77.32027899826889
76.95413191776754
76.59074174329129
76.2300793386911
75.87211629047783
75.51682488557694
75.16417808964907
74.8141495259694
74.46671345485827
74.12184475365677
73.77951889723886
73.43971193905345
time taken: 0.21556520462036133
10.471485108959092
Loss at iteration 0 10.471485108959092
122.44172975297778
117.97215500718444
115.08090702781645
113.42507622873244
112.41007917875962
111.52834171058393
110.69791080169196
109.89044486623416
109.09917086765617
108.32017766504704
107.5520048955903
106.79362521360889
106.04453031840013
105.30432514114165
104.57275035026686
103.84958397345527
103.13464405571864
102.42776317629834
101.72878793304199
101.03757211952814
100.35397616340857
99.6778652145027
99.00910881190084
98.34758028354692
97.69315654726663
97.04571786807423
96.40514772041105
95.77133264865698
95.14416215764732
94.52352860766713
93.9093271202096
93.3014554882748
92.69981409220756
```

92.10430581943342
91.51483598807918
90.93131227394008
90.35364464062685
89.7817452726359
89.21552851117771
88.65491079259155
88.09981058920272
87.55014835248487
87.00584645840237
86.46682915481412
85.93302251082825
85.40435436800163
84.88075429328516
84.36215353361867
83.84848497208486
83.33968308553571
82.83568390360803
82.33642496904994
81.84184529928244
81.35188534912554
80.86648697462078
80.3855933978874
79.90914917295099
79.4371001524899
78.96939345544615
78.50597743545256
78.0468016500311
77.59181683052083
77.1409748526978
76.69422870805192
76.25153247568952
75.8128412948329
75.3781113378921
74.94729978408483
74.52036479358546
74.0972654821843
73.67796189644139
73.26241498932163
72.85058659629783
72.44243941191256
72.03793696678851
71.63704360508012
71.23972446235945
70.84594544393
70.45567320356405
70.06887512265796
69.68551928980239

```
69.30557448076277
68.92901013886735
68.55579635579895
68.18590385278789
67.81930396220233
67.4559686095331
67.09587029576983
66.73898208016462
66.3852775633797
66.03473087101538
65.68731663751377
65.34300999043428
65.00178653509572
64.66362233958058
64.3284939200958
63.996378226684584
63.667252629283425
63.3410949041183
time taken: 0.22636771202087402
12.406213227717833
Loss at iteration 0 12.406213227717833
129.16803366198906
124.69817658530818
121.61022002326173
119.60659487002908
118.20334928336912
117.08434337058219
116.17904330651339
115.38290976171626
114.62323211485786
113.88207398762458
113.15361944067902
112.43533529932459
111.726012136345
111.02498010652957
110.33183057379114
109.6462851823496
108.96813531590928
108.29721088567752
107.63336414870132
106.97646099602918
106.32637621235376
105.68299083969968
105.04619067987883
104.4158654157285
103.79190807269197
103.17421466981
102.56268397822194
```

101.95721734243585
101.35771853977216
100.76409366429961
100.17625102751842
99.59410107128052
99.01755629020856
98.44653116185705
97.88094208340677
97.32070731400037
96.76574692201365
96.21598273667536
95.67133830352662
95.13173884326916
94.59711121359479
94.06738387362805
93.54248685064516
93.0223517087642
92.50691151932739
91.99610083272471
91.48985565143172
90.98811340405878
90.4908129202303
89.99789440613584
89.50929942061245
89.02497085163827
88.54485289313382
88.06889102198468
87.59703197521296
87.12922372723965
86.66541546719225
86.2055575762229
85.74960160481284
85.29750025004749
84.84920733285462
84.40467777520472
83.96386757727836
83.52673379461052
83.0932345152257
82.66332883678105
82.23697684373697
81.81413958457726
81.39477904910153
80.97885814581363
80.56634067943085
80.1571913285372
79.75137562340578
79.3488599240124
78.94961139826378

```
78.55359800046088
78.16078845001795
77.77115221045597
77.38465946868754
77.00128111460931
76.6209887210163
76.24375452385026
75.86955140279427
75.4983528622221
75.1301330125114
74.764866551727
74.40252874767951
74.04309542036367
73.68654292477862
73.33284813413194
72.98198842342761
72.63394165343729
72.28868615505313
71.94620071401968
71.60646455604126
71.26945733226063
70.9351591051043
70.60355033448849
70.27461186438015
time taken: 0.16817784309387207
11.263609896927058
Loss at iteration 0 11.263609896927058
135.6329449545535
131.3228477555967
128.50435801609646
126.88274745893283
125.81156481544701
124.84980990943822
123.92306086123304
123.01607748540677
122.12364309071623
121.24356073346063
120.3747023506325
119.51642446311804
118.66831618584567
117.8300917721467
117.00153480637228
116.18246930979129
115.37274342696945
114.57222019259142
113.78077211307098
112.99827796095315
112.2246208348458
```

111.45968696416253
110.70336495193294
109.95554527949632
109.2161199693558
108.48498234559581
107.76202685610707
107.04714893552213
106.34024489631588
105.64121184055689
104.94994758773652
104.2663506158303
103.59032001375068
102.9217554439354
102.26055711415215
101.60662575779098
100.95986262202062
100.32016946324156
99.68744854929508
99.06160266789949
98.4425351407874
97.83014984301737
97.22435122693214
96.62504435023425
96.03213490765167
95.4455292656695
94.86513449981081
94.29085843395951
93.72260968123143
93.16029768591551
92.60383276602741
92.05312615603846
91.50809004936846
90.96863764025692
90.43468316465639
89.90614193982051
89.38293040229134
88.86496614402193
88.35216794640155
87.84445581198368
87.34175099374814
86.84397602175954
86.35105472711419
85.86291226309658
85.37947512349399
84.90067115804295
84.42642958500603
83.95668100089932
83.49135738741117

```
83.03039211557154
82.57371994724697
82.12127703405115
81.67300091377341
81.22883050443735
80.7887060961115
80.35256934059963
79.92036323914493
79.49203212828436
79.06752166399322
78.64677880425981
78.22975179023024
77.81639012606189
77.40664455762153
77.0004670501614
76.59781076510176
76.19863003604446
75.80288034413704
75.41051829290095
75.02150158263197
74.6357889844746
74.25334031426672
73.87411640624313
73.49807908668193
73.1251911475703
72.75541632036021
72.38871924987929
72.02506546845486
71.66442137030506
71.30675418624412
time taken: 0.20532751083374023
10.716529249989893
Loss at iteration 0 10.716529249989893
118.3060817306528
113.9032238767291
111.06648131900882
109.41507283582698
108.2525442553096
107.36856833926394
106.65764691667128
105.99496463181002
105.34883531151073
104.71456594095393
104.08909297800555
103.47172607451321
102.86183410247938
102.25917375603387
101.6635022512045
```

101.07465062312903
100.49245288330695
99.91676080843867
99.34743079359887
98.78432656797456
98.2273166509693
97.67627470630484
97.13107893912053
96.59161201902394
96.05776083992664
95.5294163790621
95.00647353420695
94.48883098383446
93.97639104897956
93.46905956379477
92.96674575076491
92.46936210127005
91.97682426043593
91.48905091602053
91.00596369085227
90.52748703850386
90.05354814189033
89.58407681454025
89.1190054043245
88.65826869946487
88.20180383667753
87.7495502113351
87.30144938955686
86.85744502215871
86.41748276041344
85.98151017358771
85.54947666823595
85.12133340924201
84.69703324260875
84.27653062000233
83.85978152506456
83.44674340150982
83.03737508302694
82.63163672500855
82.22948973813175
81.83089672381514
81.43582141157732
81.04422859832212
80.65608408957588
80.27135464270047
79.89000791210586
79.51201239648407
79.1373373880855

```
78.76595292405625
78.39782973985334
78.03293922475214
77.67125337945764
77.31274477582774
76.95738651871315
76.60515220991437
76.25601591425169
75.9099521277391
75.56693574784701
75.22694204583382
74.88994664111884
74.55592547766345
74.22485480231997
73.8967111451008
73.5714713013138
73.24911231550182
72.9296114671186
72.61294625786559
72.29909440060874
71.98803380978812
71.67974259322804
71.37419904525132
71.07138164099679
70.77126903183685
70.47384004179001
70.17907366482139
69.88694906292507
69.59744556488218
69.31054266559072
69.02622002586577
68.74445747261183
68.46523499927368
68.18853276647661
67.91433110277299
67.6426105054175
time taken: 0.2046654224395752
{0: {'accuracy': 0.9743944636678201, 'precision': 0.5635062764360759, 'recall':
0.4785856238663947, 'f1_score': 0.484754468589055}, 1: {'accuracy':
0.9847750865051903, 'precision': 0.8444830002335592, 'recall':
0.7653162689903219, 'f1_score': 0.7416574615327075}, 2: {'accuracy':
0.9771626297577855, 'precision': 0.8131162450700394, 'recall':
0.7396497821914187, 'f1_score': 0.7299068646378135}, 3: {'accuracy':
0.9882352941176471, 'precision': 0.8913824632515667, 'recall':
0.7951126062984768, 'f1_score': 0.8335246293639407}, 4: {'accuracy':
0.9833910034602076, 'precision': 0.7868403007371488, 'recall':
0.7251261345177918, 'f1_score': 0.7316148479960648}}
```

```
2024/10/07 15:24:45 INFO mlflow.tracking._tracking_service.client:   View run
method-batch-lr-0.001 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/8817
03814181867940/runs/6e3e36411641450cb1549a3331525cad.
2024/10/07 15:24:45 INFO mlflow.tracking._tracking_service.client:   View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.
```

```
method-batch-lr-0.0001
14.767776180762816
Loss at iteration 0 14.767776180762816
13.89598602372362
14.239522237064907
14.496930731442239
14.70656703944576
14.886821370618742
15.047637431744793
15.194889786806174
15.332244758646468
15.46206992096139
15.585926811538366
15.704859090967739
15.819571152644166
15.93054373813533
16.038110974510726
16.142512413409797
16.243928125498652
16.342501995245716
16.438356744844715
16.5316032265267
16.622345841217097
16.710685430821233
16.7967205969826
16.880548097828406
16.962262751049572
17.041957112538103
17.11972109071905
17.195641584835546
17.26980219000918
17.342282984508355
17.41316039913288
17.482507160618425
17.55039229753758
17.616881196400048
17.68203569631248
17.745914211899663
17.808571875772472
17.870060693408103
17.930429704758158
```

17.989725148166354
18.047990623245546
18.105267250243312
18.16159382413471
18.21700696224332
18.271541244631454
18.325229346837947
18.378102164797653
18.430188931966114
18.481517328810146
18.532113584921646
18.582002574076487
18.631207902600906
18.67975199142985
18.72765615225024
18.774940658120293
18.82162480894742
18.86772699219328
18.913264739157544
18.95825477717284
19.00271307802315
19.046654902877503
19.09009484401043
19.133046863561002
19.17552432956304
19.217540049461324
19.25910630131142
19.300234862844913
19.34093703856699
19.381223685039583
19.42110523449056
19.460591716877847
19.49969278052645
19.53841771144667
19.57677545143268
19.614774615032292
19.652423505471386
19.689730129609394
19.726702211996056
19.763347208093776
19.79967231672493
19.83568449179829
19.871390453364754
19.9067966980484
19.9419095088952
19.976734964678585
20.011278948697743
20.04554715710218

```
20.07954510677301
20.11327814278959
20.146751445507743
20.17997003727388
20.21293878879768
20.24566242520412
20.278145531784485
20.310392559464212
20.342407830004536
20.374195540953323
20.405759770359953
20.437104481267472
20.46823352599493
time taken: 0.25436830520629883
10.68891056201144
Loss at iteration 0 10.68891056201144
10.285304932801484
10.611316898393794
10.864081983981743
11.072104196482272
11.25149818452962
11.41157613089124
11.558066144738401
11.694650588578805
11.823759391987199
11.947014294644784
12.065497228247418
12.179923869221419
12.290762201077204
12.39831635160529
12.502786449009603
12.604310712466363
12.70299395970637
12.798925815496808
12.892191399251898
12.982876820958534
13.07107133609808
13.156867531875285
13.240360491144836
13.321646539654212
13.400821933072553
13.47798167167304
13.553218524297046
13.626622281097891
13.698279221483386
13.768271768566084
13.836678296743255
13.903573059927584
```

13.969026211495274
14.033103891516886
14.095868361377196
14.157378170028815
14.217688339685294
14.276850561711392
14.334913395852382
14.391922467832279
14.447920661821888
14.502948305408026
14.55704334555189
14.61024151466533
14.662576486405351
14.714080021128202
14.76478210118527
14.8147110564077
14.863893680234177
14.912355337001317
14.96012006094957
15.007210647508689
15.053648737421538
15.099454894248879
15.14464867577428
15.1892486998
15.233272704794349
15.27673760581902
15.319659546133495
15.36205394484258
15.40393554092354
15.445318433941083
15.486216121732236
15.52664153531848
15.566607071279794
15.606124621804419
15.645205602608904
15.683860978905393
15.722101289577258
15.75993666970949
15.797376871607039
15.834431284422399
15.871108952502649
15.907418592556507
15.943368609732858
15.978967112694217
16.014221927761255
16.049140612197824
16.08373046670005
16.1179985471475

```
16.151951675669565
16.185596451075767
16.218939258694597
16.25198627966183
16.284743499696013
16.31721671739568
16.34941155209016
16.381333451273303
16.41298769764729
16.444379415801357
16.47551357854871
16.50639501294281
16.537028405992963
16.56741831009744
16.597569148211193
16.627485218763923
16.657170700343176
16.686629656156157
16.71586603828292
time taken: 0.21639776229858398
13.96450611117969
Loss at iteration 0 13.96450611117969
13.490751414269774
13.862324145448877
14.172408904856848
14.433477033828016
14.658828063296268
14.857635075232183
15.036150292597654
15.198739828765733
15.348536834116453
15.487846385055219
15.618402173666883
15.741533989751224
15.858280021900743
15.969463993834985
16.07574926294026
16.17767745399274
16.27569649130907
16.370181230498687
16.46144884245396
16.5497704239688
16.635379860992405
16.718480668479366
16.799251324007244
16.87784946875075
16.954415248464816
17.02907399538114
```

17.101938400407857
17.17311028768792
17.242682076278864
17.31073799359789
17.377355090327683
17.44260409528604
17.5065501403192
17.569253378862857
17.630769516904717
17.691150271297452
17.750443767434223
17.808694886003206
17.865945566733
17.92223507561094
17.97760024091782
18.032075662509357
18.085693898038073
18.13848562921183
18.1904798106973
18.24170380387677
18.292183497336147
18.341943415688664
18.391006818110487
18.439395787773357
18.487131313199026
18.534233362424338
18.58072095075092
18.626612202755272
18.671924409151476
18.716674079026745
18.760876987908187
18.80454822206595
18.847702219411214
18.890352807307817
18.932513237580718
18.974196218974342
19.015413947286675
19.056178133381614
19.09650002926128
19.136390452361592
19.175859808218455
19.214918111637182
19.253575006485438
19.29183978421834
19.32972140123438
19.367228495151753
19.404369400086612
19.441152161007548

```
19.477584547233924
19.513674065140147
19.54942797012238
19.584853277879628
19.619956775056703
19.65474502929289
19.689224398716394
19.723401040921505
19.75728092146264
19.790869821896703
19.824173347402645
19.857196934005213
19.889945855427662
19.922425229596364
19.954640024818705
19.98659506565412
20.01829503849643
20.049744496884802
20.080947866559132
20.11190945027459
20.142633432389268
20.173123883237682
20.203384763302243
20.23341992719378
20.263233127451834
time taken: 0.18457674980163574
12.574974613011682
Loss at iteration 0 12.574974613011682
11.644263665136828
12.026701737798419
12.311300362601795
12.544198262410697
12.744971873796295
12.923874987722288
13.086915123878963
13.237866694889844
13.37922094314603
13.512687369607258
13.63947939249158
13.760486001440421
13.87637889857519
13.98768122650638
14.094812540452113
14.198118673073054
14.297891798273037
14.394384048991391
14.487816857643434
14.578387442203324
```

14.66627338023061
14.751635897756634
14.834622290780096
14.915367757659274
14.993996827573644
15.070624508135312
15.145357233943011
15.218293670518824
15.289525410019575
15.359137583251938
15.427209404749423
15.493814662590358
15.559022161318717
15.622896124158082
15.685496559278537
15.746879593922332
15.807097779544668
15.866200370670558
15.924233579836013
15.981240810727732
16.03726287143109
16.092338169524734
16.146502890610577
16.199791161734154
16.252235201028295
16.30386545480093
16.354710723183715
16.404798275361944
16.454153955317118
16.502802278930726
16.550766523221878
16.598068808421335
16.644730173520156
16.690770645872448
16.7362093053779
16.781064343720995
16.82535311909907
16.8690922068313
16.91229744620374
16.954983983872644
16.997166314118086
17.038858316212917
17.080073289147425
17.120823983928034
17.161122633648343
17.200980981512654
17.240410306976038
17.27942145014997

```
17.318024834609506
17.35623048872582
17.394048065637097
17.43148686196087
17.468555835342123
17.50526362092333
17.541618546815297
17.57762864864109
17.61330168321942
17.648645141448124
17.68366626044383
17.71837203498904
17.752769228333953
17.786864382396576
17.82066382740127
17.85417369099277
17.88739990685991
17.920348222900586
17.953024208957373
17.985433264150714
18.017580623834867
18.04947136619992
18.081110418541364
18.1125025632174
18.14365244331265
18.174564568025495
18.20524331779539
18.23569294918508
18.265917599531836
18.29592129138079
18.325707936712593
time taken: 0.17571139335632324
11.10744055782661
Loss at iteration 0 11.10744055782661
10.209716329685412
10.538126745641796
10.790070835799472
10.995378589461597
11.1709526972371
11.32665095576631
11.46856547234286
11.600604489798982
11.725322695933636
11.844397623726698
11.9589250908916
12.069613896772234
12.176919086832738
12.28113425418141
```

12.382454667451801
12.481019086205203
12.576936206375196
12.670300428307872
12.761200562022479
12.849724107134834
12.935958921636734
13.019993460517105
13.101916314456263
13.18181547689364
13.259777575148249
13.33588718351601
13.410226267323097
13.482873768605392
13.553905324372467
13.623393099608153
13.691405714482741
13.758008245813516
13.823262284910594
13.887226036617092
13.949954447073186
14.011499350242437
14.071909625425173
14.13123135982103
14.189508011706907
14.246780571003422
14.303087714955327
14.358465957391662
14.412949790598596
14.466571819266125
14.519362886287759
14.571352190423475
14.622567396000623
14.673034734939856
14.72277910146672
14.77182413991327
14.820192326036329
14.867905042285091
14.914982647445772
14.961444541077885
15.007309223138684
15.052594349170791
15.097316781404961
15.141492636105852
15.18513732746507
15.22826560832235
15.270891607973414
15.313028867302132

```
15.35469037145446
15.395888580253434
15.436635456537275
15.476942492586813
15.516820734794186
15.556280806711309
15.595332930604604
15.633986947631444
15.672252336743698
15.710138232414728
15.747653441277718
15.784806457755884
15.821605478758137
15.858058417507632
15.89417291656498
15.929956360102786
15.965415885483566
16.00055839418877
16.035390562142965
16.06991884947358
16.104149509743536
16.138088598691148
16.17174198250906
16.20511534569153
16.238214198477294
16.271043883913006
16.303609584560693
16.33591632887073
16.367968997240453
16.399772327777022
16.4313309217819
16.462649248973072
16.49373165246011
16.524582353485954
16.555205455948766
16.58560495071578
16.61578471974086
time taken: 0.2521998882293701
{0: {'accuracy': 0.9647058823529412, 'precision': 0.2889564336372847, 'recall':
0.3019608344411494, 'f1_score': 0.2948580928630804}, 1: {'accuracy':
0.9688581314878892, 'precision': 0.3760123668380549, 'recall':
0.38595187038360734, 'f1_score': 0.3808068896512293}, 2: {'accuracy':
0.9570934256055363, 'precision': 0.38605001762896496, 'recall':
0.39694129321211147, 'f1_score': 0.3913694534146634}, 3: {'accuracy':
0.9695501730103806, 'precision': 0.47616728358984356, 'recall':
0.43806653583777216, 'f1_score': 0.45334418667752}, 4: {'accuracy':
0.972318339100346, 'precision': 0.44343447747110165, 'recall':
0.4590542953290713, 'f1_score': 0.45094218540341874}}
```

```
2024/10/07 15:24:55 INFO mlflow.tracking._tracking_service.client:   View run
method-batch-lr-0.0001 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881
703814181867940/runs/7e7d49a17332451babe0d8ac0fffcde5.
2024/10/07 15:24:55 INFO mlflow.tracking._tracking_service.client:   View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.
```

```
method-minibatch-lr-0.01
7.62521199863307
Loss at iteration 0 7.62521199863307
1050.3386772246029
999.539858840701
939.9487410972669
894.4817790351482
859.1592120130367
824.8320398008385
792.9860598695358
760.3058826762355
732.1203430901793
702.8291890265314
675.6416873356136
649.6187501938443
624.3826245670374
598.7493434245403
575.9433556948413
552.8911303635401
531.326917056322
509.68772788099324
490.31054855701285
472.12285407857416
453.30421141807574
435.16447337927065
418.89878732860114
402.0045024898747
387.0653326013766
372.8065567052962
357.8752409145698
343.5436834018218
330.6410784871775
317.87199239957926
305.8284641997177
294.13765655627884
283.77563702722324
273.6206555904218
263.01306134310016
252.9501232472239
243.75359295659516
233.96673290129843
```

225.37310678819824
216.87631727883183
208.72448340418134
200.50970945354396
192.55170507415497
185.69769039205926
178.86820337623752
171.93400696527743
165.55817098811772
159.8925200050509
154.53130834688295
147.7968425642123
142.46564290234917
137.19272178502558
132.11245194950965
126.7926510782281
122.5413832871565
118.3994742576012
113.74487689040281
110.18744708061634
107.26815661312523
102.51983998021288
98.60929971148997
95.50083946503506
92.41131048199462
89.68099382749882
86.03030624511672
83.13403649685704
81.875959983289
77.67349427963718
75.0060285345573
72.04980803849715
70.29363503430115
67.42266636984708
64.95920148092985
62.52342383875996
60.07378193543753
58.06147642706794
58.15219399387688
59.108786380310704
57.30553631611217
58.76953439375862
55.75522097889021
52.70980013296953
51.36276238383781
49.87750715452856
47.98943998920773
46.07254848481423

```
44.53897416519935
43.43581438444428
42.09456274516442
43.37475741043199
39.85437411965231
39.213019158837085
38.96021428148862
37.26723240167886
35.55306919747759
34.06164253358831
33.474077242416065
33.15272633488601
36.136985510016814
time taken: 0.04772806167602539
12.165040900768101
Loss at iteration 0 12.165040900768101
1339.8242843378175
1247.15748510211
1173.526112135193
1115.8340166939904
1072.9010807581865
1027.454768933285
985.3003552319807
946.2702361085707
907.9452355543958
872.5928188223673
837.6211534041146
806.8973232059301
777.4023097488134
746.334290297725
717.7682068258779
688.3973364199253
661.4924199648884
634.527913611519
609.0257028682219
586.1544838359347
560.248409634074
539.4937997130216
516.4685102922627
495.9754617765172
476.72304701555174
457.1509429609981
440.00756655063594
422.445388296378
404.4505959086056
387.6320867211671
374.1893075901628
358.1058696933017
```

343.6339138034441
330.74662162438176
318.7344257418243
306.03934973702343
293.2519890731074
283.03418708715776
271.07685400000366
261.7888678870421
250.79628838140943
241.08215693526657
230.29679630745494
220.0947255240955
210.53901281116313
202.94022466441893
195.37830512342845
187.16309938725556
178.60151944245644
171.93415592098214
165.48225324360223
159.14245087680322
153.05229655078233
147.23030709890398
141.42953180346507
136.18414365755146
130.0156691880677
125.89005854374749
120.84040197384374
115.60853942128564
112.13343920110661
109.11002189682473
103.48226672418954
99.37128013793168
95.69153607291776
92.419215194541
89.4509621461513
85.78960230524544
83.77282844191555
83.00867565140487
77.93617459318894
74.83155878147599
72.2198411645537
69.96510838457073
66.14008343717765
64.29129285087525
61.43293434684811
58.514064737323935
56.73827896345023
54.56871629448605

```
53.41012626305728
51.26743715911547
49.52041996556282
47.55993982253485
45.8041012140202
44.36614765874993
42.731511594659636
41.40893006918378
41.668960519821546
38.91731204220141
38.35268380579569
36.60495684381759
35.306690189493416
34.966708269542785
34.76005084277308
34.451070056045225
39.39487049655182
35.765829538949326
36.87460071340461
time taken: 0.05017852783203125
9.281407078958562
Loss at iteration 0 9.281407078958562
486.3320091045373
465.6610153402912
434.259937714455
416.7328419777852
398.81979056286224
382.12015564336724
367.88807622813806
353.00279724114614
339.76694091101416
326.54279034241785
313.9127061071797
301.981932604078
290.6386141377977
278.65599749849616
267.790862645511
256.72817629271
246.24745615093332
237.3968117605601
227.90301106555634
219.33781356101161
211.22888431399915
203.09616540416266
196.4438325768443
188.48957553284387
182.12379939301533
174.6240286283925
```

168.72543066150251
161.8976984570932
155.24588589674923
149.84904728538854
144.7941841921036
138.70700114739103
133.45865459854372
128.72508460394434
124.50124542671503
121.17574507448983
118.23859094610002
113.49677143322589
110.89378326481997
107.14140660500009
102.83681501643656
98.98702297502449
94.97570841443505
91.61269356829236
88.08206389842165
85.9094377156551
81.25515813056475
78.06738636915692
75.30133118081248
72.66080957803128
69.92564285172419
68.07121862003778
65.73277125735844
63.300932502841285
60.84157333748035
58.95413043389007
56.7339014677448
54.585421187733594
52.565691520858266
50.699567153297345
49.180320479991344
47.32009847655915
45.83824616040376
45.070148797384384
43.48976535767805
42.430498868007206
42.61889774749056
45.78421977488612
42.11272856277529
41.79475032961407
39.67113162779357
38.2313059190049
36.9916781827624
36.07954313717203

```
35.631947483159934
34.106060131098296
34.68682105168292
32.73167574678429
31.55973010580302
31.240240508933013
30.047790216258523
29.554913122498114
32.516868810885164
29.643355927360144
29.525106931092218
33.32130433135564
31.23039192866082
34.94312516653279
36.38660040314416
51.418214542710345
47.3854078874258
41.712633140883405
38.19536132271038
37.306628282398684
37.25135748184555
35.72291398386753
34.38660079093157
34.26881036803747
34.52279346264051
time taken: 0.05583834648132324
11.583144640485935
Loss at iteration 0 11.583144640485935
544.2009875325465
498.67954991696723
474.9321324550441
455.1886720887294
436.93136735456454
418.28381772217006
402.73537020564027
386.3955739838772
370.44939909875006
355.4498293537422
342.94078841660115
327.8880992714427
315.17258546712924
302.2055242413907
290.7657953918931
279.55123781053965
268.7118788923675
258.4224702748261
248.10691343478726
238.57217731190377
```

229.3361242815608
220.91241883722665
211.3339527254197
203.267019142051
195.54650708296654
188.33787824490702
180.87456124215575
174.0242955847973
167.78440425441383
161.85228863105834
156.112319630736
149.41420379511575
143.74638355451452
138.15342208164594
132.8890120199557
127.79433177142951
122.73755611400644
118.00672528414299
113.55259511494339
109.04795541243583
105.32115301181383
101.59059495789779
97.3619291209979
93.9970881162838
90.24069526735094
86.90258056290739
84.39353396466333
83.01923540615522
78.97787585241603
75.32609307846744
72.77847959365188
70.13700756162694
68.00509228778814
65.52506678386989
64.1233741225486
61.586392955419065
62.695552422935236
58.791044235141015
56.74264745913485
55.359029206616086
52.91437112300367
51.53922944639842
52.18857156289784
49.053954191200596
47.44416971281422
46.03000362467627
44.98370710119497
44.65123127501766

```
43.73114001100206
41.56261211394179
40.0293566889458
38.77793919280612
37.96849904398382
36.42975244895089
35.66239333753052
34.27899221610106
33.80421259691773
33.790591422383194
38.89636392433379
35.54312970255626
38.432634567501594
35.32141648021502
34.4734225992492
34.0676511887831
33.778632683851676
33.33690900226494
34.48834022740302
33.25760032618668
32.054164442053164
31.26946292217145
30.227558777645296
29.04278422487521
28.314078584875716
27.939693169869496
27.84065868146142
29.267245464475987
27.633918291757155
27.52388352613637
28.083892284568613
time taken: 0.05831003189086914
12.409919693010963
Loss at iteration 0 12.409919693010963
1107.6958019571175
1020.3392070622122
960.9094850267244
917.2885702230623
873.0895716025118
836.5436095287185
802.5698290699199
769.6029341455384
737.7756341056358
708.3657164024577
677.6811255405486
650.4370808792621
624.3621295302044
601.9200903697408
```

577.8390777722062
553.811889788308
532.8582634536066
510.65606920846733
489.61765428312816
470.2893237505792
450.6454115910189
432.9030623947818
415.0855451258746
397.74838018004306
380.5991611364282
366.2443715530028
351.05711194195413
335.7106492661366
322.051185431994
309.2137890364307
296.4771428212612
283.2204131515825
271.26675752098566
259.67186950194537
248.21593541275513
238.65500673273831
229.23243833772855
220.68098329285226
210.34840420612272
202.3601966435277
193.8414269224581
185.52829005116695
178.01642099685876
171.03886447824277
164.62544935256454
158.3372618209423
151.83987168613308
145.4537315967777
139.6797886886498
132.8571310826992
126.89742550030503
122.46078584719727
117.66726246571818
113.21282102161213
109.77344769850197
104.05381668510604
100.49855107533259
97.01757146582923
92.71083847244448
88.74343111940985
84.70203098270498
81.30841732952871

```
77.05340232333928
73.61260609887373
70.5843261488059
68.25905817934246
66.03138859665457
63.91872928241509
61.614115745941795
60.44230028288022
57.91918047786425
56.160568464952796
55.56895946554543
54.59442258741097
51.79510282938947
51.054266937836005
48.55772054439939
47.031276784799005
45.15022359588981
44.21180206909978
41.71233925239501
39.63311775565936
38.697802538119845
37.860582285350674
36.19006561611143
34.87371397941412
34.026176411575165
34.59746348797931
32.874838242978946
31.328402735493565
30.216594406670506
29.124266534828422
29.01410158722246
28.708503972685154
28.986311786798296
28.069257894922156
30.008157148139023
29.525356803705737
30.377438125620145
time taken: 0.06389689445495605
{0: {'accuracy': 0.9508650519031142, 'precision': 0.4045977011494253, 'recall':
0.5099831271091113, 'f1_score': 0.3659256718124642}, 1: {'accuracy':
0.9480968858131488, 'precision': 0.5269186712485682, 'recall':
0.6563400161112812, 'f1_score': 0.48569888533086925}, 2: {'accuracy':
0.9557093425605536, 'precision': 0.3185697808535179, 'recall':
0.3333333333333333, 'f1_score': 0.32578438310922386}, 3: {'accuracy':
0.9757785467128027, 'precision': 0.7547787788161413, 'recall':
0.7672881039155218, 'f1_score': 0.7339096625627284}, 4: {'accuracy':
0.9640138408304498, 'precision': 0.6271106941838649, 'recall':
0.9027533289076367, 'f1_score': 0.7171772152088227}}
```

```
2024/10/07 15:25:05 INFO mlflow.tracking._tracking_service.client:  View run
method-minibatch-lr-0.01 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/8
81703814181867940/runs/5ea8707fd5834b62affa925e76f58526.
2024/10/07 15:25:05 INFO mlflow.tracking._tracking_service.client:  View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.

method-minibatch-lr-0.001
11.769958940082654
Loss at iteration 0 11.769958940082654
19.808030589525046
19.425117610678434
19.252746834117776
19.020145509300683
18.92014900406945
18.881746124261266
18.859808807746173
18.863289690477835
18.874262244659185
18.898607250850823
18.926946694131587
18.86999994566204
18.909546935333886
18.956518331774884
18.98285123472208
19.020422036921644
19.027294271148666
19.050726504889123
19.07269045318127
19.311215490489023
19.025913070584803
18.996225538265133
19.016370836259167
19.05478986536983
19.082441871615323
19.120058914766798
19.149623296250383
19.168712080089804
19.19354297012385
19.20925062478386
19.23995563401931
19.260540001728792
19.271642795513976
19.28534593197048
19.31017049221978
19.334130498624674
19.350648423326827
19.393577207010313
```

19.42013653956333
19.432316666784054
19.43754872298666
19.47822895375767
19.473641647395578
19.472827788194447
19.48567302862158
19.46406324063101
19.466156433270005
19.49444593885405
19.496887511285586
19.52240182420274
19.530532576471806
19.54183382259123
19.561520444062065
19.50395790501195
19.51242989673387
19.50362975241927
19.516198209832467
19.514708025625975
19.527404442188807
19.53432682219445
19.53296614847394
19.56418494360116
19.575340734827225
19.585482573917716
19.591426972465065
19.59775260536561
19.62778808360589
19.628194426801592
19.580806483346194
19.60071718912007
19.590628010263927
19.592482218457764
19.605666093899963
19.621399417858356
19.632510104675774
19.567473335767332
19.56873682880274
19.573094711462243
19.565845252873803
19.5452622565947
19.55206609830244
19.55485793530587
19.56737082090215
19.57700206139652
19.563686818304895
19.554153212577187

```
19.574345776864657
19.629210953433635
19.51965482277004
19.536656444432037
19.536681852584728
19.55282350593811
19.54591797091018
19.553235198971556
19.544996257375775
19.516422658255596
19.523264918681893
19.517639553941464
19.526385034386553
time taken: 0.05400419235229492
10.68154772231648
Loss at iteration 0 10.68154772231648
15.799843228529717
15.714729674046776
15.625143574997384
15.615006560730407
15.668310206082053
15.697248953247266
15.79755107685448
15.909746081249212
16.012679084319483
16.10135429831907
16.172578521528322
16.293700993350225
16.208569066875317
16.290002458754355
16.271726010986992
16.35619764257543
16.433190557726842
16.494145229758576
16.551639520854465
16.536833620269253
16.598250689062617
16.652476399051544
16.707884310117755
16.74357738930211
16.81881442832639
16.862431825016568
16.899835745814727
16.92184559518099
16.934177489567407
16.968900139078386
17.006638413077464
17.063717404898103
```

17.099518061040314
17.13755315126911
17.18685758691761
17.189523960511487
17.214454047801713
17.215582157441315
17.253595344391368
17.283637749221338
17.31326046323622
17.375347344481252
17.349213797618205
17.350283517583215
17.360488465812203
17.373477742406546
17.378373570984675
17.395811291157735
17.41713270794936
17.456804011215635
17.40894695919003
17.43496327060093
17.446938937838745
17.440178813273732
17.535485518489704
17.409162412177384
17.420341410038706
17.400934135338005
17.412873426550117
17.461980693380127
17.478649778841216
17.505147022206973
17.529894436003232
17.49072073041734
17.463383729715375
17.489044430435715
17.495574269130916
17.49740034090545
17.520315163979138
17.545670888043723
17.56612573165586
17.58823774366751
17.637353451517342
17.654968134554597
17.672395269817134
17.692297150785684
17.709795067731655
17.668074495613784
17.706670501422344
17.731085062319135

```
17.745120726921805
17.784273591811953
17.70861254473967
17.728655733222322
17.736998169488874
17.762700480211702
17.773746445844193
17.754673435369554
17.77427958596792
17.788585735725455
17.754623193742926
17.763656675210047
17.777767691101722
17.80023942025585
17.808538902696185
17.8131446372852
17.82590054889002
17.82904849042317
17.787257976459234
time taken: 0.04575490951538086
9.02388728228292
Loss at iteration 0 9.02388728228292
8.707645869983915
13.577256780781003
13.417058570231166
13.38222802470386
13.37945728628602
13.426807797793503
13.480397303002079
13.6614016544514
13.527584964882147
13.632256057770224
13.752076880611284
13.719041896844484
13.79174812918357
13.760212830079311
13.861523142986234
13.942266105301773
13.979234255082922
14.126140011733362
14.020211398768808
14.00140929357342
14.040370327856412
14.119296702912505
14.108259478305932
14.19921573545176
14.296245040940972
14.342829830514917
```

14.429068453485913
14.503985154044381
14.55567049215268
14.666898604446816
14.586976271921843
14.714956963841537
14.613209858741104
14.645277330493144
14.711306656246624
14.743637276500495
14.775850014027128
14.847494737894419
14.9070826392175
14.959595313826465
15.022459308295698
15.094648262207265
15.10610278718235
15.114227670816714
15.107174539566273
15.172170426069524
15.194074226570422
15.210508513107138
15.268477479772274
15.226193583784452
15.28534805367544
15.334282269726696
15.41106694420506
15.448717432951343
15.47419364922537
15.463561875180133
15.531452436167207
15.573324822240085
15.623670385095386
15.694272766427712
15.726882240298819
15.767987847062663
15.79502999476742
15.844380743255925
15.877058893291048
15.89602348963206
15.945846612120567
15.984193968518763
16.02348565578063
16.069760026479276
16.09255742091715
16.14152846627761
16.167829744122965
16.183186835438352

```
16.214079393318286
16.248600908408573
16.296372708746837
16.347391472599497
16.36641129590083
16.385165250476213
16.39020280036566
16.370217441899474
16.385144121335596
16.409636653226546
16.426073029196797
16.46370769740146
16.48373240557718
16.511125984207627
16.520424130690948
16.469618691793674
16.49193261144312
16.4418981906103
16.459064616959242
16.470776124765756
16.483372649095955
16.471504020238953
16.52496233668609
16.544861237365513
16.49210265422225
time taken: 0.04320955276489258
14.613683853474825
Loss at iteration 0 14.613683853474825
23.033641325507606
22.729656505181183
22.53313761651083
22.449671494959343
22.37877527918829
22.35038037282184
22.34225771618722
22.318755421100082
22.302407916622034
22.324296536083448
22.346533200290224
22.34225742955224
22.35424024729563
22.311064848053068
22.337037872574147
22.219336653234453
22.229918271024086
22.227238837332745
22.22285738190651
22.198895996604854
```

22.205446671979455
22.223583291148238
22.244584578405796
22.236111952939538
22.320308110325367
22.16753752094152
22.182547422019752
22.18004836144543
22.19834714355342
22.19606217500542
22.162475268800033
22.173600863683284
22.16850258734993
22.16511750490546
22.109233234307435
22.103508515523416
22.113219405846827
22.115719735056242
22.108541799426213
22.101591541081554
22.119099979844794
22.10695721835862
22.10412790081885
22.041320727134764
22.02973577620739
22.043962166013305
22.03693191564317
22.035177264922364
21.998286933796436
21.971771014365228
21.9617769559082
21.964969196361203
21.978067115286322
21.960371701898495
21.970738201307697
21.94030012876294
21.926491764109034
21.934540009140065
21.930385540939383
21.94577011746053
21.88506325910049
21.88504110599421
21.883034961799098
21.87439445741295
21.896893516705973
21.844763654565742
21.828516806298488
21.843118385455444

```
21.84035001411392
21.83095810410969
21.818463787231305
21.82042873920125
21.797214115659933
21.805260795360034
21.810425189431587
21.806519251702667
21.815441936742687
21.778380469585027
21.76935252507532
21.762153583607496
21.761155824732835
21.768904766205438
21.772210289575966
21.779376983473163
21.76768223043564
21.800010706188747
21.741510082651594
21.720549170048884
21.73261060745614
21.712112689465208
21.687466223867276
21.664356421503754
21.666508300615774
21.660085838595958
21.654380179653657
21.643246237128544
21.62657289887002
21.5957315325895
21.599625284690685
time taken: 0.05399322509765625
10.86820180840178
Loss at iteration 0 10.86820180840178
18.262690879667442
18.107070191148832
18.020916749078356
17.97261522602174
17.91712822961119
17.941381014151652
17.894067643530413
17.947605699140613
17.992501038062716
17.96751190220548
17.937450082138916
18.017777807434463
18.0580405369219
18.02893628641759
```

18.0745841113404
18.121227795926654
18.157626373818804
18.208134300528556
18.236996075308387
18.28091764790395
18.261024728210124
18.298461611185772
18.358654901504288
18.376617872770534
18.344356632656915
18.375970308673153
18.42864949234253
18.463375745243827
18.495823174562606
18.52978559444619
18.539528635353946
18.55560844336182
18.57876863686619
18.60823597614333
18.62987150175291
18.65763096034776
18.677055916472323
18.691054003075447
18.739373471077577
18.748267205982668
18.767223407865934
18.78844177009934
18.810008373084308
18.760747879318703
18.718536798151973
18.767319967163317
18.7599587193227
18.796376504571036
18.78970706973843
18.711156383326518
18.723118936375315
18.69802285312046
18.709144393652647
18.658821645702414
18.61431830473974
18.635782904003857
18.655795817561323
18.663295976811913
18.702953144171033
18.67068649742865
18.69223830437486
18.702773558182166

```
18.721261762401678
18.696027060526752
18.744059816077105
18.708398121142633
18.72259635436576
18.74745646290069
18.76817455059546
18.81387374005663
18.83466849225704
18.837650460875444
18.85440745579218
18.8126444721285
18.82862105287821
18.84527924718003
18.8528102708429
18.819758477495657
18.839485374637928
18.87097407024385
18.88704492422843
18.908274943321597
18.872056231301613
18.907394152384096
18.905037970841207
18.9430691289417
18.934935897755448
18.954860116246795
18.982133761072323
18.971873607623266
18.934481407399414
18.9433156533894
18.911258663647658
18.888893811369133
18.932903960048737
18.95719488390429
18.977511753068438
18.983484818519532
18.994065431271135
time taken: 0.07482028007507324
{0: {'accuracy': 0.9764705882352941, 'precision': 0.5702524947706961, 'recall':
0.47037287205128064, 'f1_score': 0.4866745477538902}, 1: {'accuracy':
0.9806228373702423, 'precision': 0.8145982835438782, 'recall':
0.6303963148280517, 'f1_score': 0.6650035637918745}, 2: {'accuracy':
0.9653979238754326, 'precision': 0.45321217918061535, 'recall':
0.5080524043232226, 'f1_score': 0.4761644403023713}, 3: {'accuracy':
0.970242214532872, 'precision': 0.5761921189461677, 'recall':
0.6426735819082329, 'f1_score': 0.5294467440581786}, 4: {'accuracy':
0.9840830449826989, 'precision': 0.8559498244537614, 'recall':
0.7132655275044918, 'f1_score': 0.717492413320294}}
```

```
2024/10/07 15:25:15 INFO mlflow.tracking._tracking_service.client:   View run
method-minibatch-lr-0.001 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/
881703814181867940/runs/598e9e8e80004aff828a6f043dfbcda5.
2024/10/07 15:25:15 INFO mlflow.tracking._tracking_service.client:   View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.

method-minibatch-lr-0.0001
11.548862060204614
Loss at iteration 0 11.548862060204614
10.563820522152474
10.072408309355684
9.927000667008944
9.942923887195622
10.001047681442486
10.080882736926442
10.172455917356059
10.232877315829635
10.342928795997315
10.464946441886
10.414846398882336
10.517378403378466
10.603595122331232
10.653460552670376
10.723608997964298
10.783212408108168
10.853247482634709
10.925129425167581
10.991057083557568
11.056246761057421
11.111587016921131
11.143050092743836
11.197854025197596
11.25876351611662
11.316534794694078
11.510939390837882
11.359198144973371
11.393359205221333
11.460246917869203
11.499455219484462
11.553101517971186
11.578342048215085
11.586201631818728
11.614404201805925
11.660694086345053
11.70902415088022
11.736640080815297
11.775573901601119
```

11.826576610898154
11.85543439157847
11.898684198533351
11.931107428488623
11.951770407303368
12.001029634255199
12.255566748701305
12.033687616792724
12.045343738237523
12.088297513547213
12.124059921539414
12.148495884066005
12.181402684438558
12.21597404767874
12.231862843226727
12.264237106366819
12.288589282325656
12.320326640502197
12.348374269636137
12.379910167989356
12.411199315518939
12.430119471801087
12.46860524467034
12.49578084516702
12.507269516266984
12.538663491126355
12.559662870152923
12.56543771380503
12.590182711315073
12.60580027462679
12.609460391790643
12.636859878292162
12.669551368324063
12.683120140932127
12.69402643763887
12.714140606275498
12.743319966067444
12.750517289341728
12.790012471541953
12.782403938577314
12.799393948962482
12.808313896075656
12.820324813067463
12.847026382070071
12.87024713474863
12.890088605192805
12.914895545570566
12.944955258298629

```
12.963489957616508
12.975419852823737
13.012662891608693
13.02956982291562
13.05508115743224
13.066859671217259
13.062137830803122
13.09547958523167
13.11847798294822
13.137251233020738
13.156489454541946
13.15755338760203
13.184025008060685
time taken: 0.06895184516906738
12.26856502958927
Loss at iteration 0 12.26856502958927
11.26700644238639
11.107561479513137
10.981961295711713
10.751030253350155
10.71552573987844
10.717957456974469
10.744508196117055
10.732044768242643
10.793046745098495
10.882441919629784
10.948131198675865
10.97310634392875
11.046488913079486
11.135890582848692
11.225324179496475
11.310677702898765
11.370129113206866
11.869856369537718
11.422236730191363
11.484615129970814
11.520329166849521
11.557725010372785
11.60177331421213
11.649999159108914
11.712378408934676
11.78166353284591
11.811758621686819
11.842639205118353
11.878551135891307
11.923368698379491
11.979209191987158
12.01261522975759
```

12.046420885498234
12.089361590678811
12.125712825320097
12.171174555412964
12.227909733257134
12.271474686680964
12.292091661870234
12.345881820099587
12.386025470127716
12.410564755965448
12.465904946846416
12.482658282822243
12.501656310769299
12.536751413621007
12.574497930820336
12.616051870689576
12.62621764082097
12.650690759809073
12.674662504493169
12.706693982039726
12.736320916581166
12.773777013883732
12.760582902997573
12.798106805594669
12.824545630967116
12.849145635752787
12.896361159768743
12.91803158500244
12.936133719031853
12.96040561910899
12.991948812658555
13.004801925513757
13.00142352371684
13.039901751916164
13.061951148509918
13.077508970728095
13.103364118462046
13.126941832078828
13.162054389029203
13.1813181191163
13.197225728202312
13.226610920653398
13.250185368928667
13.279335603314609
13.312663820103404
13.325529695870387
13.345486361050966
13.379006155564213

```
13.400800960647913
13.418571737051733
13.450783247081336
13.467355846495474
13.497425333278466
13.527182090777742
13.538155147564451
13.528079496696114
13.565302181529171
13.580737262959355
13.59680128408277
13.614019064571458
13.629094493364107
13.649401140000556
13.660650982618355
13.686107538159337
13.705606118114117
13.72642132615423
13.74525248760666
time taken: 0.04325723648071289
10.490560463169087
Loss at iteration 0 10.490560463169087
10.39683337800479
9.7558218260426
9.525966005505497
9.461205809556658
9.536562748457095
9.607359090319003
9.708687731190093
9.82004390920171
9.902126409588769
9.928953304461686
9.990738575290738
10.022651383870917
10.111453491567149
10.17012513107033
10.206706837249685
10.268578511895889
10.354384112220137
10.431714259167364
10.484575673605546
10.54289214453868
10.587953822837363
10.647970005652288
10.715166362984684
10.711516816957442
10.774257066773872
10.821868305265786
```

10.871850041204343
10.877497232435253
10.923920843323058
10.957285399676413
10.98750135407615
11.01112518478253
11.04280610749111
11.084765710544152
11.130844864010424
11.181740090863043
11.23494330318302
11.270473031550758
11.315628209860343
11.347333643438967
11.382133979331144
11.426462013102306
11.473686338224335
11.505748550333566
11.54211564544188
11.572386182448115
11.610764101893635
11.637994533440231
11.672442847344712
11.716770771029546
11.738634885239028
11.781610287171395
11.773667864171392
11.805500158746543
11.826110373888659
11.862923962966951
11.888148011712184
11.91335737479096
11.955146820092189
11.969055565101435
12.012713603961798
12.040229876318554
12.068169580428775
12.098665756581862
12.130785778359165
12.154686971420118
12.183287934555207
12.196698486556592
12.216946489029088
12.240242150595048
12.49401562107981
12.264020144424412
12.291166298705688
12.324051901232256

```
12.335836692740138
12.356176010353343
12.369689370674061
12.396065658419282
12.40125822637629
12.431798142229955
12.457084215418897
12.48667746580858
12.510318028804065
12.540750245445299
12.551389435550872
12.580135770984008
12.61064213981662
12.636783558579387
12.656946738829076
12.657204020392093
12.677063373442024
12.703066105131063
12.733869471390491
12.750335578258793
12.746705468247143
12.77732694866013
12.79928740957143
13.089822495470878
12.820487669117984
time taken: 0.055852651596069336
10.88479362593655
Loss at iteration 0 10.88479362593655
10.781258410933537
10.626635653267197
10.565287636167055
10.619245151619847
10.662709005703816
10.725517805894473
10.777462962213257
10.861941954107762
10.904342151055465
10.9634780842922
11.048433039878226
11.130432165435707
11.168006850855908
11.223366791345802
11.301273838043745
11.320559468018864
11.352807630742621
11.379375888193225
11.415027034793946
11.516780494257999
```

11.512820330641928
11.555799756848119
11.593488018196897
11.626382663764932
11.676688098058097
11.71527056637436
11.70997242724136
11.748693254460816
11.75841745035624
11.801395164732671
11.857794113423665
11.846594096173657
11.87633091868335
11.92581741585267
11.97481634776811
11.948710373317157
11.98068813417476
12.012055332021626
12.053348210165536
12.076146101672979
12.09756691609396
12.099458865350101
12.164503643709718
12.139483102812624
12.152242648628434
12.194637872191642
12.194481226681802
12.246667470186216
12.253950620890222
12.276220962410717
12.294712161850642
12.317297931768277
12.313503129034965
12.343157389828976
12.346037206717174
12.35824635614593
12.411219644966737
12.390613545589172
12.425262547253423
12.433049990960482
12.479808293714973
12.490865231064728
12.52267954841395
12.610104778034536
12.542199434124374
12.536227878595085
12.559631946394465
12.585278807421586

```
12.599634541334078
12.617219173708422
12.643256763871358
12.642216865761133
12.683633633057429
12.680152647885418
12.675040088146758
12.709340893551268
12.693848981220317
12.704724754817972
12.709783727418827
12.738903679287873
12.775144590507221
12.783718359569127
12.763269959990415
12.79125012721934
12.804604547164235
12.814085770037954
12.84719022325974
12.842528840018515
12.86450070053018
12.89768829053584
12.90343030602276
12.94014465449446
12.933928186347286
12.95664134486225
12.972792217267669
12.986197027566408
13.004317781871093
13.045229546633816
13.034573542379487
time taken: 0.05995965003967285
13.445860386407814
Loss at iteration 0 13.445860386407814
12.497236839454825
12.218593498592778
11.931683761861414
11.883351265971983
11.908104272537342
11.965555140434951
12.060415017554272
12.148233770916361
12.245781631133871
12.339061011612412
12.431768531301614
12.535341005025021
12.620256957670632
12.692395752094075
```

12.777694176377237
12.843728507092552
12.915406912056417
12.99185878852889
13.06570442321554
13.121662556475567
13.135871471349923
13.194457112745441
13.2336576331331
13.269100413410092
13.322183616146923
13.383329584017519
13.42610479986201
13.453590235231903
13.472847662595257
13.532332520688472
13.517227962443323
13.571372454349937
13.608686098381437
13.67548004408542
13.718653159631982
13.762937736292448
13.803364800738184
13.852752608417374
13.879404867306029
13.909881951859276
13.94333017717328
13.959834219682435
14.001523205799021
14.043529768055201
14.084526045078102
14.130297278526596
14.153439564390501
14.185591309531754
14.19384439345532
14.232626358222491
14.231514816895398
14.270239670410158
14.320375856331514
14.349459320911777
14.351974819154249
14.40271665673793
14.423409548501134
14.427871633412192
14.452061870997442
14.482574713460291
14.514211185780693
14.554941542325201

```
14.58359649807619
14.61894585830798
14.632164249224765
14.646856125162262
14.669902048049133
14.69803109133489
14.711761295965777
14.74683624795233
14.770896956413173
14.798964468536193
14.840458396717846
14.868396587739472
14.887219795301586
14.918303011334837
14.935775368665507
14.954973997111788
14.992522727733757
14.997089467134234
15.027255801534386
15.037703580061013
15.049676346387523
15.076257699333347
15.114790239970779
15.126150383440223
15.141391977787618
15.155274446513577
15.173590456642827
15.19792489314719
15.218932265723007
15.234418555251189
15.244916261276378
15.26799021958376
15.285210511204749
15.30368640567769
15.330198078612204
15.322600285740732
15.343274054930095
time taken: 0.043910980224609375
{0: {'accuracy': 0.9660899653979239, 'precision': 0.29370188943317005, 'recall':
0.2760698946722569, 'f1_score': 0.2813451505126813}, 1: {'accuracy':
0.9681660899653979, 'precision': 0.3580720092915215, 'recall':
0.3464284012389802, 'f1_score': 0.34791141580132406}, 2: {'accuracy':
0.9557093425605536, 'precision': 0.4448700092850511, 'recall':
0.3603869981494891, 'f1_score': 0.37156979808737}, 3: {'accuracy':
0.9640138408304498, 'precision': 0.41878806029749427, 'recall':
0.3651352677319877, 'f1_score': 0.3765046855376292}, 4: {'accuracy':
0.9716262975778547, 'precision': 0.4811965811965812, 'recall':
0.41379310344827586, 'f1_score': 0.4355028888102818}}
```

```
2024/10/07 15:25:23 INFO mlflow.tracking._tracking_service.client:  View run
method-minibatch-lr-0.0001 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments
/881703814181867940/runs/523edec44f6c447198f9679d272a6f36.
2024/10/07 15:25:23 INFO mlflow.tracking._tracking_service.client:  View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.

method-sto-lr-0.01
11.643409406239329
Loss at iteration 0 11.643409406239329
11.493693729798483
10.836698157379436
10.585663094454874
10.381967877047137
9.565709851503373
9.562777128300056
8.866636751926578
8.804883052808115
8.62964517342996
8.19166984341966
8.193317200369535
7.646895959354306
7.55716027340608
7.459249746580833
7.023988949614566
6.885199244037991
6.387790154992853
6.394035322867251
6.294800145964139
6.010281061964591
5.691730080466902
5.418733616135212
5.473440989432836
6.404877302477929
5.041627947405956
5.684492558485715
4.685710177498185
4.722589007488969
4.519365245779757
4.402490816734364
4.001865930436825
4.101105741355345
4.11142891710422
3.914542979384683
3.7697731776451118
3.522445428320482
3.3469449386149908
3.6344044938147406
```

3.420425920316111
3.2737989607433584
2.8395934132701597
2.998089704851785
2.9386674002545017
2.79125522378074
3.119213393603551
2.922732320159204
2.5895682372051616
2.6907485934430295
2.5914171935160155
2.569333906656805
2.2651924541677193
2.2492454817680767
2.442950826610953
2.384673846276158
1.9934990843621228
2.248878026631556
2.16882688770551
2.111847549377736
1.8556391564019483
1.870716578608447
1.9504793031578576
1.9215702078274806
1.675793279185226
1.5945629037233342
2.161713508370294
1.681782200212555
1.8298534757936573
1.7740256797169458
1.7731964248906047
1.5213413284991957
1.736513404808573
1.5930544157541595
1.2808820142238517
1.608472052278488
1.9322243756728743
1.5575146198490768
1.1272897919461626
1.3355342631274234
1.4866546866282662
1.4584703339973633
1.551175984547959
1.2823929804345897
1.8785676471637776
2.1772748005526172
1.6148388883077265
1.4596843768867063

```
1.2998387266480846
1.0106696083897142
1.245403236120357
1.380229253818288
1.6991615619753955
1.195339155812158
1.0900109072309192
1.343180734393691
1.0808114543495122
1.302330516661549
1.2071743433928082
1.278777679796475
0.9989121921971487
time taken: 0.009606599807739258
10.711709954087299
Loss at iteration 0 10.711709954087299
9.291517552256998
8.877039745374898
8.955482453458925
8.305701528591502
8.427063768875477
7.989200917485826
9.514278880516658
7.155245668420465
6.897803828616156
6.492421772243638
7.49519555867246
6.646959227691551
6.34696103873687
5.844567780249246
5.993046707404337
5.623349520606493
5.287031996209556
5.444023375928448
5.210586421724417
6.494945774019261
5.4552565153071235
4.317832013311757
4.624239610418342
4.357162213764793
4.151371118792149
4.191299841285706
4.346279488073058
4.021596154016105
3.950926254279187
3.789562004371553
3.406441892751182
3.573317724521131
```

3.4802314430273382
3.5144634546267506
3.3575469895119014
2.926999098073199
2.9158611761063185
3.0535586681450995
2.6209461678633708
2.8986675135695403
2.7788995184073166
2.243797207135358
2.1761113905336233
2.7681962241080367
2.548560415737955
2.2380746257568207
2.390660983533555
2.1470775704610916
2.4893638726359995
2.102281695565178
1.982640939530463
3.2900066079102945
2.1084051778958464
1.8481635101864267
1.8449329263325203
1.9815707900680404
1.7776147476593638
2.9431197536446403
1.728671660698263
1.7526349370098528
1.8231411047687596
1.7675922851145183
3.139365616992761
2.674727659031708
2.5517815796936336
1.4542983209775753
1.6613842271686021
1.4106138891353328
2.0061707275612335
1.681373294329366
2.3288291683783804
1.7115062467594409
1.470346290956663
1.4621028787140329
1.3874104346177827
1.5797802320965373
1.3969521900178783
1.6598203949711143
1.1854804501068503
1.1705130599894364

```
1.338524635522345
1.1260189687202504
1.1237603199308108
0.8282745014816777
1.2704973750704966
1.5251993884028074
1.208505022486146
1.2108696278310962
1.1955271232792617
1.235348258420484
1.0256115899444684
0.9747172427670103
1.1522641907101692
0.9862952155582323
1.2699261246715103
0.9772256019615658
0.8102442715650224
1.0831941311710875
1.1437429790636715
time taken: 0.00920867919921875
11.458481579368398
Loss at iteration 0 11.458481579368398
12.246465968844136
10.548753734222814
10.623469078863597
9.919523809122143
9.59572622573075
9.321497083634092
8.933619157107715
8.46938855532449
8.291066688627577
8.09666886768529
7.811959139009875
7.926478957755193
7.5447403875395525
8.093894383970758
6.679021942520461
6.542617038633518
6.576480633525609
6.971412808008838
6.412793743127377
6.477248924618287
5.634449968739389
6.082660280587431
5.363512641319234
5.632423036369648
5.128407779969738
4.859180180653855
```

4.488903514354035
4.366862227335586
4.365814228650242
4.219510469027353
3.8607291483653183
3.6791265662011527
3.631641165817004
4.0852453213507
3.6075723577800662
3.4278366566354856
3.457689999183158
3.400958513896299
5.016465964868217
3.110765663680586
3.0266213576951904
3.1802166390940876
2.715569619441649
2.650461916695435
2.9312607844473515
3.438576686797295
2.3041382834552913
2.7261008938410534
2.5445472597914804
2.765396384525447
2.632612557115478
2.5914726108308677
2.252113652973632
1.9982256138021453
2.138411686852882
1.933478141309467
1.8616546369090246
2.618257296856531
2.2193137946588806
1.8420920654350976
2.1837600758946927
1.6674902352454315
1.6897125478329724
1.5006323199970744
1.5318272019623969
1.5161007345047433
1.7086139168173688
1.6260158381179755
1.268663150215128
1.4314319669793472
1.7717352377474422
1.7791541304511769
1.5412111845463177
1.4110371299463653

```
1.563907126513346
1.2906670085984882
1.367086849419315
1.373631522765475
1.6248213684297417
1.3878169604631867
1.8480470679321024
1.2656455608462256
1.1531385068617328
2.145563432332399
1.2184931515605213
1.6109477999001203
1.8549609880672295
1.0306500980863575
1.2590843587556688
1.1224065072345124
0.922957667887945
1.2784184533196121
0.9910090480869814
0.9172704546119824
1.9864101139224712
1.026493006552315
0.947208893947361
1.8740803387056197
0.9074695532496069
time taken: 0.00873708724975586
11.394682620519799
Loss at iteration 0 11.394682620519799
11.713493538931653
10.33589780984537
9.933957981812759
9.600785620123673
9.281647508269108
9.066109501750427
8.681737317878625
8.361566697486829
8.483146549815833
7.7103281284285545
7.418821578215088
7.191433780667543
7.049328283604394
6.875892393762247
6.868491047822158
6.629665891099156
6.052365707777508
5.981355816495532
5.368544940176514
5.517838241653555
```

6.048998579346246
5.538324725735124
4.929165146652014
5.037892605756067
4.75416917195159
4.61191028524655
4.271686005361588
4.271169774332893
4.460402981317793
4.07395619876088
4.613750889208006
3.8288511590203065
4.258357559321559
3.614606870253716
3.4274219475620047
3.3742266369392633
3.1222496671126745
3.2875310705089373
3.8526775622346587
3.054463836707789
3.4449641748378417
2.953568463992914
2.8381980556615884
2.7045126171577802
2.439592664150047
2.6135209280467504
2.550654271576804
2.489039449906408
2.463013503393478
2.182238254595238
2.1141093759481255
2.620680584424145
1.834047116076048
1.9016442001415528
2.1637980483882724
1.8361113558168685
2.0416898644384904
1.628240808154339
2.0096471953302415
2.0533027552031626
2.1069860358674513
1.8830553746896928
1.7483062586838105
1.772123629852175
1.9281002798428477
2.1897009113205126
1.7297106113753327
1.6349841628911808

```
1.6087998884420904
1.8353693874984072
1.8599923621644847
1.792378669610228
1.4965642355653972
1.6903272999438275
1.3103815682962168
1.480411411939627
1.1427606781846409
1.261302821818285
1.2605823927994617
1.1979706917811406
2.4958171982813004
1.3602943250271742
1.4533930422034613
1.4853391300123677
1.0931947623017104
1.0855838961593967
0.9750355040182356
1.1181225736563207
1.2633088673694461
1.2238357193060159
1.3826750714979092
1.1426337292134638
1.3214141926993803
1.2368510295120054
1.0016592458182005
1.0089489001715408
1.1049628690663384
0.9883563506933699
1.2382331594762173
time taken: 0.009077310562133789
15.465762403974924
Loss at iteration 0 15.465762403974924
14.247774794602016
13.861041757175162
13.335639511531081
12.553113815277552
12.400517145558876
12.115236384486153
11.15299183247222
11.361652578142937
10.907385117175588
10.605895907467724
10.120587624126802
9.513955848345216
9.266028346177961
8.922181935930949
```

8.621498656902995
8.231116102634566
7.679546040695431
7.529599301009096
7.335923988970072
8.543577421541226
6.9137059042872195
6.433383860611482
6.463488637944129
6.220339180398966
6.061761454630969
5.67510200080152
5.453368807128006
5.384860848319994
5.202263922688129
4.971258944928123
4.7943991852633765
4.834820857773504
4.482311326628183
4.497922918975047
4.17977986094668
4.208700799617114
4.158113600237103
3.959292096206096
3.6635273491614164
3.3459412460445166
3.4830544713149885
3.5328317143055328
3.2211748717555864
3.361394330616716
3.0958318821254567
2.9539124711593723
3.0606899308510505
2.851293647784102
2.61351534728405
2.5873022812046846
2.6539257793807405
2.695973105733533
2.6773356202521184
2.6329342641556934
2.387780865742382
2.611201982977776
2.0691796612484636
2.1401630147459647
2.269719947987804
2.453036238745627
2.1633483252494643
1.9402198501861974

```
2.152356921512408
1.7688884867392336
2.003653585179838
2.043673789622853
1.9148105556923773
1.7971607721828673
2.0724686562957233
1.804525286993461
1.8060726841015229
1.8086905092444223
1.777627570215408
1.7114262919315664
1.7133025947110818
1.6622641025758131
1.3307605392109463
2.713078414819564
1.4255388948337209
1.6269198168004593
1.3564252763448927
1.2024489127181064
1.2652829934976189
1.3140968901478949
1.3995530403531389
1.3771881588906563
2.386317544972829
1.3428998117987359
1.4041971547961298
1.442355595444473
1.2837704321826693
1.349794267667375
1.5019984140383982
1.2253470900523822
1.288539019797345
1.3327891685753
1.1965807714958108
1.3255135666361757
1.2098334472833967
time taken: 0.010089635848999023
{0: {'accuracy': 0.9667820069204153, 'precision': 0.2418628808864266, 'recall':
0.25, 'f1_score': 0.24586413234776489}, 1: {'accuracy': 0.9681660899653979,
'precision': 0.32272202998846594, 'recall': 0.3333333333333333, 'f1_score':
0.32794186591654945}, 2: {'accuracy': 0.9570934256055363, 'precision':
0.38605001762896496, 'recall': 0.39694129321211147, 'f1_score':
0.3913694534146634}, 3: {'accuracy': 0.9647058823529412, 'precision':
0.4888811674774149, 'recall': 0.35748411363423976, 'f1_score':
0.37011706762264796}, 4: {'accuracy': 0.9667820069204153, 'precision':
0.3222606689734718, 'recall': 0.3333333333333333, 'f1_score':
0.32770349519117997}}
```

```
2024/10/07 15:25:32 INFO mlflow.tracking._tracking_service.client:   View run
method-sto-lr-0.01 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/8817038
14181867940/runs/91719d5924ea4bbb9763d4050f693811.
2024/10/07 15:25:32 INFO mlflow.tracking._tracking_service.client:   View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.
```

```
method-sto-lr-0.001
10.511857155191246
Loss at iteration 0 10.511857155191246
11.956700093862388
11.284710912994896
11.022220766813973
11.535163294809276
11.318242412782254
11.230619647280607
11.506188615333148
10.986678431720177
11.250191347644947
11.571359462443725
11.093817697494158
11.298009425447402
10.647151991326222
10.811298550868008
10.8110419338926
10.266833264491053
10.981262029945622
11.234163783799373
11.203653839613175
10.44483857333852
10.996904687629703
10.99843608129826
10.37856660516917
10.537226717631382
10.013667305733009
10.934290560093615
10.712632670129604
10.663295690745024
9.617305149277751
10.78313470335415
10.043909238998758
9.915398345097314
10.02385332660926
9.580984524750718
10.755076143686328
10.146480703896055
10.35484274878562
10.293033874009653
```

9.84952386770964
9.679941729566105
10.092733064591254
10.166723279301838
10.12468253277363
9.790999680081194
9.744812084199525
10.168299914344441
9.155732622395515
11.302420201667374
9.529140593251702
10.400248732884133
9.44642603224733
9.122543069415764
9.900027472613399
9.581412921090243
9.482070731057787
9.526318142044373
9.240888832873697
9.120818958701722
9.557733165732259
9.487457735868302
9.269848430703197
8.941184342841634
9.453039265050277
9.308068076468725
9.693880588505507
9.324646002805846
10.172045771157467
8.766251333597996
8.725065032106441
8.892426551576307
9.204430391469701
9.170687345232558
8.942365177831535
10.10700017469063
8.201504887357702
8.648563165059334
8.877292639331573
8.931017958568436
8.472705524441395
8.515696030363197
9.86638508547173
8.439265931944044
8.72068497694106
8.55044285447484
8.755621437125129
8.38297300191057

```
8.369108219810387
9.332302612679538
8.485909856349224
8.604811953915535
8.056943455709861
8.022216736393348
8.13978353660865
8.251551835336851
8.086476601084183
8.433854814846246
7.701973800363161
7.914082554685931
7.853516533884215
time taken: 0.021735191345214844
10.444929633049847
Loss at iteration 0 10.444929633049847
11.068410777816965
10.367960940001534
10.049197610364951
9.591386217639519
10.80694746026783
10.727067621887393
10.328994584183134
10.296070662098238
10.883511458134665
11.757902460663235
10.810456306001225
10.887508048402701
10.649614100652936
10.044465688737379
10.383920182821939
9.934193422354225
10.26351330173634
10.071758528171207
9.830721120012704
9.922074180065767
10.002563507422307
10.195739524054131
9.900113777630121
10.183285701174452
9.70357600846608
8.837431867926233
9.752153670104764
9.58777848998358
9.325837220784733
9.786177854091001
9.624190104700924
9.845998872556347
```

9.606322982261139
9.275900739646797
9.63300504665599
9.598642350510762
9.911072741057733
9.633064516712869
9.243601224379551
9.209055402388913
9.364272274954832
9.153683587239145
8.914727323503266
9.067478119545646
10.586303385000331
9.27469094645858
8.912007416166514
9.194513693691013
9.030040143183953
8.86698220085729
8.697228727265722
8.703209267029253
9.22797039722175
8.639110250848894
8.83578156305068
9.192504401926936
8.907897130601373
8.823526998972657
8.843539497312847
8.82446166221012
8.678988704945164
8.749082144839482
8.349924769742657
8.702216790871088
8.444975707552873
8.305960754211142
8.279554820591128
8.349722545085026
8.350282754744155
8.773591694745615
7.854650712332248
7.819503960826281
8.403442569809561
7.8325468724187655
8.446780484174852
8.3563310998509
8.664049735717812
8.570921476219453
8.460728983495395
7.6615861721133784

```
7.728827406917852
8.38901743463321
7.789722521946739
7.776768908383815
7.996151150541803
10.117443667325052
8.038672626163146
7.5925113586733755
7.8059619352309335
7.96791528336701
7.684538850116781
7.733137724436575
7.889302382105344
7.577075739229944
7.959826701824789
7.703476570254554
7.208079684755295
7.259817596328132
7.8118262246578745
time taken: 0.043148040771484375
10.868154897513874
Loss at iteration 0 10.868154897513874
10.587244637672015
9.067160480237227
10.007408242607989
10.011698602228451
10.398200265884979
9.900871872853601
10.317874687157797
10.287743432146623
9.679981563567765
9.796985817318072
9.310068641749012
10.12750730063671
10.029536325600464
10.061128426669697
9.960292295404397
9.77046493294404
9.926019227036061
9.960967485112414
9.547095454139463
9.36355977001466
10.060221492915332
9.936684613963362
9.500425707009573
9.39486615417798
9.570786821541722
9.611473242735823
```

9.562447226230844
9.294005437702204
9.227442219598418
9.998829181089008
9.130636924565607
8.530664229268035
9.403060630894547
9.160616248399243
9.307536763330017
9.274896184737452
9.361853436316347
9.017868429916732
9.072088755511981
8.603903880327035
9.227665758646925
8.949650804141928
8.81699646227865
8.268502574491315
8.72818220920172
9.203915457102697
8.73998926377715
8.643549396102474
9.101330242377053
8.081288569568272
8.68031913338532
8.84302552727895
8.21123727788275
8.242418558863843
8.914043847646358
8.884717341473896
7.663050105338949
8.32959960194014
8.376033608132
8.525009503861932
8.647970607172525
8.248959920725937
8.433892922033536
8.294589610409606
8.763452525724094
8.058735013638163
8.536089918578579
7.914689519301177
7.920730359882123
8.04834442725074
7.857649859642023
7.7527844550342415
8.024643565025276
7.166736204380924

```
8.056220546224695
8.093790187466656
8.203175381505247
7.595786359621652
7.771996480062773
7.7639494870912715
7.921667311233049
7.035861577210975
7.864914046158008
7.668722560014374
7.814887447452225
7.5273734089580415
7.539318993232172
7.408686008861936
7.465744782705281
6.489668635156278
7.624406052125838
7.46878828257021
7.457815600928932
7.409915533725218
7.344137068467151
7.544343891658583
7.474415401678164
7.293204426847596
7.332786621221857
time taken: 0.033721208572387695
10.682562751833595
Loss at iteration 0 10.682562751833595
10.727236769045694
10.997181792491428
10.758193258036552
10.56871163786144
10.535254331057958
10.55342465054018
10.714213896832248
10.410129478167086
10.708555536977808
10.648744760599508
10.395953341405182
10.616483305637361
10.24352116849107
10.862234179777797
10.984341152565648
10.4494103811494
10.455363033546833
10.059299383524568
10.13601981997945
10.709324251139625
```

9.993179110154088
9.950927192130049
10.102563513046146
10.144448778805572
10.065914396001624
9.678979529430304
10.416980070798441
9.705661920060017
9.952148638982138
10.200053197544944
9.448895852824098
9.44035424034843
9.492542840147092
10.410065955373494
9.628939047560353
9.785475842187456
9.466807207267793
9.60491425182139
9.306362564272051
9.297647011444253
9.728202661759235
9.769573419644512
9.21494314959918
9.369913807094127
9.328630274097698
9.334492686286309
9.65045902228156
8.911712249773322
9.011298304558704
9.217274262450259
9.857789767230052
9.257907123966552
8.916678418949363
9.026397978594483
9.160153894186035
8.753902116822102
9.078715669664053
8.92053335371007
9.368947902219015
8.670811993384039
8.837764441634732
9.046402884782006
8.63031706863798
8.784268199249441
8.328292582204968
9.050698326121728
8.369371078409166
8.670867825990543

```
8.581322897742119
8.549963360662185
8.378883706717918
8.301764370471217
8.602084115592495
8.302351974298281
8.482973206633705
8.608962634309801
8.371067275079689
8.199199959749281
8.32024062668388
8.173718217746922
8.107116691434715
8.210157469340018
8.048116016084533
7.94442076684877
7.919877839880353
8.230960531238901
8.27080331418143
7.794694719860601
8.020148505140076
8.004334326679421
7.627377505485001
8.00377520222726
7.997572849338071
8.137945123923465
7.655362232163277
7.692511973634559
7.735988515069217
7.984713783989667
8.24420357247419
time taken: 0.02156376838684082
7.866984663630525
Loss at iteration 0 7.866984663630525
7.46749539346258
8.9688092233446
7.5417151160068965
7.701028628272072
7.5111661654227415
7.71972711303697
7.891422163801797
7.375504524720026
7.296791972181883
7.269850666772254
7.458913791661125
7.4388610446038514
7.286197525510886
7.451124548444829
```

7.206238058083846
7.1789987937222595
7.177774904988468
7.180313512491207
7.211754024360131
7.017414234907639
7.418137004189988
6.9925060830963455
7.082348530190329
7.323282184158481
6.920235269872007
6.983134522441923
6.843345141726134
7.0959072927092475
6.794091199489959
6.8512446939843095
6.861056842622042
6.733899609389074
6.685933248809061
7.111397228189969
6.770411981608701
6.6719016241837625
6.800905183993118
6.801797528728238
6.677630927116432
7.1839840476240076
6.569355329255957
6.4445534152294055
6.493188035813336
6.564049972151611
6.4630042508373275
6.462306474998599
6.440138635096721
6.527874623672684
6.607044467065251
6.568499204317198
6.287833540687653
6.417409258561438
6.244569378943684
6.3935477637249525
6.363371249102631
6.3737603590694825
6.330898138245847
6.193283142809666
6.205928369307881
6.210811439521317
6.141875380187073
6.17703119276569

```
6.1155636099515736
6.097024568527127
6.115155033604517
6.1416214463419925
6.069799880280052
6.128972397697848
6.031460721929815
6.014461043619757
5.936113191179572
6.019604804981838
5.8495830539375975
7.137724808837651
6.227376296121391
5.93329317195271
5.823720779168722
6.396310892268051
7.055602744231566
5.830388002816448
6.132773664556748
5.7969936260768975
5.714604245782082
5.686301353772146
5.6715472992957965
5.646627925224762
5.5881762176822996
5.616255799899178
5.79999785065214
5.550073277263142
5.553800412095637
5.997948620002396
5.616807403966325
5.599136795405793
5.537636210082862
5.848170972998009
5.483771640307473
5.441152274878372
5.40952731404144
time taken: 0.01995992660522461
{0: {'accuracy': 0.21591695501730104, 'precision': 0.22033898305084745,
'recall': 0.05583392984967788, 'f1_score': 0.0890919474585951}, 1: {'accuracy':
0.3017301038062284, 'precision': 0.3333333333333333, 'recall':
0.10388372647128902, 'f1_score': 0.15840145322434152}, 2: {'accuracy':
0.07889273356401384, 'precision': 0.21348314606741572, 'recall':
0.027516292541636497, 'f1_score': 0.04874919820397691}, 3: {'accuracy':
0.10934256055363321, 'precision': 0.5799908762325859, 'recall':
0.38099827118960844, 'f1_score': 0.2312878780782525}, 4: {'accuracy':
0.8581314878892734, 'precision': 0.34704415954415957, 'recall':
0.46892463800876566, 'f1_score': 0.34351974111262046}}
```

```
2024/10/07 15:25:41 INFO mlflow.tracking._tracking_service.client:   View run
method-sto-lr-0.001 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703
814181867940/runs/f48c3d6e8ad34f778fa1b198175c38bb.
2024/10/07 15:25:41 INFO mlflow.tracking._tracking_service.client:   View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.
```

```
method-sto-lr-0.0001
8.166291277962422
Loss at iteration 0 8.166291277962422
8.562146107401855
8.19109995059271
8.266610874869738
8.160001436115943
8.395265616432745
8.158813660149871
8.379434110216904
8.696575091289688
8.397857265508783
7.3994563310680785
8.179616687365892
8.435547752942949
8.392279281992437
8.124019852304771
8.202649885764764
8.165440185382401
8.3814853720077
8.098436524342953
8.15640177232739
8.426805603370362
8.348880372047702
8.35722441269138
8.61472330264854
8.360668346838654
7.824234457122237
8.681769627130771
8.143883853364912
7.9315995723971575
8.248700185596501
8.157876274731265
8.25137394311077
8.146836891455402
8.140605795564902
8.236118122689327
8.84377856339526
8.263571826324771
8.316824509722744
8.29480398522289
```

8.696569798041692
8.691356625219576
8.061324842296715
8.394553050257143
8.304800237588195
8.146654641504835
8.299768915837259
7.923579395092095
8.359483857401012
8.0452838218287
8.276288426368746
8.281163241331841
8.290802041030584
8.35454250556192
8.65367981239891
7.296759790898806
8.015476114104509
8.730807538804822
8.113477125070792
7.8586347958307705
8.274577918889138
8.472018394913052
8.045414676465139
8.256094565729851
8.269189130462482
7.98450891164863
8.325920970887685
8.089806304940643
8.227830544479756
7.963235676070905
7.840428704679703
7.990686074199438
8.269584610650552
8.201767173086091
8.709517265657132
8.19592605620596
8.210373210920592
7.956492566385819
8.01738608323064
7.927468203115633
7.617859126652183
8.168610663002118
7.989612477330468
8.194927667851646
8.164291145781014
8.71766748060445
8.185508069171311
7.90660942525184

```
8.025576426423886
8.198691612433873
7.994530571986551
8.137927818600158
8.135837262907273
7.934273882368611
7.708597947151337
8.734069094505667
8.280135839797833
7.9420410730994
7.905483029922481
8.043512126562835
7.902074706759047
time taken: 0.0187990665435791
9.67545868739207
Loss at iteration 0 9.67545868739207
9.754081762831442
10.117507847382026
9.91161429337919
9.879733479510277
10.10465921381877
9.750738830949615
9.903979497273191
9.909875615048996
9.571783902255829
11.197001086675844
9.778464772852512
9.902969422029438
11.055920681754982
9.720464782022045
9.67935252147364
9.878121210066045
9.798052458707962
9.835133357792735
9.776328348288597
9.339955902175392
10.000646765014011
9.700550866702653
11.430995780462489
9.84929322610722
9.762545095005636
10.027872270910178
9.489349741793541
9.927725538211693
9.62285012665858
9.533428129635869
9.869387176748937
9.853095914016905
```

9.557807551870395
9.955430855950823
9.71602064339626
9.291995492470353
10.983919571084295
9.605436519621659
9.894132869817842
10.84983229243552
9.635424301892096
9.96801727571069
9.747745892536702
9.597886774658747
9.482111757530864
9.64837679573515
9.592662117148306
9.68794881220467
9.558249323060199
9.48589735698846
9.897303296975087
11.69448590706337
9.595812361747857
11.90087894974043
11.01551509607072
9.827973023262839
10.47237185110249
9.790876314398636
9.534823606110587
9.726557200788099
9.52228302415256
9.72882710104876
9.722859375999809
9.520148314460332
9.344743864218653
10.892871401951986
10.089149446118363
9.787345360867167
9.78362592277026
9.537778721477991
9.537281000506356
9.495624642881081
9.687617208373098
9.634191260613644
9.666626966953624
9.925842137842789
10.05920892790767
9.917705621561232
9.662242537668329
9.580559033554739

```
9.904052864097329
9.49655485202397
9.907242191999538
9.446088171271764
9.422004003147741
9.547662549762611
10.717145316037847
9.583892620686704
10.867610007343217
9.759550223299001
9.649053242436889
9.465423927796293
9.306169872047866
10.055444489487094
9.79756280512586
9.61290654934076
9.730167339402596
9.407266295326078
9.780222258556492
time taken: 0.016943931579589844
8.335391025805926
Loss at iteration 0 8.335391025805926
7.918073354919412
8.121155006953895
8.845183800558232
7.731619437402576
8.1044986487314
8.202942316473507
8.011402413830167
8.88784583688856
8.065214948884565
7.983569440740264
8.427027116805327
7.791224329446564
8.41695544700248
8.556421228158763
7.806497605041505
8.02911903978963
7.805276293774389
8.277311078452481
7.790681127443861
7.769961184697284
7.795186106806379
7.947559773297921
7.784274134178825
8.238134971338106
8.232983161824526
8.077387349956915
```

7.785648598731034
8.996729363789969
8.17316113460586
8.136584501404487
8.05198655600925
8.101025863612845
7.617230953988534
7.621754111065281
8.663895627353622
7.605821109431265
8.159826372278088
7.840500587887377
8.171169413147087
8.77712327796674
8.097097163583355
8.075832037412976
7.660270391802402
7.816395896637147
8.162064360483646
8.089169566055473
7.7196154910229
7.90591649145607
8.653582345189442
7.8380088617658945
7.903512020242981
8.111178802757033
8.84155492316285
8.008350877659655
8.251213558758243
7.985214084388282
8.488511855169955
8.156254605535468
8.601897381193867
7.671869648279757
8.004413962255475
7.811355614879507
8.132040545640615
7.661157280986126
8.121611711568367
8.950142422164886
7.469332350187118
7.582536285363925
8.346449778523
7.661320680567797
8.156263421630442
8.440743121361509
8.223775169151843
7.652162282621893

```
8.251687923586466
7.574223913213927
7.908976605320045
8.898841490657812
7.954472146397904
7.630906619743296
7.554079804753107
7.0485986554845645
7.483174889351353
7.733407908550981
7.754438287284134
7.974234016151873
7.50521114272285
7.611917402601353
8.057924090806106
7.828300936319378
8.622725088161529
7.072401338090329
8.014609080584998
8.05308406390914
7.718616957360968
7.6609510515358545
7.576342630834072
8.849837617653066
7.558980427137869
time taken: 0.011392593383789062
11.926937625809462
Loss at iteration 0 11.926937625809462
12.157625520915504
11.459871919658857
11.196264433833937
11.36926346394192
11.388447488906213
11.124867151984734
11.009082667839245
11.785090013032187
11.379567015426598
11.34186859265099
12.144069504984039
12.194762761553012
12.651374779324195
10.827106645894967
11.485594166663716
11.15355961095901
11.983037098963534
12.607331459276722
11.638928398497384
11.313563713999207
```

11.716429474934046
12.016805869881331
12.525830390219639
11.313297754738777
10.940196430630683
11.97753168612208
11.546961907229356
12.352216203666234
11.472536150849532
11.385354890903935
11.729981708321155
11.893834041419332
11.089635270160596
11.280663478911563
11.458056341087344
11.437948359190624
11.171520093813317
13.420197169441263
11.895949680604303
10.938397971028591
11.227737731361367
11.689282829845126
12.156435517982594
11.624617788353827
11.736750896575623
11.286111709809559
11.220098978826357
11.204358305362257
10.765802651552516
11.142151017895417
11.722971370883029
12.315313884765933
10.779504497772798
10.72229327780086
11.121805254704647
11.188914040354108
11.746720983535194
12.698461494903576
11.444326427806692
12.193657991836764
11.330765755352983
11.343431765004127
11.512350915497095
11.578682874701776
11.711421402003424
11.698469331107496
10.757144767452257
12.31266685203001

```
11.119692900686019
10.709926951979458
10.919718874899138
11.453469726934879
12.510706933309812
11.338529163687364
12.388483505692896
11.45634004350108
11.459689854498015
11.36621052532269
13.279500661696341
10.581011908801965
11.357651497528881
11.081407070899536
10.870815176034476
11.285121453497691
11.055791591316785
11.050520979692521
11.066085582314374
11.758816289462015
11.03942144602846
12.48908965347013
11.276140767883994
11.205954283464612
11.104477936628728
10.547442217477014
12.30310109141177
11.015542493465565
11.18278247730167
11.417861112281216
10.8384975778689
time taken: 0.00995779037475586
10.884217277526016
Loss at iteration 0 10.884217277526016
10.611220851374142
10.758598677731708
10.754500638061169
11.047455050817
10.582261470376519
11.0579136948627
10.487663031689003
10.711915640096239
10.65518305168999
11.01342326965164
10.628930185379797
10.38852453300145
10.580747517700006
10.50584418530682
```

10.621169224463625
10.366252285639916
10.433749581816622
10.686686014613263
10.626526664835849
10.621786852590551
11.189206619465148
10.417809934411551
10.456333260013025
10.671057526871188
10.291391882902031
10.526324004521364
10.530802900909505
10.954419007233168
10.952492281160392
10.424271823113404
10.536462364462572
10.752203965274958
11.096861655103046
10.460089222900262
10.379342185598393
10.548500588780996
10.744068037880723
10.339448690677946
10.518153272675791
10.594215106636128
10.451339467652325
10.321491250319209
10.47875548748006
10.512590927722316
10.628688159258457
10.969595261934773
10.305461812397363
10.3752953532353
10.468829766675343
10.321819237174044
10.734086109171503
10.233441766506076
10.367500542113708
10.734165561753969
10.662391186995789
10.37025278550347
10.263557243237209
10.489609585951603
10.303488229624232
10.4809922499948
10.439097144861387
10.4876301548302

```
10.373439686857752
10.896777508927808
10.37771360112048
10.323620503552835
10.235710319696654
10.226145538193327
10.30670343934153
10.51199005302217
10.374534003929176
10.399684333478378
10.215920878212886
10.104185667141449
10.381329983899239
10.576095181522817
10.488261033680786
10.227341470961797
10.595982172306185
10.337856686528417
10.452171270486115
10.243108426730602
10.393227933016119
10.336254213427718
10.064832431945417
10.403219306254414
10.283231738959596
10.352736793833627
10.212123936560767
10.238479586838151
10.152916626254298
10.373556220683486
10.475697887671512
10.222100062308966
10.428592581903038
10.259698818404802
10.172642941623852
10.394924688145053
10.208434252761682
time taken: 0.00872182846069336
{0: {'accuracy': 0.05674740484429066, 'precision': 0.2619589638581006, 'recall':
0.3490612940726907, 'f1_score': 0.040981377257036974}, 1: {'accuracy':
0.2664359861591695, 'precision': 0.33222706558351495, 'recall':
0.2168101932219159, 'f1_score': 0.1835994475095394}, 2: {'accuracy':
0.08512110726643599, 'precision': 0.34217636944239743, 'recall':
0.35433260922037174, 'f1_score': 0.05673853893836641}, 3: {'accuracy':
0.04913494809688582, 'precision': 0.3333333333333333, 'recall':
0.017001915708812262, 'f1_score': 0.032353611300979725}, 4: {'accuracy':
0.8685121107266436, 'precision': 0.3774733637747336, 'recall':
0.6258649486995944, 'f1_score': 0.38955948003394286}}
```

```
2024/10/07 15:25:51 INFO mlflow.tracking._tracking_service.client:  View run
method-sto-lr-0.0001 at: http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/88170
3814181867940/runs/5d092d7cc1684607a2f02fb12e95c571.
2024/10/07 15:25:51 INFO mlflow.tracking._tracking_service.client:  View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.
2024/10/07 15:25:51 INFO mlflow.tracking._tracking_service.client:  View
experiment at:
http://mlflow.ml.brain.cs.ait.ac.th/#/experiments/881703814181867940.
```

According the experiment results, the mini-batch gradient method with learning rate 0.001 is the best model.

3 7. Testing

Now, the best model will be searched and loaded to test on test set

[206]:
```
model_uri = f"models:/st125066-a3-model@dsai-ait"
model = mlflow.sklearn.load_model(model_uri)
```

```
Downloading artifacts: 100%|     | 5/5 [00:00<00:00,  9.75it/s]
```

[210]:
```
# do inference on the test set
yhat = model.predict(X_test.to_numpy())
```

[1004]:
```
# accuracy of the test set
model.accuracy(Y_true=y_test.to_numpy(),Y_pred=y_hat)
```

[1004]:
```
np.float64(1.5635440164345504)
```

[1005]:
```
# weighted precision of the test set
model.macro_metric(metric="precision",Y_true=y_test.
  ↪to_numpy(),Y_pred=y_hat,weighted=True)
```

[1005]:
```
np.float64(0.551287877583187)
```

[ ]:
```
# weighted
model.macro_metric(metric="recall",Y_true=y_test.
  ↪to_numpy(),Y_pred=y_hat,weighted=True)
```

[ ]:
```
# weighted f1 score
model.macro_metric(metric="f1",Y_true=y_test.
  ↪to_numpy(),Y_pred=y_hat,weighted=True)
```

4 8. Inference

[219]:
```
# predicted value
model.predict([X_test.iloc[0]])
```

[219]: array([0])

[215]:
```
# true value
y_test.iloc[0]
```

[215]: 0