

Created by "Harianto .T" At 04 March 2024 03:00 AM

Dataset Origin : <https://www.kaggle.com/datasets/sureshmecad/supplement-sales-prediction/data>

Others Link :

[https://drive.google.com/file/d/1v9IW7W\\_UgDmJWaWkZzQbiLT2MbKEl8i/view?usp=sharing](https://drive.google.com/file/d/1v9IW7W_UgDmJWaWkZzQbiLT2MbKEl8i/view?usp=sharing)

## Supplement Sales Analysis and Prediction

### Dataset Feature Overview

- ID (String) : Order Unique Identity.
- Store\_id (Integer) : Id to represent womart store entity.
- Store\_Type (String) : Type of store based on womart business strategic.
- Location\_Type (String) : Type of location of each store.
- Region\_Code (String) : Id of region where each store located.
- Date (String) : Day when transaction occurs on each store.
- Holiday (Integer) : Is it holiday or not.
- Discount (String) : it has a discount or not.
- Order (Integer) : Number of orders.
- Sales (Float) : Total revenue of each order.

```
In [63]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

```
In [64]: %cd /content/drive/My Drive/Colab Notebooks/RAKAMIN PERSONAL/learning/Supplement Sales Analysis

/content/drive/My Drive/Colab Notebooks/RAKAMIN PERSONAL/learning/Supplement Sales Analysis
```

```
In [65]: import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import skew, kurtosis, stats
from collections import Counter
from sklearn.preprocessing import LabelEncoder
%matplotlib inline
```

```
# Package Information
print('numpy version : ', np.__version__)
print('pandas version : ', pd.__version__)
print('seaborn version : ', sns.__version__)
```

```
numpy version : 1.25.2
pandas version : 1.5.3
seaborn version : 0.13.1
```

```
In [66]: #Konstanta
sns.set(rc={'figure.figsize':(20.7,8.27)})
```

```
sns.set_style("whitegrid")
sns.color_palette("dark")
plt.style.use("fivethirtyeight")
```

# 1. Load & Describe Data

```
In [67]: # Load datasets
dataset_raw = pd.read_csv('dataset/train.csv')
print(dataset_raw.shape)
dataset_raw.head()
```

(188340, 10)

```
Out[67]:
```

	ID	Store_id	Store_Type	Location_Type	Region_Code	Date	Holiday	Discount	#Order
0	T1000001	1	S1	L3	R1	2018-01-01	1	Yes	9
1	T1000002	253	S4	L2	R1	2018-01-01	1	Yes	60
2	T1000003	252	S3	L2	R1	2018-01-01	1	Yes	42
3	T1000004	251	S2	L3	R1	2018-01-01	1	Yes	23
4	T1000005	250	S2	L3	R4	2018-01-01	1	Yes	62

```
In [68]: # Information
dataset_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188340 entries, 0 to 188339
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   ID              188340 non-null object
 1   Store_id       188340 non-null int64
 2   Store_Type     188340 non-null object
 3   Location_Type  188340 non-null object
 4   Region_Code    188340 non-null object
 5   Date           188340 non-null object
 6   Holiday        188340 non-null int64
 7   Discount       188340 non-null object
 8   #Order         188340 non-null int64
 9   Sales          188340 non-null float64
dtypes: float64(1), int64(3), object(6)
memory usage: 14.4+ MB
```

There are no null values in the entire dataset

```
In [69]: #Summary Statistic
dataset_raw.describe()
```

Out[69]:

	Store_id	Holiday	#Order	Sales
<b>count</b>	188340.000000	188340.000000	188340.000000	188340.000000
<b>mean</b>	183.000000	0.131783	68.205692	42784.327982
<b>std</b>	105.366308	0.338256	30.467415	18456.708302
<b>min</b>	1.000000	0.000000	0.000000	0.000000
<b>25%</b>	92.000000	0.000000	48.000000	30426.000000
<b>50%</b>	183.000000	0.000000	63.000000	39678.000000
<b>75%</b>	274.000000	0.000000	82.000000	51909.000000
<b>max</b>	365.000000	1.000000	371.000000	247215.000000

## Data Preparation

Plan:

1. Remove ID
2. Lowercase Column Name
3. Change type object to date on 'date' column
4. Split date to (Year, Month, Day)
5. Rename #Order to Order
6. Convert discount value to numeric
7. Convert feature to object
8. Remove Duplicates & Missing Value

```
In [70]: #Remove ID from Dataset
dataset_raw = dataset_raw.drop('ID', axis=1) # Cause it not relevant
```

```
In [71]: #Lowercase Column Name
dataset_raw.rename(columns={col: col.lower() for col in dataset_raw.columns}, inplace=True)
```

```
In [72]: #Convert data type to date
dataset_raw['date'] = pd.to_datetime(dataset_raw.date)
```

```
In [73]: #Rename #order = order
dataset_raw.rename(columns={'#order': 'order'}, inplace=True)
```

```
In [74]: # Rename the values on "discount"
dataset_raw['discount'] = dataset_raw['discount'].map({'Yes': 1, 'No': 0})
```

```
In [75]: # Changing data types
dataset_raw[['store_id', 'holiday', 'discount']] = dataset_raw[['store_id', 'holiday', 'discount']].astype(int)
```

```
In [76]: # Describe Categorical Feature
dataset_raw[['store_id', 'holiday', 'discount']].describe()
```

Out[76]:

	store_id	holiday	discount
count	188340	188340	188340
unique	365	2	2
top	1	0	0
freq	516	163520	104051

```
In [77]: # Remove Duplicates
print('Number of duplicates:', len(dataset_raw[dataset_raw.duplicated()]))

# Remove missing value
print('Number of missing values:', dataset_raw.isnull().sum().sum())
```

Number of duplicates: 0  
Number of missing values: 0

```
In [78]: #Create new feature from date feature
dataset_raw['year']=dataset_raw['date'].dt.year
dataset_raw['month']=dataset_raw['date'].dt.month
dataset_raw['day_of_week'] = dataset_raw['date'].dt.dayofweek
dataset_raw['week'] = dataset_raw['date'].dt.day.apply(lambda x: (x - 1) // 7 + 1)
dataset_raw['quarter'] = dataset_raw['date'].dt.quarter

# dataset_raw = dataset_raw.drop(['date'], axis = 1) #drop this column cause we don't need it
```

```
In [79]: dataset_raw
```

Out[79]:

	store_id	store_type	location_type	region_code	date	holiday	discount	order	sale
0	1	S1	L3	R1	2018-01-01	1	1	9	7011.84
1	253	S4	L2	R1	2018-01-01	1	1	60	51789.11
2	252	S3	L2	R1	2018-01-01	1	1	42	36868.20
3	251	S2	L3	R1	2018-01-01	1	1	23	19715.10
4	250	S2	L3	R4	2018-01-01	1	1	62	45614.51
...	...	...	...	...	...	...	...	...	...
188335	149	S2	L3	R2	2019-05-31	1	1	51	37272.00
188336	153	S4	L2	R1	2019-05-31	1	0	90	54572.64
188337	154	S1	L3	R2	2019-05-31	1	0	56	31624.50
188338	155	S3	L1	R2	2019-05-31	1	1	70	49162.41
188339	152	S2	L1	R1	2019-05-31	1	0	47	37977.00

188340 rows × 14 columns



In [80]:

```
# Data types
print('Data type per variable:')
print(dataset_raw.dtypes)

Data type per variable:
store_id          object
store_type        object
location_type     object
region_code       object
date              datetime64[ns]
holiday           object
discount          object
order            int64
sales            float64
year             int64
month            int64
day_of_week       int64
week             int64
quarter          int64
dtype: object
```

# EDA

## Insight

In [81]: `dataset_raw.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188340 entries, 0 to 188339
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   store_id            188340 non-null object
 1   store_type          188340 non-null object
 2   location_type       188340 non-null object
 3   region_code         188340 non-null object
 4   date                188340 non-null datetime64[ns]
 5   holiday             188340 non-null object
 6   discount            188340 non-null object
 7   order              188340 non-null int64
 8   sales              188340 non-null float64
 9   year               188340 non-null int64
10  month              188340 non-null int64
11  day_of_week        188340 non-null int64
12  week               188340 non-null int64
13  quarter            188340 non-null int64
dtypes: datetime64[ns](1), float64(1), int64(6), object(6)
memory usage: 20.1+ MB
```

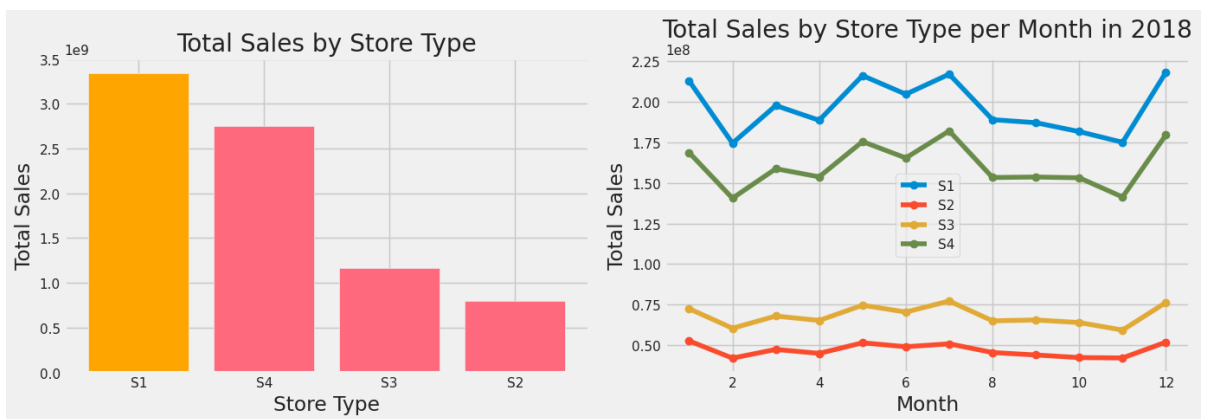
```
In [82]: # Total Sales By Store Type
plt.figure(figsize=(14, 5))

# Subplot 1: Total Sales By Store Type
plt.subplot(1, 2, 1)
sales_by_store_type = dataset_raw.groupby('store_type')['sales'].sum()
sales_by_store_type = sales_by_store_type.sort_values(ascending=False)
plt.bar(sales_by_store_type.index, sales_by_store_type.values, color='#ff6b81')
max_sales = sales_by_store_type.max()
max_index = sales_by_store_type.idxmax()
plt.bar(max_index, max_sales, color='#ffa502')
plt.xlabel('Store Type')
plt.ylabel('Total Sales')
plt.title('Total Sales by Store Type')
plt.grid(True)

# Subplot 2: Total Sales by Store Type per Month in 2018
plt.subplot(1, 2, 2)
df_2018 = dataset_raw[dataset_raw['year'] == 2018]
sales_by_store_type_month_2018 = df_2018.groupby(['month', 'store_type'])['sales'].
for store_type in sales_by_store_type_month_2018.columns:
    plt.plot(sales_by_store_type_month_2018.index, sales_by_store_type_month_2018[s
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.title('Total Sales by Store Type per Month in 2018')
plt.legend()
plt.grid(True)

# Adjust Layout
plt.tight_layout()

# Show plot
plt.show()
```



Store Type S1 consistently show the highest sales, followed by Store Types S2 and S3, while Store Type S4 has the lowest sales. The sales figures for all store types generally follow a pattern of **increased sales during the summer months and decreased sales towards the end of the year, except for Store Type S4, which experiences a peak in December.**

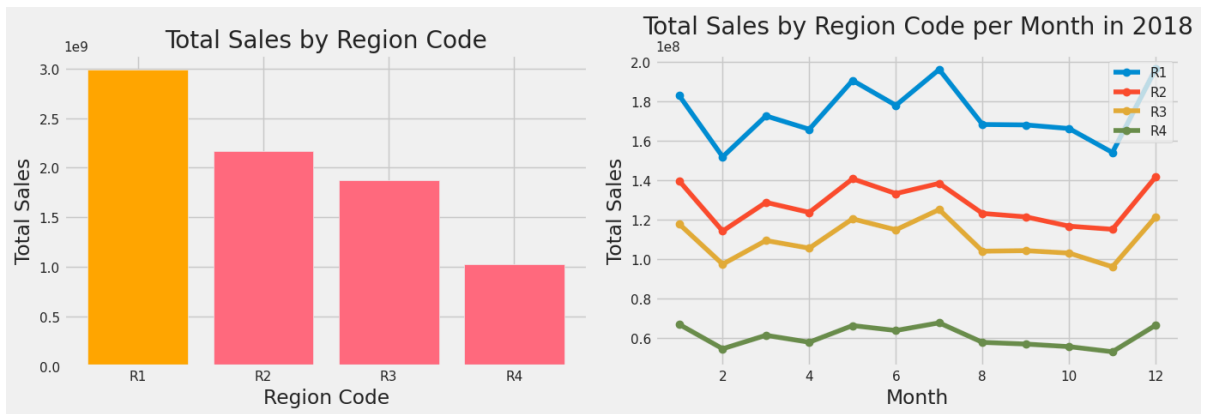
```
In [83]: # Total Sales By Region Code
plt.figure(figsize=(14, 5))

# Subplot 1: Total Sales By Region Code
plt.subplot(1, 2, 1)
sales_by_region_code = dataset_raw.groupby('region_code')['sales'].sum()
sales_by_region_code = sales_by_region_code.sort_values(ascending=False)
plt.bar(sales_by_region_code.index, sales_by_region_code.values, color='#ff6b81')
max_sales = sales_by_region_code.max()
max_index = sales_by_region_code.idxmax()
plt.bar(max_index, max_sales, color='#ffa502')
plt.xlabel('Region Code')
plt.ylabel('Total Sales')
plt.title('Total Sales by Region Code')
plt.grid(True)

# Subplot 2: Total Sales by Region Code per Month in 2018
plt.subplot(1, 2, 2)
df_2018 = dataset_raw[dataset_raw['year'] == 2018]
sales_by_region_code_month_2018 = df_2018.groupby(['month', 'region_code'])['sales']
for region_code in sales_by_region_code_month_2018.columns:
    plt.plot(sales_by_region_code_month_2018.index, sales_by_region_code_month_2018[region_code])
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.title('Total Sales by Region Code per Month in 2018')
plt.legend()
plt.grid(True)

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```



There are differences in the volume of sales between regions. Some regions consistently have higher sales compared to others.

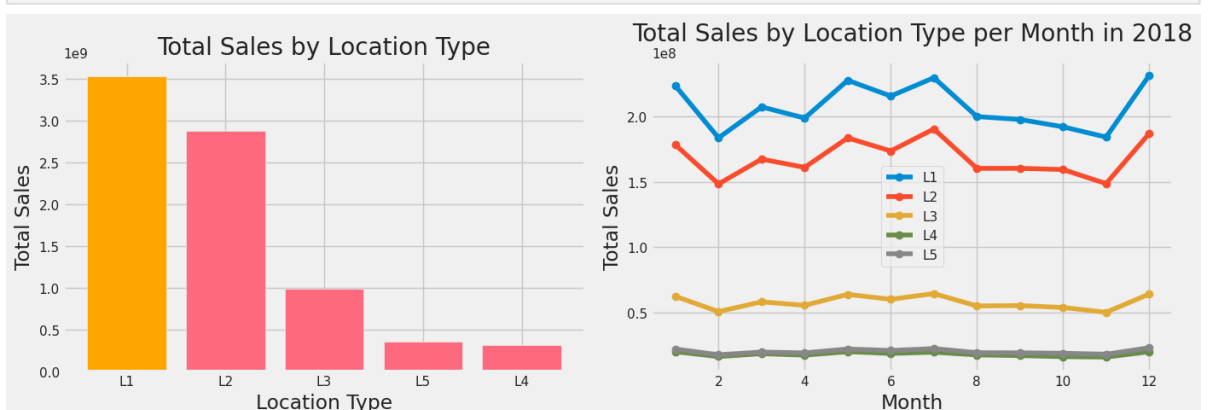
```
In [84]: # Total Sales By Location Type
plt.figure(figsize=(14, 5))

# Subplot 1: Total Sales By Location Type
plt.subplot(1, 2, 1)
sales_by_location_type = dataset_raw.groupby('location_type')['sales'].sum()
sales_by_location_type = sales_by_location_type.sort_values(ascending=False)
plt.bar(sales_by_location_type.index, sales_by_location_type.values, color='#ff6b81')
max_sales = sales_by_location_type.max()
max_index = sales_by_location_type.idxmax()
plt.bar(max_index, max_sales, color='#ffa502')
plt.xlabel('Location Type')
plt.ylabel('Total Sales')
plt.title('Total Sales by Location Type')
plt.grid(True)

# Subplot 2: Total Sales by Location Type per Month in 2018
plt.subplot(1, 2, 2)
df_2018 = dataset_raw[dataset_raw['year'] == 2018]
sales_by_location_type_month_2018 = df_2018.groupby(['month', 'location_type'])['sales'].sum()
for location_type in sales_by_location_type_month_2018.columns:
    plt.plot(sales_by_location_type_month_2018.index, sales_by_location_type_month_2018[location_type])
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.title('Total Sales by Location Type per Month in 2018')
plt.legend()
plt.grid(True)

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```





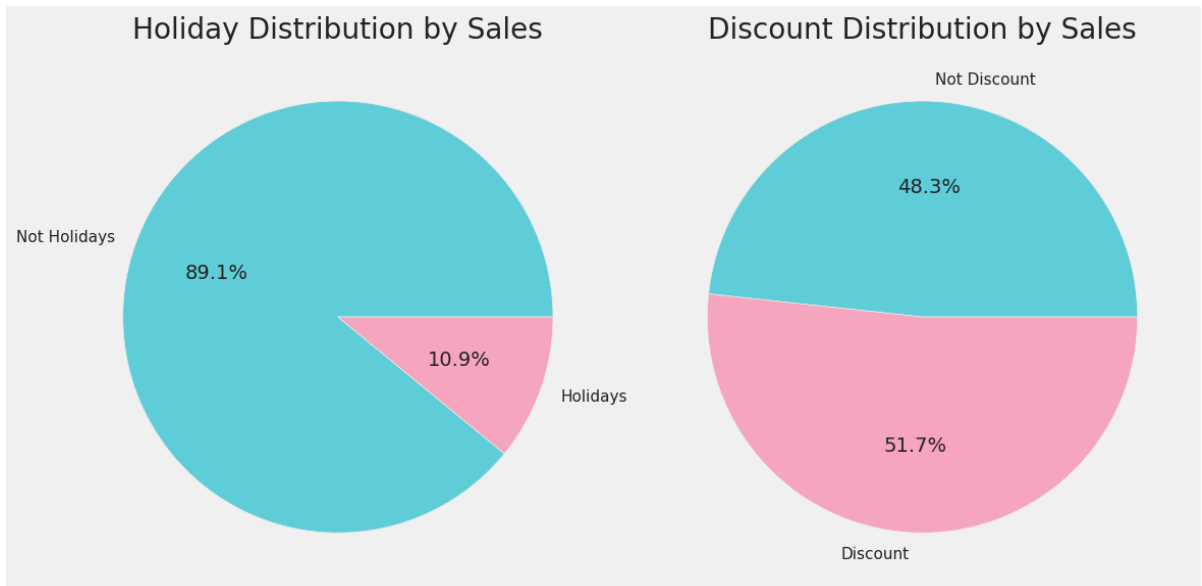
Location types L1 and L2 consistently have higher sales compared to L3, L4, and L5. L1 and L2 typically exhibit the highest sales figures, followed by a decline in sales for L3, L4, and L5.

```
In [85]: #Holiday & Discount Distribution by Sales
plt.figure(figsize=(12, 6))

# Subplot 1: Distribusi Penjualan berdasarkan Hari Libur
plt.subplot(1, 2, 1)
sales_by_holiday = dataset_raw.groupby('holiday')['sales'].sum()
plt.pie(sales_by_holiday.values, labels=['Not Holidays', 'Holidays'], colors=['#63c9e0', '#f08080'],
plt.title('Holiday Distribution by Sales')

# Subplot 2: Distribusi Penjualan berdasarkan Diskon
plt.subplot(1, 2, 2)
sales_by_discount = dataset_raw.groupby('discount')['sales'].sum()
plt.pie(sales_by_discount.values, labels=['Not Discount', 'Discount'], colors=['#63c9e0', '#f08080'],
plt.title('Discount Distribution by Sales')

plt.tight_layout()
plt.show()
```

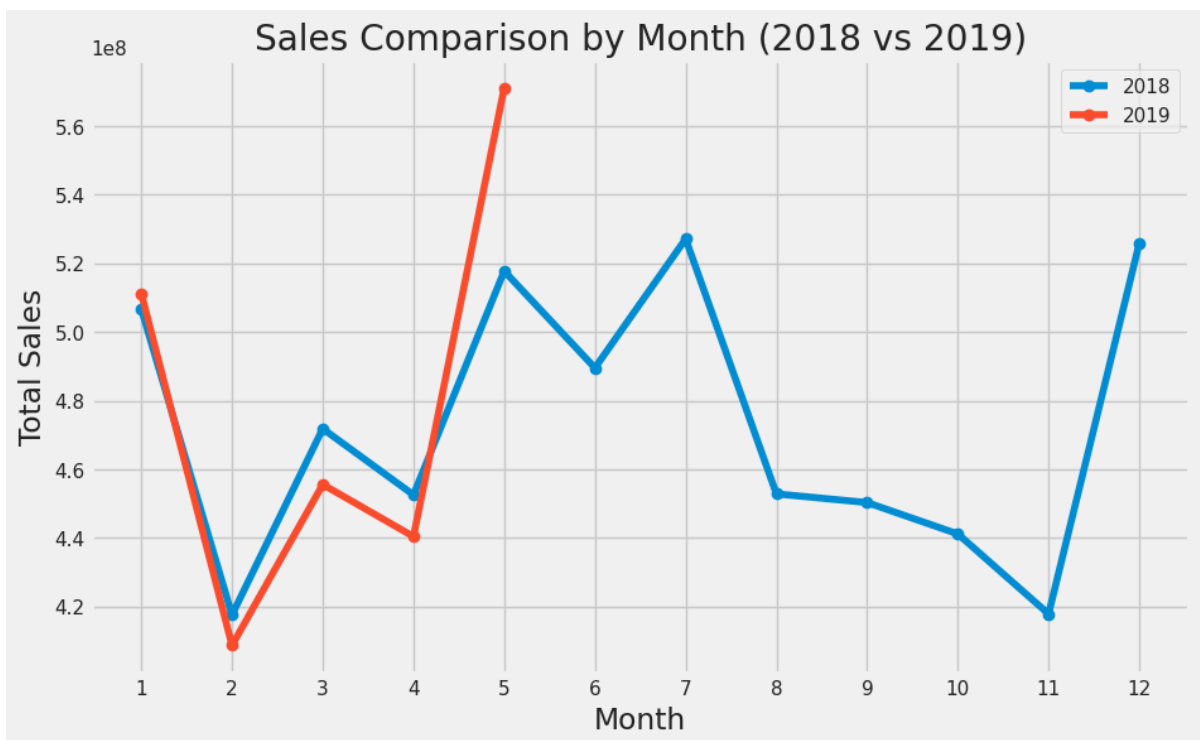


```
In [86]: df = dataset_raw

# Slicing data untuk tahun 2018 dan 2019
df_2018 = df[df['year'] == 2018]
df_2019 = df[df['year'] == 2019]

# Mengelompokkan penjualan berdasarkan bulan untuk masing-masing tahun
sales_by_month_2018 = df_2018.groupby('month')['sales'].sum()
sales_by_month_2019 = df_2019.groupby('month')['sales'].sum()

# Membuat plot
plt.figure(figsize=(10, 6))
plt.plot(sales_by_month_2018.index, sales_by_month_2018.values, marker='o', label='2018')
plt.plot(sales_by_month_2019.index, sales_by_month_2019.values, marker='o', label='2019')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.title('Sales Comparison by Month (2018 vs 2019)')
plt.xticks(range(1, 13))
plt.legend()
plt.grid(True)
plt.show()
```



```
In [87]: # Sales by Weekday
df_2018 = df.loc[(df['year'] == 2018) & (df['month'].between(1, 5))]

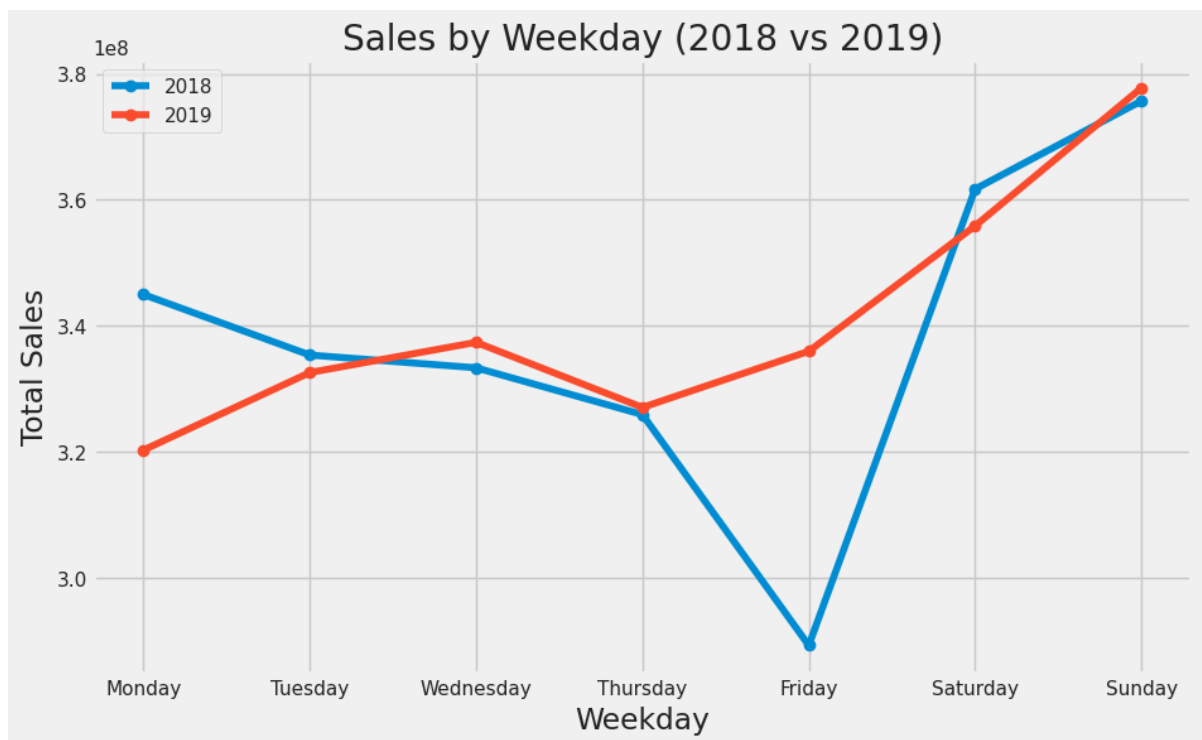
df_2019 = df.loc[(df['year'] == 2019) & (df['month'].between(1, 5))]

# Mengelompokkan penjualan berdasarkan hari dalam seminggu untuk tahun 2018
sales_by_weekday_2018 = df_2018.groupby('day_of_week')['sales'].sum()
# Mengelompokkan penjualan berdasarkan hari dalam seminggu untuk tahun 2019
sales_by_weekday_2019 = df_2019.groupby('day_of_week')['sales'].sum()

# Membuat plot garis untuk tahun 2018
plt.figure(figsize=(10, 6))
plt.plot(sales_by_weekday_2018.index, sales_by_weekday_2018.values, marker='o', label='2018')

# Membuat plot garis untuk tahun 2019
plt.plot(sales_by_weekday_2019.index, sales_by_weekday_2019.values, marker='o', label='2019')

plt.xlabel('Weekday')
plt.ylabel('Total Sales')
plt.title('Sales by Weekday (2018 vs 2019)')
plt.xticks(range(7), ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
plt.legend()
plt.grid(True)
plt.show()
```



```
In [88]: # Sales by Weekday
df_2018 = df.loc[(df['year'] == 2018) & (df['month'].between(1, 5))]

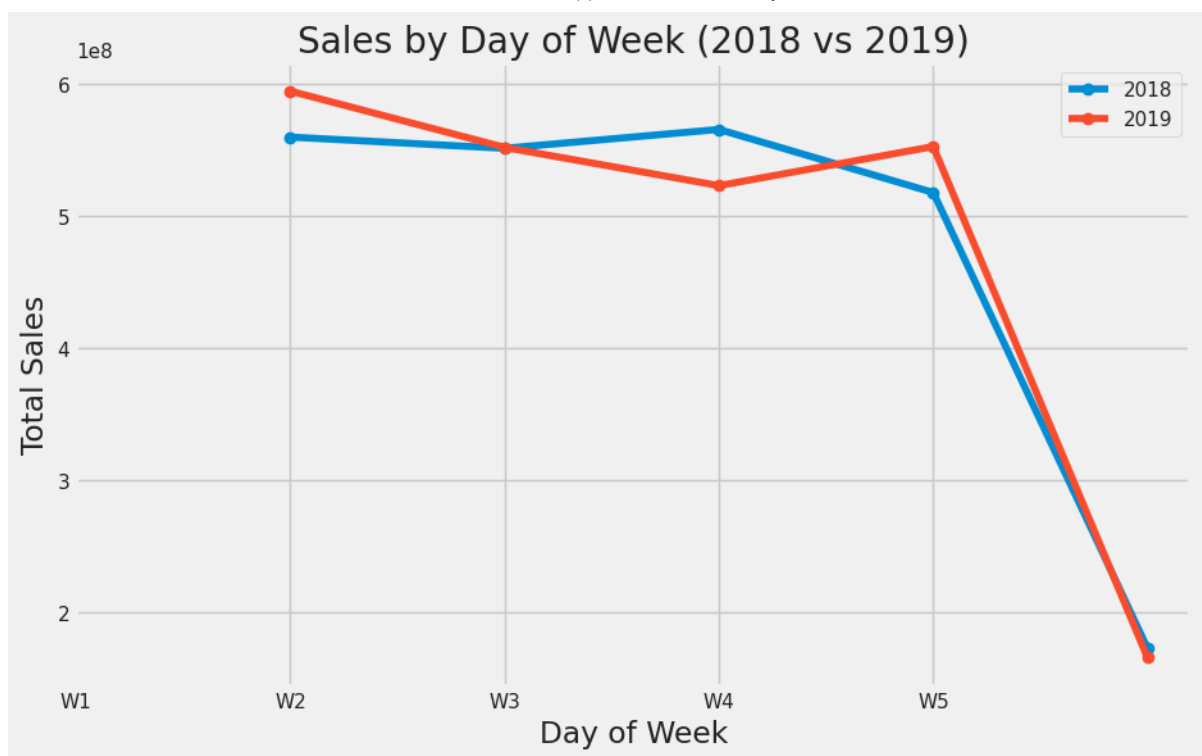
df_2019 = df.loc[(df['year'] == 2019) & (df['month'].between(1, 5))]

# Mengelompokkan penjualan berdasarkan hari dalam seminggu untuk tahun 2018
sales_by_weekday_2018 = df_2018.groupby('week')['sales'].sum()
# Mengelompokkan penjualan berdasarkan hari dalam seminggu untuk tahun 2019
sales_by_weekday_2019 = df_2019.groupby('week')['sales'].sum()

# Membuat plot garis untuk tahun 2018
plt.figure(figsize=(10, 6))
plt.plot(sales_by_weekday_2018.index, sales_by_weekday_2018.values, marker='o', label='2018')

# Membuat plot garis untuk tahun 2019
plt.plot(sales_by_weekday_2019.index, sales_by_weekday_2019.values, marker='o', label='2019')

plt.xlabel('Day of Week')
plt.ylabel('Total Sales')
plt.title('Sales by Day of Week (2018 vs 2019)')
plt.xticks(range(5), ['W1', 'W2', 'W3', 'W4', 'W5'])
plt.legend()
plt.grid(True)
plt.show()
```



### Distribution and Outlier

```
In [89]: # Convert back feature to integer
dataset_raw[['store_id', 'holiday', 'discount']] = dataset_raw[['store_id', 'holiday', 'discount']].astype(int)
```

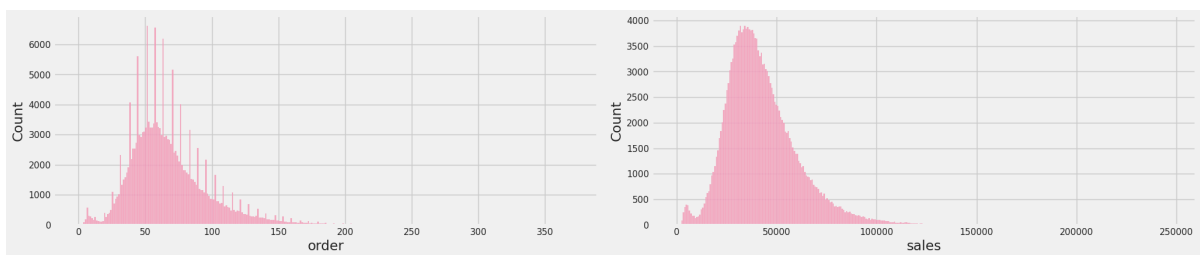
```
In [90]: # Data types
print('Data type per variable:')
print(dataset_raw.dtypes)
```

```
Data type per variable:
store_id          int64
store_type        object
location_type     object
region_code       object
date              datetime64[ns]
holiday           int64
discount          int64
order            int64
sales            float64
year             int64
month            int64
day_of_week       int64
week             int64
quarter          int64
dtype: object
```

```
In [91]: #Distribution on Feature Order & Sales
def plot_distribution(data):
    features = ['order', 'sales']
    for i in range(0, len(features)):
        plt.subplot(2, 2, i+1)
        sns.histplot(x=data[features[i]], color='#f78fb3')
        plt.xlabel(features[i])

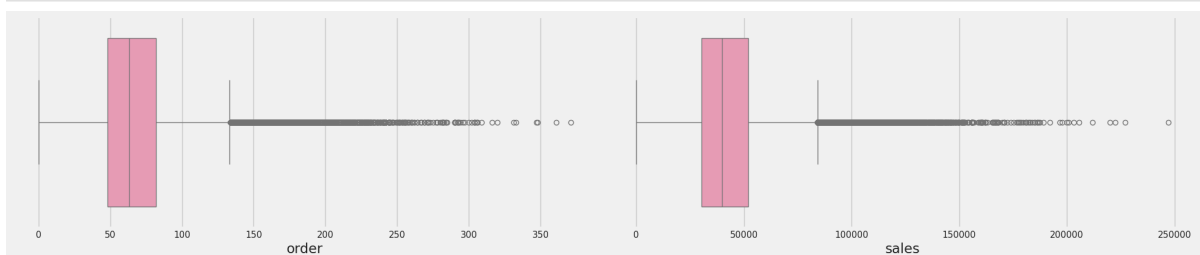
    plt.tight_layout()

plot_distribution(dataset_raw)
```



```
In [92]: #Outlier on Feature Order & Sales
def plot_outliner(data):
    features = ['order', 'sales']
    plt.subplots_adjust(hspace=0.5)
    for i in range(0, len(features)):
        plt.subplot(2, 2, i+1)
        sns.boxplot(x=data[features[i]], color='#f78fb3')
        plt.xlabel(features[i])
    plt.tight_layout()

plot_outliner(dataset_raw)
```



## Feature Engineering

```
In [93]: # Dealing with the outlier
def handling_outlier(dataset, feature):
    for f in feature:
        q1 = dataset[f].quantile(0.25)
        q3 = dataset[f].quantile(0.75)
        iqr = q3-q1
        Lower_tail = q1 - 1.5 * iqr
        Upper_tail = q3 + 1.5 * iqr
        med = np.median(dataset[f])
        for i in dataset[f]:
            if i > Upper_tail or i < Lower_tail:
                dataset[f] = dataset[f].replace(i, med)

handling_outlier(dataset_raw, ['sales', 'order'])
```

```
In [94]: #Menghapus Outlier pada distribusi Normal
from scipy import stats

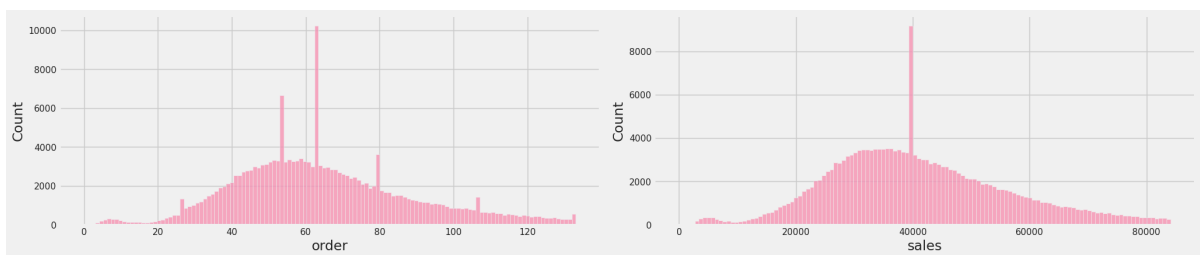
print("Before removing outlier: ", len(dataset_raw))

for num in ['sales', 'order']:
    z_scores = np.abs(stats.zscore(dataset_raw[num]))
    dataset_raw = dataset_raw[z_scores <= 3]

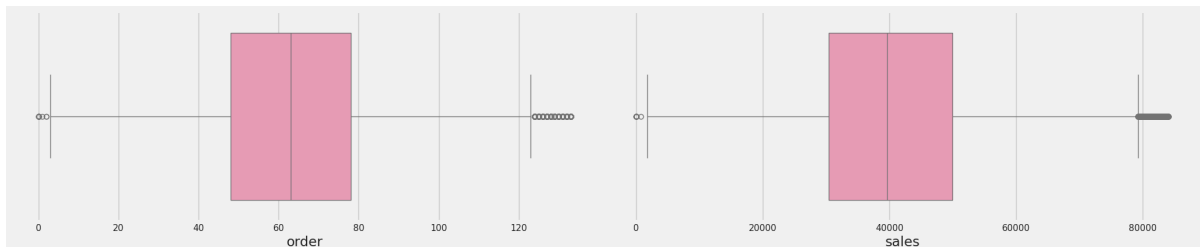
print("After removing outlier: ", len(dataset_raw))
```

Before removing outlier: 188340  
After removing outlier: 188340

```
In [95]: #Check Distribution
plot_distribution(dataset_raw)
```



```
In [96]: #Check Outlier
plot_outliner(dataset_raw)
```



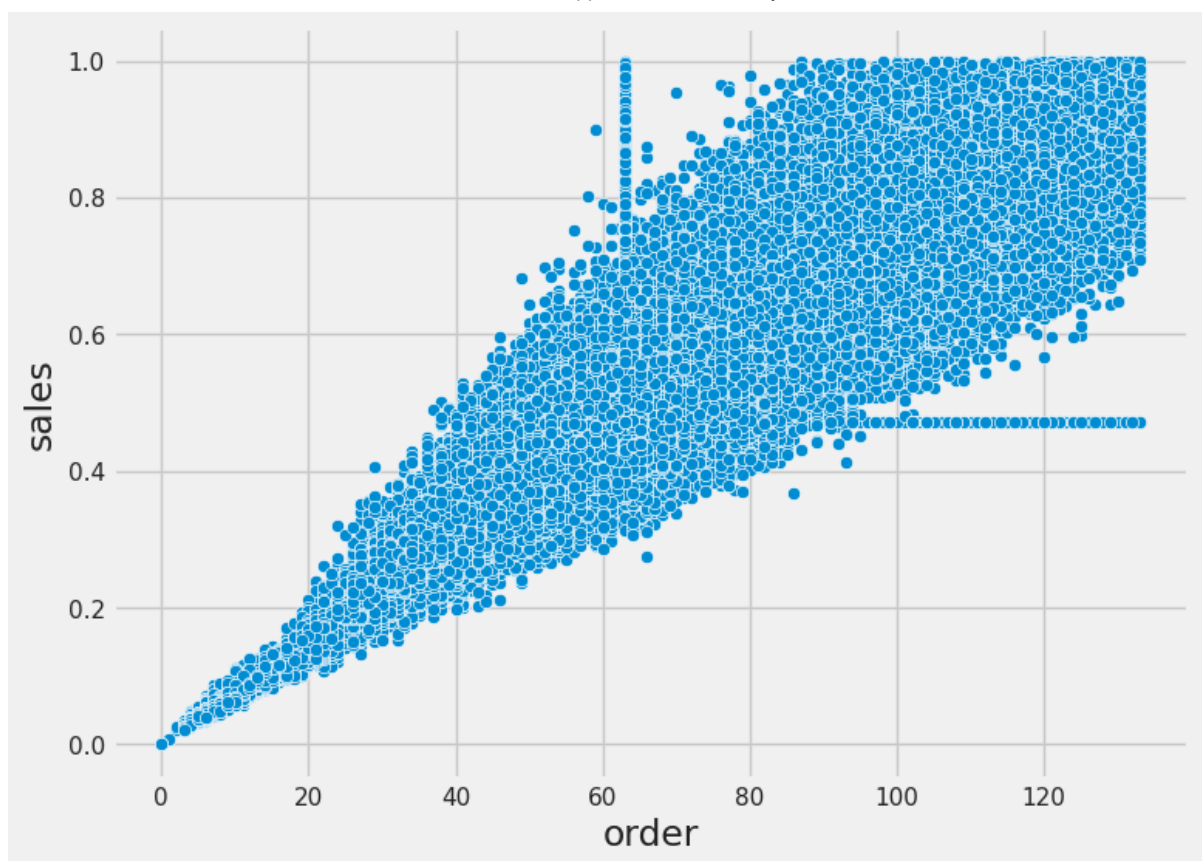
```
In [97]: from sklearn.preprocessing import MinMaxScaler
#Normalisasi Target Feature
dataset_raw['sales'] = MinMaxScaler().fit_transform(np.array(dataset_raw['sales'])).
```

```
In [98]: # Do One Hot Encoding to column with object data type
dataset_raw = pd.get_dummies(dataset_raw, columns=['store_type', 'location_type', 'region_code'])
dataset_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 188340 entries, 0 to 188339
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   store_id              188340 non-null  int64
1   date                 188340 non-null  datetime64[ns]
2   holiday              188340 non-null  int64
3   discount             188340 non-null  int64
4   order               188340 non-null  int64
5   sales               188340 non-null  float64
6   year                188340 non-null  int64
7   month               188340 non-null  int64
8   day_of_week         188340 non-null  int64
9   week                188340 non-null  int64
10  quarter             188340 non-null  int64
11  store_type_S2       188340 non-null  uint8
12  store_type_S3       188340 non-null  uint8
13  store_type_S4       188340 non-null  uint8
14  location_type_L2    188340 non-null  uint8
15  location_type_L3    188340 non-null  uint8
16  location_type_L4    188340 non-null  uint8
17  location_type_L5    188340 non-null  uint8
18  region_code_R2      188340 non-null  uint8
19  region_code_R3      188340 non-null  uint8
20  region_code_R4      188340 non-null  uint8
dtypes: datetime64[ns](1), float64(1), int64(9), uint8(10)
memory usage: 19.0 MB
```

```
In [99]: #Strength correlation between order and sales
plt.figure(figsize=(8, 6))
ax = sns.scatterplot(x="order", y="sales", data=dataset_raw)
plt.plot()
```

```
Out[99]: []
```

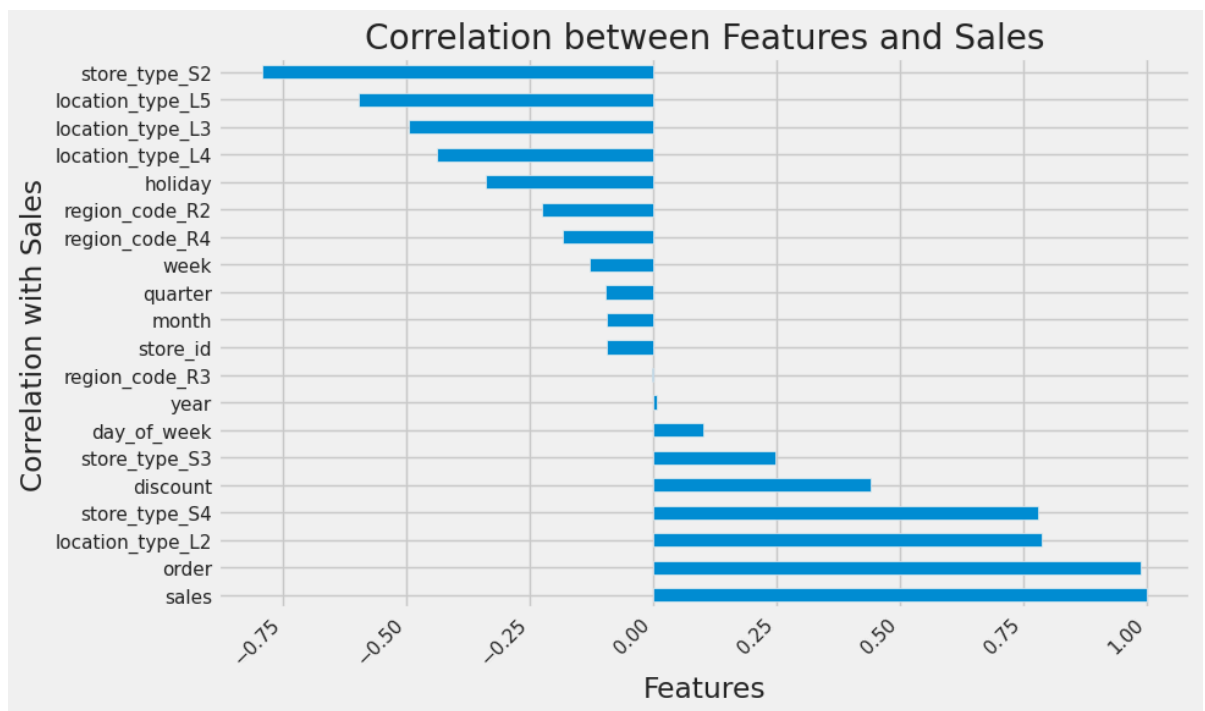


In [100...

```
import seaborn as sns
correlation_matrix = dataset_raw.corr()

# Count correlation between feature and label
correlation = correlation_matrix.corr()['sales'].sort_values(ascending=False)

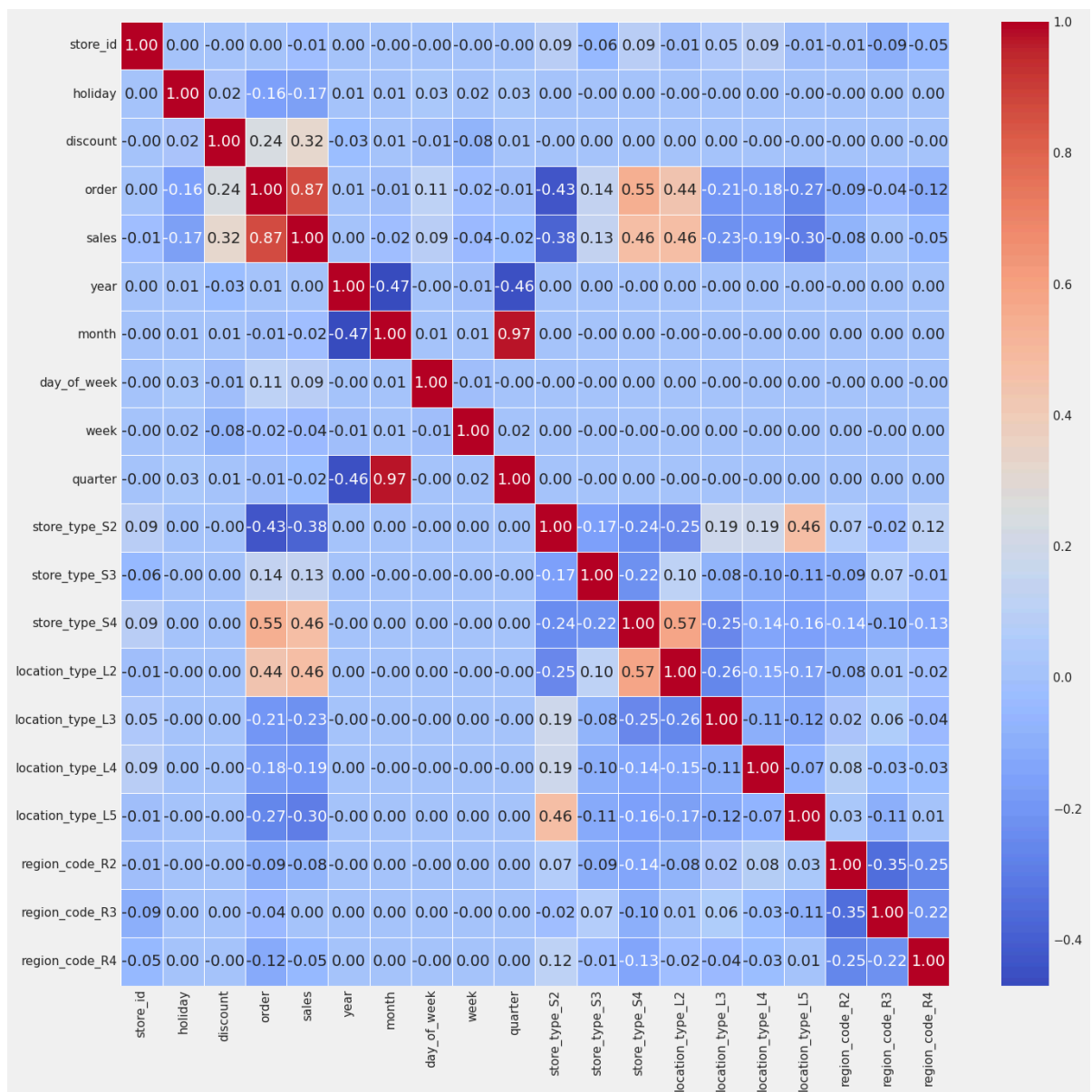
# Memvisualisasikan korelasi
plt.figure(figsize=(10, 6))
correlation.plot(kind='barh')
plt.xlabel('Features')
plt.ylabel('Correlation with Sales')
plt.title('Correlation between Features and Sales')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



In [101...

```
#Heatmap
plt.figure(figsize=(15,15))
sns.heatmap(correlation_matrix,annot=True,linewidths=0.7,fmt=".2f",cmap="coolwarm")
plt.show()
```





## Prediction Model

### Feature Important

```
In [102... # Define Feature and Label
clean_df = dataset_raw.drop(['store_id', 'year', 'date', 'region_code_R3'], axis=1)
sales_df = dataset_raw.copy() #for arima model

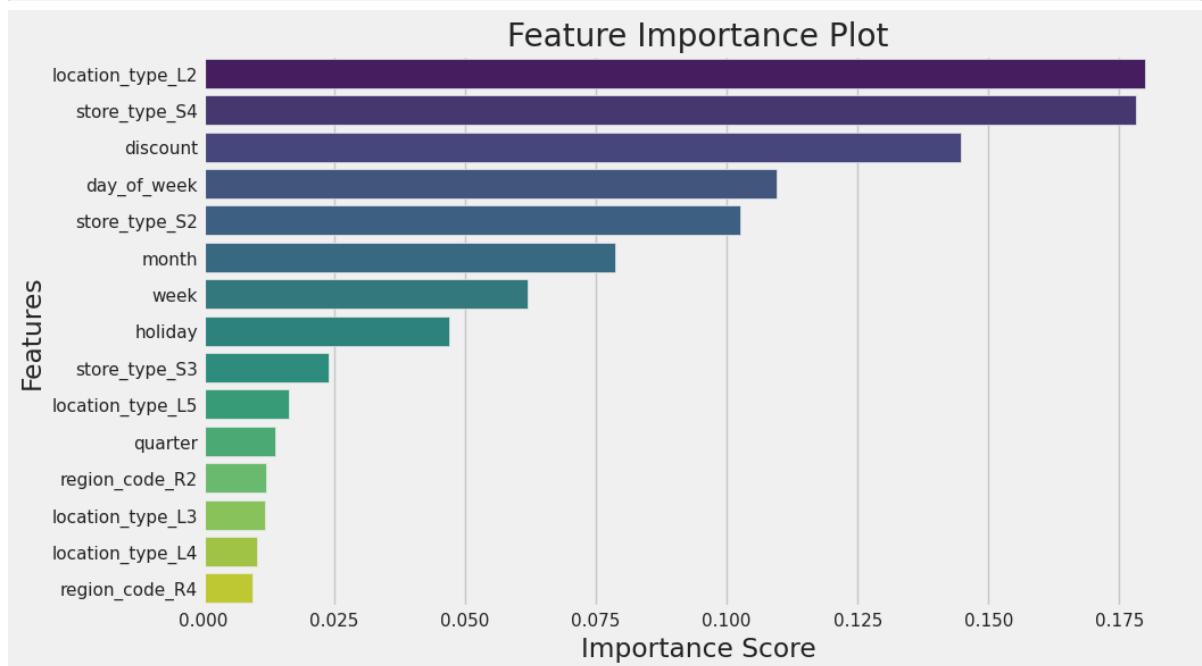
y = clean_df['sales']
x = clean_df.drop(['sales', 'order'], axis=1)
```

```
In [103... from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor()
rf_model.fit(x, y)
feature_importance = rf_model.feature_importances_
feature_importance_df = pd.DataFrame({'Feature': x.columns, 'Importance': feature_i
# feature_importance_df
```

```
In [104... #Visualize Feature Important
sorted_df = feature_importance_df.sort_values(by='Importance', ascending=False)
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=sorted_df, palette='viridis')
plt.xlabel('Importance Score')
plt.ylabel('Features')
plt.title('Feature Importance Plot')
plt.show()
```



```
In [105... #Splitting data 80:20
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x,y,test_size=0.2, random_state=42)
```

## Modeling

```
In [106... # !pip install xgboost
# !pip install lightgbm
!pip install catboost
```

Collecting catboost

Downloading catboost-1.2.3-cp310-cp310-manylinux2014\_x86\_64.whl (98.5 MB)

98.5/98.5 MB 4.6 MB/s eta 0:00:00

Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from catboost) (0.20.1)

Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from catboost) (3.7.1)

Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.25.2)

Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.5.3)

Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from catboost) (1.11.4)

Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from catboost) (5.15.0)

Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from catboost) (1.16.0)

Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2023.4)

Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.2.0)

Requirement already satisfied: cyclers>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (4.49.0)

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.4.5)

Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (23.2)

Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (9.4.0)

Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (3.1.1)

Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly->catboost) (8.2.3)

Installing collected packages: catboost

Successfully installed catboost-1.2.3

In [107...

```
import pandas as pd
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet, LassoCV
from sklearn.neighbors import KNeighborsRegressor
from sklearn.neural_network import MLPRegressor
from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor
from xgboost import XGBRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import AdaBoostRegressor
import pickle

regression_models = [
    (LinearRegression(), 'LinearRegression'),
    (Ridge(), 'RidgeRegression'),
    (Lasso(), 'LassoRegression'),
    (ElasticNet(), 'ElasticNetRegression'),
    (LassoCV(), 'LassoCVRegression'),
    (ElasticNetCV(), 'ElasticNetCVRegression'),
    (DecisionTreeRegressor(), 'DecisionTreeRegressor'),
    (RandomForestRegressor(), 'RandomForestRegressor'),
```

```
(AdaBoostRegressor(), 'AdaBoostRegressor'),
(LGBMRegressor(), 'LGBMRegressor'),
(GradientBoostingRegressor(), 'GradientBoostingRegressor'),
(XGBRegressor(), 'XGBoostRegression'),
(CatBoostRegressor(), 'CatBoostRegressor'),
# (KNeighborsRegressor(), 'KNeighbors Regression'), # too Long
# (MLPRegressor(), 'MLP Regression'), # too Long
]

results = []

for model, label in regression_models:
    print("Progress ", label)
    model.fit(X_train, y_train)
    with open('models/'+label+'_model.pkl', 'wb') as f:
        pickle.dump(model, f)
    y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    results.append([label, mse, mae, r2])

df_results = pd.DataFrame(results, columns=['Model', 'MSE', 'MAE', 'R-squared'])
```

```

Progress LinearRegression
Progress RidgeRegression
Progress LassoRegression
Progress ElasticNetRegression
Progress LassoCVRegression
Progress ElasticNetCVRegression
Progress DecisionTreeRegressor
Progress RandomForestRegressor
Progress AdaBoostRegressor
Progress LGBMRegressor
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing
was 0.009789 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 53
[LightGBM] [Info] Number of data points in the train set: 150672, number of used f
eatures: 15
[LightGBM] [Info] Start training from score 0.485958
Progress GradientBoostingRegressor
Progress XGBoostRegression
Progress CatBoostRegressor
Learning rate set to 0.090429
0:      learn: 0.1708135      total: 71.4ms      remaining: 1m 11s
1:      learn: 0.1639279      total: 89.2ms      remaining: 44.5s
2:      learn: 0.1580035      total: 107ms       remaining: 35.7s
3:      learn: 0.1529278      total: 126ms       remaining: 31.5s
4:      learn: 0.1485432      total: 145ms       remaining: 28.9s
5:      learn: 0.1446942      total: 163ms       remaining: 27s
6:      learn: 0.1414089      total: 182ms       remaining: 25.8s
7:      learn: 0.1386612      total: 200ms       remaining: 24.8s
8:      learn: 0.1362818      total: 219ms       remaining: 24.2s
9:      learn: 0.1340041      total: 239ms       remaining: 23.6s
10:     learn: 0.1320555      total: 259ms       remaining: 23.3s
11:     learn: 0.1303934      total: 287ms       remaining: 23.6s
12:     learn: 0.1289907      total: 317ms       remaining: 24s
13:     learn: 0.1277017      total: 336ms       remaining: 23.6s
14:     learn: 0.1265990      total: 354ms       remaining: 23.2s
15:     learn: 0.1255927      total: 373ms       remaining: 22.9s
16:     learn: 0.1247755      total: 391ms       remaining: 22.6s
17:     learn: 0.1240481      total: 410ms       remaining: 22.4s
18:     learn: 0.1235007      total: 429ms       remaining: 22.1s
19:     learn: 0.1227468      total: 447ms       remaining: 21.9s
20:     learn: 0.1221018      total: 466ms       remaining: 21.7s
21:     learn: 0.1215684      total: 488ms       remaining: 21.7s
22:     learn: 0.1209809      total: 506ms       remaining: 21.5s
23:     learn: 0.1206108      total: 524ms       remaining: 21.3s
24:     learn: 0.1201876      total: 549ms       remaining: 21.4s
25:     learn: 0.1197557      total: 567ms       remaining: 21.2s
26:     learn: 0.1194141      total: 585ms       remaining: 21.1s
27:     learn: 0.1191670      total: 603ms       remaining: 20.9s
28:     learn: 0.1188092      total: 621ms       remaining: 20.8s
29:     learn: 0.1183783      total: 638ms       remaining: 20.6s
30:     learn: 0.1179692      total: 655ms       remaining: 20.5s
31:     learn: 0.1174637      total: 674ms       remaining: 20.4s
32:     learn: 0.1172618      total: 695ms       remaining: 20.4s
33:     learn: 0.1170538      total: 714ms       remaining: 20.3s
34:     learn: 0.1168898      total: 731ms       remaining: 20.2s
35:     learn: 0.1166021      total: 749ms       remaining: 20s
36:     learn: 0.1160559      total: 766ms       remaining: 19.9s
37:     learn: 0.1159423      total: 785ms       remaining: 19.9s
38:     learn: 0.1157336      total: 806ms       remaining: 19.9s
39:     learn: 0.1156070      total: 831ms       remaining: 20s
40:     learn: 0.1153637      total: 850ms       remaining: 19.9s
41:     learn: 0.1150932      total: 869ms       remaining: 19.8s

```

42:	learn: 0.1149174	total: 887ms	remaining: 19.8s
43:	learn: 0.1148090	total: 909ms	remaining: 19.7s
44:	learn: 0.1147063	total: 927ms	remaining: 19.7s
45:	learn: 0.1145318	total: 945ms	remaining: 19.6s
46:	learn: 0.1144474	total: 963ms	remaining: 19.5s
47:	learn: 0.1141118	total: 982ms	remaining: 19.5s
48:	learn: 0.1140460	total: 1000ms	remaining: 19.4s
49:	learn: 0.1139883	total: 1.02s	remaining: 19.4s
50:	learn: 0.1138856	total: 1.04s	remaining: 19.3s
51:	learn: 0.1137421	total: 1.05s	remaining: 19.2s
52:	learn: 0.1136476	total: 1.07s	remaining: 19.2s
53:	learn: 0.1134945	total: 1.09s	remaining: 19.2s
54:	learn: 0.1134242	total: 1.11s	remaining: 19.1s
55:	learn: 0.1132415	total: 1.13s	remaining: 19.1s
56:	learn: 0.1131536	total: 1.15s	remaining: 19s
57:	learn: 0.1130285	total: 1.17s	remaining: 19s
58:	learn: 0.1129719	total: 1.19s	remaining: 18.9s
59:	learn: 0.1129193	total: 1.21s	remaining: 18.9s
60:	learn: 0.1128410	total: 1.23s	remaining: 18.9s
61:	learn: 0.1127790	total: 1.24s	remaining: 18.8s
62:	learn: 0.1125528	total: 1.27s	remaining: 18.9s
63:	learn: 0.1124814	total: 1.3s	remaining: 19s
64:	learn: 0.1124177	total: 1.32s	remaining: 18.9s
65:	learn: 0.1122175	total: 1.33s	remaining: 18.9s
66:	learn: 0.1120966	total: 1.35s	remaining: 18.8s
67:	learn: 0.1119020	total: 1.37s	remaining: 18.8s
68:	learn: 0.1118702	total: 1.39s	remaining: 18.8s
69:	learn: 0.1117841	total: 1.41s	remaining: 18.8s
70:	learn: 0.1117321	total: 1.43s	remaining: 18.7s
71:	learn: 0.1116288	total: 1.45s	remaining: 18.7s
72:	learn: 0.1114910	total: 1.46s	remaining: 18.6s
73:	learn: 0.1114658	total: 1.48s	remaining: 18.5s
74:	learn: 0.1114183	total: 1.51s	remaining: 18.6s
75:	learn: 0.1113989	total: 1.52s	remaining: 18.5s
76:	learn: 0.1113061	total: 1.54s	remaining: 18.5s
77:	learn: 0.1112112	total: 1.56s	remaining: 18.4s
78:	learn: 0.1111017	total: 1.57s	remaining: 18.4s
79:	learn: 0.1110659	total: 1.59s	remaining: 18.3s
80:	learn: 0.1109642	total: 1.61s	remaining: 18.3s
81:	learn: 0.1109050	total: 1.63s	remaining: 18.2s
82:	learn: 0.1108713	total: 1.65s	remaining: 18.2s
83:	learn: 0.1108441	total: 1.67s	remaining: 18.2s
84:	learn: 0.1107977	total: 1.68s	remaining: 18.1s
85:	learn: 0.1107419	total: 1.71s	remaining: 18.1s
86:	learn: 0.1106318	total: 1.73s	remaining: 18.1s
87:	learn: 0.1105669	total: 1.75s	remaining: 18.1s
88:	learn: 0.1105318	total: 1.77s	remaining: 18.1s
89:	learn: 0.1104666	total: 1.79s	remaining: 18.1s
90:	learn: 0.1104477	total: 1.8s	remaining: 18s
91:	learn: 0.1103997	total: 1.82s	remaining: 18s
92:	learn: 0.1103775	total: 1.84s	remaining: 18s
93:	learn: 0.1103361	total: 1.86s	remaining: 17.9s
94:	learn: 0.1102889	total: 1.88s	remaining: 17.9s
95:	learn: 0.1102423	total: 1.89s	remaining: 17.8s
96:	learn: 0.1101822	total: 1.92s	remaining: 17.8s
97:	learn: 0.1101612	total: 1.93s	remaining: 17.8s
98:	learn: 0.1101258	total: 1.95s	remaining: 17.8s
99:	learn: 0.1100985	total: 1.97s	remaining: 17.7s
100:	learn: 0.1099703	total: 1.99s	remaining: 17.7s
101:	learn: 0.1099374	total: 2.01s	remaining: 17.7s
102:	learn: 0.1098844	total: 2.03s	remaining: 17.7s
103:	learn: 0.1098451	total: 2.05s	remaining: 17.6s
104:	learn: 0.1098186	total: 2.06s	remaining: 17.6s
105:	learn: 0.1097983	total: 2.08s	remaining: 17.6s

106:	learn: 0.1097456	total: 2.1s	remaining: 17.5s
107:	learn: 0.1096487	total: 2.12s	remaining: 17.5s
108:	learn: 0.1096311	total: 2.14s	remaining: 17.5s
109:	learn: 0.1095979	total: 2.16s	remaining: 17.5s
110:	learn: 0.1095538	total: 2.18s	remaining: 17.5s
111:	learn: 0.1095335	total: 2.2s	remaining: 17.4s
112:	learn: 0.1094518	total: 2.22s	remaining: 17.4s
113:	learn: 0.1094348	total: 2.23s	remaining: 17.4s
114:	learn: 0.1093853	total: 2.27s	remaining: 17.5s
115:	learn: 0.1093606	total: 2.29s	remaining: 17.5s
116:	learn: 0.1092823	total: 2.31s	remaining: 17.4s
117:	learn: 0.1092647	total: 2.33s	remaining: 17.4s
118:	learn: 0.1091999	total: 2.35s	remaining: 17.4s
119:	learn: 0.1091445	total: 2.37s	remaining: 17.4s
120:	learn: 0.1091060	total: 2.38s	remaining: 17.3s
121:	learn: 0.1090856	total: 2.4s	remaining: 17.3s
122:	learn: 0.1090717	total: 2.42s	remaining: 17.3s
123:	learn: 0.1090373	total: 2.44s	remaining: 17.2s
124:	learn: 0.1090168	total: 2.46s	remaining: 17.2s
125:	learn: 0.1089880	total: 2.47s	remaining: 17.2s
126:	learn: 0.1089628	total: 2.49s	remaining: 17.1s
127:	learn: 0.1089328	total: 2.51s	remaining: 17.1s
128:	learn: 0.1088861	total: 2.53s	remaining: 17.1s
129:	learn: 0.1088346	total: 2.56s	remaining: 17.1s
130:	learn: 0.1087732	total: 2.57s	remaining: 17.1s
131:	learn: 0.1087440	total: 2.59s	remaining: 17s
132:	learn: 0.1087246	total: 2.61s	remaining: 17s
133:	learn: 0.1086759	total: 2.63s	remaining: 17s
134:	learn: 0.1086357	total: 2.65s	remaining: 17s
135:	learn: 0.1086165	total: 2.67s	remaining: 16.9s
136:	learn: 0.1085872	total: 2.68s	remaining: 16.9s
137:	learn: 0.1085635	total: 2.7s	remaining: 16.9s
138:	learn: 0.1085254	total: 2.72s	remaining: 16.9s
139:	learn: 0.1085133	total: 2.75s	remaining: 16.9s
140:	learn: 0.1084962	total: 2.76s	remaining: 16.8s
141:	learn: 0.1084575	total: 2.78s	remaining: 16.8s
142:	learn: 0.1083956	total: 2.8s	remaining: 16.8s
143:	learn: 0.1083619	total: 2.82s	remaining: 16.7s
144:	learn: 0.1083215	total: 2.83s	remaining: 16.7s
145:	learn: 0.1082836	total: 2.85s	remaining: 16.7s
146:	learn: 0.1082560	total: 2.88s	remaining: 16.7s
147:	learn: 0.1082351	total: 2.89s	remaining: 16.7s
148:	learn: 0.1082054	total: 2.92s	remaining: 16.7s
149:	learn: 0.1081776	total: 2.94s	remaining: 16.6s
150:	learn: 0.1081538	total: 2.96s	remaining: 16.6s
151:	learn: 0.1081435	total: 2.97s	remaining: 16.6s
152:	learn: 0.1080835	total: 2.99s	remaining: 16.6s
153:	learn: 0.1080488	total: 3.01s	remaining: 16.5s
154:	learn: 0.1080403	total: 3.03s	remaining: 16.5s
155:	learn: 0.1080107	total: 3.04s	remaining: 16.5s
156:	learn: 0.1079822	total: 3.06s	remaining: 16.4s
157:	learn: 0.1079599	total: 3.08s	remaining: 16.4s
158:	learn: 0.1079372	total: 3.1s	remaining: 16.4s
159:	learn: 0.1079251	total: 3.12s	remaining: 16.4s
160:	learn: 0.1078923	total: 3.14s	remaining: 16.4s
161:	learn: 0.1078686	total: 3.16s	remaining: 16.4s
162:	learn: 0.1078402	total: 3.18s	remaining: 16.3s
163:	learn: 0.1078052	total: 3.2s	remaining: 16.3s
164:	learn: 0.1077873	total: 3.22s	remaining: 16.3s
165:	learn: 0.1077571	total: 3.25s	remaining: 16.3s
166:	learn: 0.1077313	total: 3.27s	remaining: 16.3s
167:	learn: 0.1077078	total: 3.29s	remaining: 16.3s
168:	learn: 0.1076736	total: 3.31s	remaining: 16.3s
169:	learn: 0.1076475	total: 3.33s	remaining: 16.2s

170:	learn: 0.1076311	total: 3.35s	remaining: 16.2s
171:	learn: 0.1076014	total: 3.36s	remaining: 16.2s
172:	learn: 0.1075838	total: 3.38s	remaining: 16.2s
173:	learn: 0.1075685	total: 3.4s	remaining: 16.1s
174:	learn: 0.1075549	total: 3.42s	remaining: 16.1s
175:	learn: 0.1075307	total: 3.43s	remaining: 16.1s
176:	learn: 0.1075175	total: 3.45s	remaining: 16.1s
177:	learn: 0.1075020	total: 3.47s	remaining: 16s
178:	learn: 0.1074918	total: 3.49s	remaining: 16s
179:	learn: 0.1074800	total: 3.51s	remaining: 16s
180:	learn: 0.1074681	total: 3.52s	remaining: 15.9s
181:	learn: 0.1074462	total: 3.54s	remaining: 15.9s
182:	learn: 0.1074405	total: 3.56s	remaining: 15.9s
183:	learn: 0.1074192	total: 3.58s	remaining: 15.9s
184:	learn: 0.1073949	total: 3.6s	remaining: 15.9s
185:	learn: 0.1073647	total: 3.62s	remaining: 15.8s
186:	learn: 0.1073554	total: 3.64s	remaining: 15.8s
187:	learn: 0.1073458	total: 3.65s	remaining: 15.8s
188:	learn: 0.1073352	total: 3.67s	remaining: 15.8s
189:	learn: 0.1073102	total: 3.69s	remaining: 15.7s
190:	learn: 0.1073037	total: 3.71s	remaining: 15.7s
191:	learn: 0.1072779	total: 3.73s	remaining: 15.7s
192:	learn: 0.1072616	total: 3.75s	remaining: 15.7s
193:	learn: 0.1072333	total: 3.77s	remaining: 15.7s
194:	learn: 0.1072260	total: 3.79s	remaining: 15.6s
195:	learn: 0.1072101	total: 3.81s	remaining: 15.6s
196:	learn: 0.1071845	total: 3.83s	remaining: 15.6s
197:	learn: 0.1071747	total: 3.84s	remaining: 15.6s
198:	learn: 0.1071692	total: 3.86s	remaining: 15.5s
199:	learn: 0.1071484	total: 3.88s	remaining: 15.5s
200:	learn: 0.1071152	total: 3.9s	remaining: 15.5s
201:	learn: 0.1070950	total: 3.92s	remaining: 15.5s
202:	learn: 0.1070818	total: 3.93s	remaining: 15.4s
203:	learn: 0.1070452	total: 3.96s	remaining: 15.4s
204:	learn: 0.1070157	total: 3.98s	remaining: 15.4s
205:	learn: 0.1070043	total: 3.99s	remaining: 15.4s
206:	learn: 0.1069704	total: 4.01s	remaining: 15.4s
207:	learn: 0.1069452	total: 4.03s	remaining: 15.3s
208:	learn: 0.1069290	total: 4.05s	remaining: 15.3s
209:	learn: 0.1069138	total: 4.07s	remaining: 15.3s
210:	learn: 0.1068792	total: 4.09s	remaining: 15.3s
211:	learn: 0.1068603	total: 4.11s	remaining: 15.3s
212:	learn: 0.1068507	total: 4.13s	remaining: 15.2s
213:	learn: 0.1068302	total: 4.15s	remaining: 15.2s
214:	learn: 0.1068007	total: 4.17s	remaining: 15.2s
215:	learn: 0.1067943	total: 4.18s	remaining: 15.2s
216:	learn: 0.1067758	total: 4.22s	remaining: 15.2s
217:	learn: 0.1067539	total: 4.24s	remaining: 15.2s
218:	learn: 0.1067320	total: 4.26s	remaining: 15.2s
219:	learn: 0.1067200	total: 4.28s	remaining: 15.2s
220:	learn: 0.1067073	total: 4.3s	remaining: 15.2s
221:	learn: 0.1066902	total: 4.32s	remaining: 15.1s
222:	learn: 0.1066620	total: 4.33s	remaining: 15.1s
223:	learn: 0.1066438	total: 4.35s	remaining: 15.1s
224:	learn: 0.1066270	total: 4.37s	remaining: 15.1s
225:	learn: 0.1066195	total: 4.39s	remaining: 15s
226:	learn: 0.1065977	total: 4.41s	remaining: 15s
227:	learn: 0.1065848	total: 4.43s	remaining: 15s
228:	learn: 0.1065786	total: 4.45s	remaining: 15s
229:	learn: 0.1065592	total: 4.47s	remaining: 15s
230:	learn: 0.1065524	total: 4.49s	remaining: 14.9s
231:	learn: 0.1065376	total: 4.51s	remaining: 14.9s
232:	learn: 0.1065211	total: 4.53s	remaining: 14.9s
233:	learn: 0.1065130	total: 4.55s	remaining: 14.9s



234:	learn: 0.1065019	total: 4.57s	remaining: 14.9s
235:	learn: 0.1064916	total: 4.59s	remaining: 14.9s
236:	learn: 0.1064746	total: 4.62s	remaining: 14.9s
237:	learn: 0.1064567	total: 4.65s	remaining: 14.9s
238:	learn: 0.1064428	total: 4.67s	remaining: 14.9s
239:	learn: 0.1064317	total: 4.68s	remaining: 14.8s
240:	learn: 0.1064150	total: 4.7s	remaining: 14.8s
241:	learn: 0.1063893	total: 4.72s	remaining: 14.8s
242:	learn: 0.1063829	total: 4.74s	remaining: 14.8s
243:	learn: 0.1063743	total: 4.76s	remaining: 14.7s
244:	learn: 0.1063533	total: 4.78s	remaining: 14.7s
245:	learn: 0.1063398	total: 4.8s	remaining: 14.7s
246:	learn: 0.1063272	total: 4.82s	remaining: 14.7s
247:	learn: 0.1063176	total: 4.84s	remaining: 14.7s
248:	learn: 0.1062991	total: 4.85s	remaining: 14.6s
249:	learn: 0.1062852	total: 4.87s	remaining: 14.6s
250:	learn: 0.1062701	total: 4.89s	remaining: 14.6s
251:	learn: 0.1062617	total: 4.91s	remaining: 14.6s
252:	learn: 0.1062511	total: 4.92s	remaining: 14.5s
253:	learn: 0.1062179	total: 4.94s	remaining: 14.5s
254:	learn: 0.1062103	total: 4.96s	remaining: 14.5s
255:	learn: 0.1062003	total: 4.98s	remaining: 14.5s
256:	learn: 0.1061801	total: 5s	remaining: 14.5s
257:	learn: 0.1061704	total: 5.02s	remaining: 14.4s
258:	learn: 0.1061570	total: 5.04s	remaining: 14.4s
259:	learn: 0.1061508	total: 5.06s	remaining: 14.4s
260:	learn: 0.1061324	total: 5.08s	remaining: 14.4s
261:	learn: 0.1061228	total: 5.09s	remaining: 14.3s
262:	learn: 0.1061165	total: 5.11s	remaining: 14.3s
263:	learn: 0.1061121	total: 5.13s	remaining: 14.3s
264:	learn: 0.1061014	total: 5.15s	remaining: 14.3s
265:	learn: 0.1060789	total: 5.18s	remaining: 14.3s
266:	learn: 0.1060691	total: 5.21s	remaining: 14.3s
267:	learn: 0.1060587	total: 5.23s	remaining: 14.3s
268:	learn: 0.1060476	total: 5.24s	remaining: 14.2s
269:	learn: 0.1060321	total: 5.26s	remaining: 14.2s
270:	learn: 0.1060251	total: 5.28s	remaining: 14.2s
271:	learn: 0.1060185	total: 5.3s	remaining: 14.2s
272:	learn: 0.1060032	total: 5.31s	remaining: 14.2s
273:	learn: 0.1059954	total: 5.33s	remaining: 14.1s
274:	learn: 0.1059805	total: 5.35s	remaining: 14.1s
275:	learn: 0.1059603	total: 5.37s	remaining: 14.1s
276:	learn: 0.1059436	total: 5.39s	remaining: 14.1s
277:	learn: 0.1059173	total: 5.41s	remaining: 14.1s
278:	learn: 0.1059008	total: 5.43s	remaining: 14s
279:	learn: 0.1058902	total: 5.45s	remaining: 14s
280:	learn: 0.1058843	total: 5.46s	remaining: 14s
281:	learn: 0.1058703	total: 5.48s	remaining: 14s
282:	learn: 0.1058622	total: 5.5s	remaining: 13.9s
283:	learn: 0.1058593	total: 5.52s	remaining: 13.9s
284:	learn: 0.1058431	total: 5.54s	remaining: 13.9s
285:	learn: 0.1058353	total: 5.55s	remaining: 13.9s
286:	learn: 0.1058252	total: 5.57s	remaining: 13.8s
287:	learn: 0.1058141	total: 5.59s	remaining: 13.8s
288:	learn: 0.1058003	total: 5.61s	remaining: 13.8s
289:	learn: 0.1057892	total: 5.63s	remaining: 13.8s
290:	learn: 0.1057743	total: 5.65s	remaining: 13.8s
291:	learn: 0.1057637	total: 5.67s	remaining: 13.7s
292:	learn: 0.1057572	total: 5.68s	remaining: 13.7s
293:	learn: 0.1057401	total: 5.7s	remaining: 13.7s
294:	learn: 0.1057344	total: 5.72s	remaining: 13.7s
295:	learn: 0.1057216	total: 5.75s	remaining: 13.7s
296:	learn: 0.1057076	total: 5.77s	remaining: 13.7s
297:	learn: 0.1057007	total: 5.79s	remaining: 13.6s

298:	learn: 0.1056915	total: 5.81s	remaining: 13.6s
299:	learn: 0.1056795	total: 5.83s	remaining: 13.6s
300:	learn: 0.1056725	total: 5.85s	remaining: 13.6s
301:	learn: 0.1056572	total: 5.87s	remaining: 13.6s
302:	learn: 0.1056454	total: 5.88s	remaining: 13.5s
303:	learn: 0.1056373	total: 5.9s	remaining: 13.5s
304:	learn: 0.1056295	total: 5.92s	remaining: 13.5s
305:	learn: 0.1056191	total: 5.94s	remaining: 13.5s
306:	learn: 0.1056063	total: 5.96s	remaining: 13.4s
307:	learn: 0.1055919	total: 5.98s	remaining: 13.4s
308:	learn: 0.1055854	total: 6s	remaining: 13.4s
309:	learn: 0.1055763	total: 6.02s	remaining: 13.4s
310:	learn: 0.1055677	total: 6.03s	remaining: 13.4s
311:	learn: 0.1055552	total: 6.05s	remaining: 13.4s
312:	learn: 0.1055389	total: 6.07s	remaining: 13.3s
313:	learn: 0.1055273	total: 6.09s	remaining: 13.3s
314:	learn: 0.1055178	total: 6.11s	remaining: 13.3s
315:	learn: 0.1054980	total: 6.13s	remaining: 13.3s
316:	learn: 0.1054937	total: 6.15s	remaining: 13.3s
317:	learn: 0.1054889	total: 6.18s	remaining: 13.2s
318:	learn: 0.1054803	total: 6.2s	remaining: 13.2s
319:	learn: 0.1054733	total: 6.22s	remaining: 13.2s
320:	learn: 0.1054629	total: 6.24s	remaining: 13.2s
321:	learn: 0.1054561	total: 6.25s	remaining: 13.2s
322:	learn: 0.1054419	total: 6.27s	remaining: 13.1s
323:	learn: 0.1054337	total: 6.29s	remaining: 13.1s
324:	learn: 0.1054273	total: 6.31s	remaining: 13.1s
325:	learn: 0.1054206	total: 6.33s	remaining: 13.1s
326:	learn: 0.1054121	total: 6.35s	remaining: 13.1s
327:	learn: 0.1054026	total: 6.37s	remaining: 13s
328:	learn: 0.1053902	total: 6.38s	remaining: 13s
329:	learn: 0.1053820	total: 6.41s	remaining: 13s
330:	learn: 0.1053763	total: 6.42s	remaining: 13s
331:	learn: 0.1053683	total: 6.44s	remaining: 13s
332:	learn: 0.1053572	total: 6.46s	remaining: 12.9s
333:	learn: 0.1053484	total: 6.48s	remaining: 12.9s
334:	learn: 0.1053360	total: 6.5s	remaining: 12.9s
335:	learn: 0.1053281	total: 6.52s	remaining: 12.9s
336:	learn: 0.1053151	total: 6.54s	remaining: 12.9s
337:	learn: 0.1053078	total: 6.55s	remaining: 12.8s
338:	learn: 0.1052966	total: 6.57s	remaining: 12.8s
339:	learn: 0.1052881	total: 6.59s	remaining: 12.8s
340:	learn: 0.1052822	total: 6.61s	remaining: 12.8s
341:	learn: 0.1052762	total: 6.63s	remaining: 12.8s
342:	learn: 0.1052715	total: 6.65s	remaining: 12.7s
343:	learn: 0.1052649	total: 6.67s	remaining: 12.7s
344:	learn: 0.1052600	total: 6.68s	remaining: 12.7s
345:	learn: 0.1052539	total: 6.71s	remaining: 12.7s
346:	learn: 0.1052500	total: 6.73s	remaining: 12.7s
347:	learn: 0.1052464	total: 6.74s	remaining: 12.6s
348:	learn: 0.1052379	total: 6.76s	remaining: 12.6s
349:	learn: 0.1052272	total: 6.78s	remaining: 12.6s
350:	learn: 0.1052199	total: 6.8s	remaining: 12.6s
351:	learn: 0.1052147	total: 6.82s	remaining: 12.6s
352:	learn: 0.1052047	total: 6.84s	remaining: 12.5s
353:	learn: 0.1051935	total: 6.86s	remaining: 12.5s
354:	learn: 0.1051906	total: 6.88s	remaining: 12.5s
355:	learn: 0.1051843	total: 6.9s	remaining: 12.5s
356:	learn: 0.1051800	total: 6.93s	remaining: 12.5s
357:	learn: 0.1051722	total: 6.96s	remaining: 12.5s
358:	learn: 0.1051683	total: 6.99s	remaining: 12.5s
359:	learn: 0.1051617	total: 7.03s	remaining: 12.5s
360:	learn: 0.1051552	total: 7.07s	remaining: 12.5s
361:	learn: 0.1051508	total: 7.1s	remaining: 12.5s

362:	learn: 0.1051471	total: 7.15s	remaining: 12.5s
363:	learn: 0.1051430	total: 7.18s	remaining: 12.5s
364:	learn: 0.1051385	total: 7.21s	remaining: 12.5s
365:	learn: 0.1051253	total: 7.24s	remaining: 12.5s
366:	learn: 0.1051203	total: 7.27s	remaining: 12.5s
367:	learn: 0.1051175	total: 7.32s	remaining: 12.6s
368:	learn: 0.1051049	total: 7.35s	remaining: 12.6s
369:	learn: 0.1050999	total: 7.4s	remaining: 12.6s
370:	learn: 0.1050930	total: 7.45s	remaining: 12.6s
371:	learn: 0.1050874	total: 7.5s	remaining: 12.7s
372:	learn: 0.1050686	total: 7.55s	remaining: 12.7s
373:	learn: 0.1050609	total: 7.6s	remaining: 12.7s
374:	learn: 0.1050571	total: 7.64s	remaining: 12.7s
375:	learn: 0.1050475	total: 7.7s	remaining: 12.8s
376:	learn: 0.1050358	total: 7.72s	remaining: 12.8s
377:	learn: 0.1050311	total: 7.75s	remaining: 12.8s
378:	learn: 0.1050250	total: 7.78s	remaining: 12.7s
379:	learn: 0.1050206	total: 7.8s	remaining: 12.7s
380:	learn: 0.1050091	total: 7.84s	remaining: 12.7s
381:	learn: 0.1050016	total: 7.89s	remaining: 12.8s
382:	learn: 0.1049966	total: 7.94s	remaining: 12.8s
383:	learn: 0.1049878	total: 7.98s	remaining: 12.8s
384:	learn: 0.1049751	total: 8.03s	remaining: 12.8s
385:	learn: 0.1049699	total: 8.07s	remaining: 12.8s
386:	learn: 0.1049641	total: 8.12s	remaining: 12.9s
387:	learn: 0.1049514	total: 8.17s	remaining: 12.9s
388:	learn: 0.1049454	total: 8.22s	remaining: 12.9s
389:	learn: 0.1049318	total: 8.26s	remaining: 12.9s
390:	learn: 0.1049185	total: 8.28s	remaining: 12.9s
391:	learn: 0.1049130	total: 8.32s	remaining: 12.9s
392:	learn: 0.1049021	total: 8.34s	remaining: 12.9s
393:	learn: 0.1048911	total: 8.38s	remaining: 12.9s
394:	learn: 0.1048820	total: 8.41s	remaining: 12.9s
395:	learn: 0.1048704	total: 8.45s	remaining: 12.9s
396:	learn: 0.1048648	total: 8.5s	remaining: 12.9s
397:	learn: 0.1048550	total: 8.55s	remaining: 12.9s
398:	learn: 0.1048456	total: 8.6s	remaining: 13s
399:	learn: 0.1048282	total: 8.64s	remaining: 13s
400:	learn: 0.1048242	total: 8.68s	remaining: 13s
401:	learn: 0.1048212	total: 8.72s	remaining: 13s
402:	learn: 0.1048144	total: 8.78s	remaining: 13s
403:	learn: 0.1048093	total: 8.82s	remaining: 13s
404:	learn: 0.1047993	total: 8.87s	remaining: 13s
405:	learn: 0.1047915	total: 8.92s	remaining: 13s
406:	learn: 0.1047859	total: 8.96s	remaining: 13.1s
407:	learn: 0.1047823	total: 9.02s	remaining: 13.1s
408:	learn: 0.1047757	total: 9.06s	remaining: 13.1s
409:	learn: 0.1047574	total: 9.09s	remaining: 13.1s
410:	learn: 0.1047510	total: 9.14s	remaining: 13.1s
411:	learn: 0.1047469	total: 9.18s	remaining: 13.1s
412:	learn: 0.1047435	total: 9.23s	remaining: 13.1s
413:	learn: 0.1047324	total: 9.27s	remaining: 13.1s
414:	learn: 0.1047269	total: 9.32s	remaining: 13.1s
415:	learn: 0.1047223	total: 9.37s	remaining: 13.2s
416:	learn: 0.1047189	total: 9.41s	remaining: 13.2s
417:	learn: 0.1047094	total: 9.47s	remaining: 13.2s
418:	learn: 0.1047009	total: 9.52s	remaining: 13.2s
419:	learn: 0.1046903	total: 9.56s	remaining: 13.2s
420:	learn: 0.1046817	total: 9.6s	remaining: 13.2s
421:	learn: 0.1046792	total: 9.63s	remaining: 13.2s
422:	learn: 0.1046743	total: 9.67s	remaining: 13.2s
423:	learn: 0.1046693	total: 9.72s	remaining: 13.2s
424:	learn: 0.1046643	total: 9.76s	remaining: 13.2s
425:	learn: 0.1046557	total: 9.79s	remaining: 13.2s

426:	learn: 0.1046518	total: 9.82s	remaining: 13.2s
427:	learn: 0.1046388	total: 9.86s	remaining: 13.2s
428:	learn: 0.1046336	total: 9.89s	remaining: 13.2s
429:	learn: 0.1046299	total: 9.94s	remaining: 13.2s
430:	learn: 0.1046195	total: 9.98s	remaining: 13.2s
431:	learn: 0.1046132	total: 10s	remaining: 13.2s
432:	learn: 0.1046063	total: 10.1s	remaining: 13.2s
433:	learn: 0.1045982	total: 10.1s	remaining: 13.2s
434:	learn: 0.1045891	total: 10.1s	remaining: 13.2s
435:	learn: 0.1045841	total: 10.2s	remaining: 13.2s
436:	learn: 0.1045738	total: 10.2s	remaining: 13.1s
437:	learn: 0.1045695	total: 10.3s	remaining: 13.2s
438:	learn: 0.1045643	total: 10.3s	remaining: 13.2s
439:	learn: 0.1045602	total: 10.3s	remaining: 13.2s
440:	learn: 0.1045570	total: 10.4s	remaining: 13.2s
441:	learn: 0.1045489	total: 10.4s	remaining: 13.2s
442:	learn: 0.1045415	total: 10.5s	remaining: 13.2s
443:	learn: 0.1045330	total: 10.5s	remaining: 13.2s
444:	learn: 0.1045262	total: 10.5s	remaining: 13.1s
445:	learn: 0.1045243	total: 10.6s	remaining: 13.1s
446:	learn: 0.1045181	total: 10.6s	remaining: 13.1s
447:	learn: 0.1045137	total: 10.6s	remaining: 13.1s
448:	learn: 0.1045085	total: 10.6s	remaining: 13s
449:	learn: 0.1045047	total: 10.6s	remaining: 13s
450:	learn: 0.1044915	total: 10.7s	remaining: 13s
451:	learn: 0.1044856	total: 10.7s	remaining: 12.9s
452:	learn: 0.1044770	total: 10.7s	remaining: 12.9s
453:	learn: 0.1044732	total: 10.7s	remaining: 12.9s
454:	learn: 0.1044633	total: 10.7s	remaining: 12.8s
455:	learn: 0.1044545	total: 10.8s	remaining: 12.8s
456:	learn: 0.1044489	total: 10.8s	remaining: 12.8s
457:	learn: 0.1044466	total: 10.8s	remaining: 12.8s
458:	learn: 0.1044426	total: 10.8s	remaining: 12.7s
459:	learn: 0.1044358	total: 10.8s	remaining: 12.7s
460:	learn: 0.1044241	total: 10.8s	remaining: 12.7s
461:	learn: 0.1044214	total: 10.9s	remaining: 12.7s
462:	learn: 0.1044186	total: 10.9s	remaining: 12.6s
463:	learn: 0.1044153	total: 10.9s	remaining: 12.6s
464:	learn: 0.1044103	total: 10.9s	remaining: 12.6s
465:	learn: 0.1044060	total: 10.9s	remaining: 12.5s
466:	learn: 0.1044008	total: 11s	remaining: 12.5s
467:	learn: 0.1043868	total: 11s	remaining: 12.5s
468:	learn: 0.1043795	total: 11s	remaining: 12.5s
469:	learn: 0.1043731	total: 11s	remaining: 12.4s
470:	learn: 0.1043638	total: 11s	remaining: 12.4s
471:	learn: 0.1043580	total: 11.1s	remaining: 12.4s
472:	learn: 0.1043505	total: 11.1s	remaining: 12.3s
473:	learn: 0.1043452	total: 11.1s	remaining: 12.3s
474:	learn: 0.1043369	total: 11.1s	remaining: 12.3s
475:	learn: 0.1043329	total: 11.1s	remaining: 12.2s
476:	learn: 0.1043303	total: 11.1s	remaining: 12.2s
477:	learn: 0.1043250	total: 11.2s	remaining: 12.2s
478:	learn: 0.1043218	total: 11.2s	remaining: 12.2s
479:	learn: 0.1043131	total: 11.2s	remaining: 12.2s
480:	learn: 0.1043110	total: 11.2s	remaining: 12.1s
481:	learn: 0.1043066	total: 11.3s	remaining: 12.1s
482:	learn: 0.1043037	total: 11.3s	remaining: 12.1s
483:	learn: 0.1043010	total: 11.3s	remaining: 12s
484:	learn: 0.1042993	total: 11.3s	remaining: 12s
485:	learn: 0.1042909	total: 11.3s	remaining: 12s
486:	learn: 0.1042848	total: 11.3s	remaining: 12s
487:	learn: 0.1042822	total: 11.4s	remaining: 11.9s
488:	learn: 0.1042777	total: 11.4s	remaining: 11.9s
489:	learn: 0.1042733	total: 11.4s	remaining: 11.9s

490:	learn: 0.1042698	total: 11.4s	remaining: 11.8s
491:	learn: 0.1042663	total: 11.4s	remaining: 11.8s
492:	learn: 0.1042585	total: 11.5s	remaining: 11.8s
493:	learn: 0.1042548	total: 11.5s	remaining: 11.8s
494:	learn: 0.1042486	total: 11.5s	remaining: 11.7s
495:	learn: 0.1042450	total: 11.5s	remaining: 11.7s
496:	learn: 0.1042432	total: 11.5s	remaining: 11.7s
497:	learn: 0.1042409	total: 11.6s	remaining: 11.6s
498:	learn: 0.1042367	total: 11.6s	remaining: 11.6s
499:	learn: 0.1042334	total: 11.6s	remaining: 11.6s
500:	learn: 0.1042253	total: 11.6s	remaining: 11.6s
501:	learn: 0.1042164	total: 11.6s	remaining: 11.5s
502:	learn: 0.1042132	total: 11.6s	remaining: 11.5s
503:	learn: 0.1042053	total: 11.7s	remaining: 11.5s
504:	learn: 0.1041988	total: 11.7s	remaining: 11.5s
505:	learn: 0.1041936	total: 11.7s	remaining: 11.4s
506:	learn: 0.1041897	total: 11.7s	remaining: 11.4s
507:	learn: 0.1041817	total: 11.7s	remaining: 11.4s
508:	learn: 0.1041784	total: 11.8s	remaining: 11.3s
509:	learn: 0.1041716	total: 11.8s	remaining: 11.3s
510:	learn: 0.1041668	total: 11.8s	remaining: 11.3s
511:	learn: 0.1041577	total: 11.8s	remaining: 11.3s
512:	learn: 0.1041544	total: 11.8s	remaining: 11.2s
513:	learn: 0.1041466	total: 11.9s	remaining: 11.2s
514:	learn: 0.1041433	total: 11.9s	remaining: 11.2s
515:	learn: 0.1041369	total: 11.9s	remaining: 11.2s
516:	learn: 0.1041330	total: 11.9s	remaining: 11.1s
517:	learn: 0.1041285	total: 11.9s	remaining: 11.1s
518:	learn: 0.1041226	total: 11.9s	remaining: 11.1s
519:	learn: 0.1041190	total: 12s	remaining: 11.1s
520:	learn: 0.1041121	total: 12s	remaining: 11s
521:	learn: 0.1041101	total: 12s	remaining: 11s
522:	learn: 0.1041069	total: 12s	remaining: 11s
523:	learn: 0.1041042	total: 12.1s	remaining: 10.9s
524:	learn: 0.1040942	total: 12.1s	remaining: 10.9s
525:	learn: 0.1040914	total: 12.1s	remaining: 10.9s
526:	learn: 0.1040798	total: 12.1s	remaining: 10.9s
527:	learn: 0.1040785	total: 12.1s	remaining: 10.8s
528:	learn: 0.1040748	total: 12.2s	remaining: 10.8s
529:	learn: 0.1040644	total: 12.2s	remaining: 10.8s
530:	learn: 0.1040605	total: 12.2s	remaining: 10.8s
531:	learn: 0.1040569	total: 12.2s	remaining: 10.8s
532:	learn: 0.1040516	total: 12.2s	remaining: 10.7s
533:	learn: 0.1040449	total: 12.3s	remaining: 10.7s
534:	learn: 0.1040358	total: 12.3s	remaining: 10.7s
535:	learn: 0.1040312	total: 12.3s	remaining: 10.6s
536:	learn: 0.1040280	total: 12.3s	remaining: 10.6s
537:	learn: 0.1040242	total: 12.3s	remaining: 10.6s
538:	learn: 0.1040181	total: 12.4s	remaining: 10.6s
539:	learn: 0.1040102	total: 12.4s	remaining: 10.5s
540:	learn: 0.1040047	total: 12.4s	remaining: 10.5s
541:	learn: 0.1040024	total: 12.4s	remaining: 10.5s
542:	learn: 0.1039968	total: 12.4s	remaining: 10.5s
543:	learn: 0.1039923	total: 12.5s	remaining: 10.4s
544:	learn: 0.1039833	total: 12.5s	remaining: 10.4s
545:	learn: 0.1039771	total: 12.5s	remaining: 10.4s
546:	learn: 0.1039723	total: 12.5s	remaining: 10.4s
547:	learn: 0.1039684	total: 12.5s	remaining: 10.3s
548:	learn: 0.1039652	total: 12.5s	remaining: 10.3s
549:	learn: 0.1039584	total: 12.6s	remaining: 10.3s
550:	learn: 0.1039529	total: 12.6s	remaining: 10.3s
551:	learn: 0.1039473	total: 12.6s	remaining: 10.2s
552:	learn: 0.1039415	total: 12.6s	remaining: 10.2s
553:	learn: 0.1039347	total: 12.6s	remaining: 10.2s

554:	learn: 0.1039304	total: 12.7s	remaining: 10.2s
555:	learn: 0.1039285	total: 12.7s	remaining: 10.1s
556:	learn: 0.1039252	total: 12.7s	remaining: 10.1s
557:	learn: 0.1039223	total: 12.7s	remaining: 10.1s
558:	learn: 0.1039197	total: 12.7s	remaining: 10s
559:	learn: 0.1039126	total: 12.7s	remaining: 10s
560:	learn: 0.1039045	total: 12.8s	remaining: 9.99s
561:	learn: 0.1039020	total: 12.8s	remaining: 9.96s
562:	learn: 0.1038976	total: 12.8s	remaining: 9.94s
563:	learn: 0.1038920	total: 12.8s	remaining: 9.92s
564:	learn: 0.1038888	total: 12.8s	remaining: 9.89s
565:	learn: 0.1038848	total: 12.9s	remaining: 9.86s
566:	learn: 0.1038818	total: 12.9s	remaining: 9.84s
567:	learn: 0.1038793	total: 12.9s	remaining: 9.81s
568:	learn: 0.1038721	total: 12.9s	remaining: 9.79s
569:	learn: 0.1038640	total: 12.9s	remaining: 9.76s
570:	learn: 0.1038615	total: 13s	remaining: 9.73s
571:	learn: 0.1038566	total: 13s	remaining: 9.71s
572:	learn: 0.1038539	total: 13s	remaining: 9.68s
573:	learn: 0.1038501	total: 13s	remaining: 9.65s
574:	learn: 0.1038471	total: 13s	remaining: 9.63s
575:	learn: 0.1038412	total: 13.1s	remaining: 9.61s
576:	learn: 0.1038390	total: 13.1s	remaining: 9.58s
577:	learn: 0.1038343	total: 13.1s	remaining: 9.55s
578:	learn: 0.1038319	total: 13.1s	remaining: 9.54s
579:	learn: 0.1038295	total: 13.1s	remaining: 9.52s
580:	learn: 0.1038233	total: 13.2s	remaining: 9.49s
581:	learn: 0.1038168	total: 13.2s	remaining: 9.47s
582:	learn: 0.1038097	total: 13.2s	remaining: 9.44s
583:	learn: 0.1038035	total: 13.2s	remaining: 9.42s
584:	learn: 0.1038014	total: 13.2s	remaining: 9.4s
585:	learn: 0.1037987	total: 13.3s	remaining: 9.37s
586:	learn: 0.1037900	total: 13.3s	remaining: 9.34s
587:	learn: 0.1037868	total: 13.3s	remaining: 9.32s
588:	learn: 0.1037825	total: 13.3s	remaining: 9.29s
589:	learn: 0.1037744	total: 13.3s	remaining: 9.27s
590:	learn: 0.1037710	total: 13.4s	remaining: 9.24s
591:	learn: 0.1037686	total: 13.4s	remaining: 9.21s
592:	learn: 0.1037664	total: 13.4s	remaining: 9.19s
593:	learn: 0.1037631	total: 13.4s	remaining: 9.16s
594:	learn: 0.1037580	total: 13.4s	remaining: 9.14s
595:	learn: 0.1037550	total: 13.4s	remaining: 9.11s
596:	learn: 0.1037528	total: 13.5s	remaining: 9.09s
597:	learn: 0.1037473	total: 13.5s	remaining: 9.06s
598:	learn: 0.1037330	total: 13.5s	remaining: 9.04s
599:	learn: 0.1037314	total: 13.5s	remaining: 9.01s
600:	learn: 0.1037263	total: 13.5s	remaining: 8.99s
601:	learn: 0.1037199	total: 13.6s	remaining: 8.97s
602:	learn: 0.1037160	total: 13.6s	remaining: 8.94s
603:	learn: 0.1037115	total: 13.6s	remaining: 8.91s
604:	learn: 0.1037068	total: 13.6s	remaining: 8.89s
605:	learn: 0.1037037	total: 13.6s	remaining: 8.86s
606:	learn: 0.1036918	total: 13.7s	remaining: 8.84s
607:	learn: 0.1036896	total: 13.7s	remaining: 8.81s
608:	learn: 0.1036849	total: 13.7s	remaining: 8.79s
609:	learn: 0.1036801	total: 13.7s	remaining: 8.76s
610:	learn: 0.1036766	total: 13.7s	remaining: 8.74s
611:	learn: 0.1036739	total: 13.7s	remaining: 8.71s
612:	learn: 0.1036708	total: 13.8s	remaining: 8.69s
613:	learn: 0.1036682	total: 13.8s	remaining: 8.66s
614:	learn: 0.1036652	total: 13.8s	remaining: 8.64s
615:	learn: 0.1036619	total: 13.8s	remaining: 8.61s
616:	learn: 0.1036605	total: 13.8s	remaining: 8.59s
617:	learn: 0.1036562	total: 13.9s	remaining: 8.57s

618:	learn: 0.1036527	total: 13.9s	remaining: 8.54s
619:	learn: 0.1036502	total: 13.9s	remaining: 8.52s
620:	learn: 0.1036441	total: 13.9s	remaining: 8.49s
621:	learn: 0.1036398	total: 13.9s	remaining: 8.47s
622:	learn: 0.1036363	total: 13.9s	remaining: 8.44s
623:	learn: 0.1036324	total: 14s	remaining: 8.42s
624:	learn: 0.1036292	total: 14s	remaining: 8.39s
625:	learn: 0.1036223	total: 14s	remaining: 8.37s
626:	learn: 0.1036185	total: 14s	remaining: 8.34s
627:	learn: 0.1036161	total: 14s	remaining: 8.32s
628:	learn: 0.1036116	total: 14.1s	remaining: 8.3s
629:	learn: 0.1036082	total: 14.1s	remaining: 8.28s
630:	learn: 0.1036048	total: 14.1s	remaining: 8.26s
631:	learn: 0.1035989	total: 14.1s	remaining: 8.23s
632:	learn: 0.1035956	total: 14.2s	remaining: 8.21s
633:	learn: 0.1035920	total: 14.2s	remaining: 8.19s
634:	learn: 0.1035878	total: 14.2s	remaining: 8.16s
635:	learn: 0.1035827	total: 14.2s	remaining: 8.14s
636:	learn: 0.1035782	total: 14.2s	remaining: 8.11s
637:	learn: 0.1035762	total: 14.3s	remaining: 8.09s
638:	learn: 0.1035736	total: 14.3s	remaining: 8.06s
639:	learn: 0.1035714	total: 14.3s	remaining: 8.04s
640:	learn: 0.1035701	total: 14.3s	remaining: 8.01s
641:	learn: 0.1035656	total: 14.3s	remaining: 7.99s
642:	learn: 0.1035632	total: 14.4s	remaining: 7.97s
643:	learn: 0.1035593	total: 14.4s	remaining: 7.94s
644:	learn: 0.1035542	total: 14.4s	remaining: 7.92s
645:	learn: 0.1035488	total: 14.4s	remaining: 7.89s
646:	learn: 0.1035455	total: 14.4s	remaining: 7.87s
647:	learn: 0.1035422	total: 14.4s	remaining: 7.84s
648:	learn: 0.1035382	total: 14.5s	remaining: 7.82s
649:	learn: 0.1035337	total: 14.5s	remaining: 7.8s
650:	learn: 0.1035309	total: 14.5s	remaining: 7.78s
651:	learn: 0.1035262	total: 14.5s	remaining: 7.75s
652:	learn: 0.1035160	total: 14.5s	remaining: 7.73s
653:	learn: 0.1035142	total: 14.6s	remaining: 7.7s
654:	learn: 0.1035119	total: 14.6s	remaining: 7.68s
655:	learn: 0.1035088	total: 14.6s	remaining: 7.65s
656:	learn: 0.1035039	total: 14.6s	remaining: 7.63s
657:	learn: 0.1035007	total: 14.6s	remaining: 7.6s
658:	learn: 0.1034971	total: 14.6s	remaining: 7.58s
659:	learn: 0.1034932	total: 14.7s	remaining: 7.56s
660:	learn: 0.1034906	total: 14.7s	remaining: 7.53s
661:	learn: 0.1034886	total: 14.7s	remaining: 7.51s
662:	learn: 0.1034859	total: 14.7s	remaining: 7.49s
663:	learn: 0.1034826	total: 14.7s	remaining: 7.46s
664:	learn: 0.1034816	total: 14.8s	remaining: 7.43s
665:	learn: 0.1034776	total: 14.8s	remaining: 7.41s
666:	learn: 0.1034754	total: 14.8s	remaining: 7.39s
667:	learn: 0.1034736	total: 14.8s	remaining: 7.36s
668:	learn: 0.1034706	total: 14.8s	remaining: 7.34s
669:	learn: 0.1034673	total: 14.8s	remaining: 7.31s
670:	learn: 0.1034650	total: 14.9s	remaining: 7.29s
671:	learn: 0.1034613	total: 14.9s	remaining: 7.27s
672:	learn: 0.1034591	total: 14.9s	remaining: 7.25s
673:	learn: 0.1034544	total: 14.9s	remaining: 7.22s
674:	learn: 0.1034495	total: 14.9s	remaining: 7.2s
675:	learn: 0.1034458	total: 15s	remaining: 7.17s
676:	learn: 0.1034401	total: 15s	remaining: 7.15s
677:	learn: 0.1034383	total: 15s	remaining: 7.12s
678:	learn: 0.1034330	total: 15s	remaining: 7.1s
679:	learn: 0.1034300	total: 15.1s	remaining: 7.08s
680:	learn: 0.1034276	total: 15.1s	remaining: 7.06s
681:	learn: 0.1034258	total: 15.1s	remaining: 7.04s

682:	learn: 0.1034234	total: 15.1s	remaining: 7.02s
683:	learn: 0.1034212	total: 15.1s	remaining: 6.99s
684:	learn: 0.1034175	total: 15.2s	remaining: 6.97s
685:	learn: 0.1034135	total: 15.2s	remaining: 6.95s
686:	learn: 0.1034102	total: 15.2s	remaining: 6.92s
687:	learn: 0.1034048	total: 15.2s	remaining: 6.9s
688:	learn: 0.1034002	total: 15.2s	remaining: 6.88s
689:	learn: 0.1033980	total: 15.3s	remaining: 6.85s
690:	learn: 0.1033960	total: 15.3s	remaining: 6.83s
691:	learn: 0.1033914	total: 15.3s	remaining: 6.81s
692:	learn: 0.1033869	total: 15.3s	remaining: 6.78s
693:	learn: 0.1033851	total: 15.3s	remaining: 6.76s
694:	learn: 0.1033802	total: 15.3s	remaining: 6.73s
695:	learn: 0.1033776	total: 15.4s	remaining: 6.71s
696:	learn: 0.1033746	total: 15.4s	remaining: 6.68s
697:	learn: 0.1033727	total: 15.4s	remaining: 6.66s
698:	learn: 0.1033702	total: 15.4s	remaining: 6.64s
699:	learn: 0.1033686	total: 15.4s	remaining: 6.61s
700:	learn: 0.1033647	total: 15.5s	remaining: 6.59s
701:	learn: 0.1033632	total: 15.5s	remaining: 6.57s
702:	learn: 0.1033605	total: 15.5s	remaining: 6.55s
703:	learn: 0.1033574	total: 15.5s	remaining: 6.52s
704:	learn: 0.1033531	total: 15.5s	remaining: 6.5s
705:	learn: 0.1033479	total: 15.6s	remaining: 6.48s
706:	learn: 0.1033444	total: 15.6s	remaining: 6.45s
707:	learn: 0.1033416	total: 15.6s	remaining: 6.43s
708:	learn: 0.1033373	total: 15.6s	remaining: 6.41s
709:	learn: 0.1033337	total: 15.6s	remaining: 6.38s
710:	learn: 0.1033318	total: 15.6s	remaining: 6.36s
711:	learn: 0.1033287	total: 15.7s	remaining: 6.33s
712:	learn: 0.1033251	total: 15.7s	remaining: 6.31s
713:	learn: 0.1033218	total: 15.7s	remaining: 6.29s
714:	learn: 0.1033196	total: 15.7s	remaining: 6.27s
715:	learn: 0.1033148	total: 15.7s	remaining: 6.24s
716:	learn: 0.1033082	total: 15.8s	remaining: 6.22s
717:	learn: 0.1033041	total: 15.8s	remaining: 6.2s
718:	learn: 0.1032998	total: 15.8s	remaining: 6.17s
719:	learn: 0.1032953	total: 15.8s	remaining: 6.15s
720:	learn: 0.1032909	total: 15.8s	remaining: 6.13s
721:	learn: 0.1032861	total: 15.9s	remaining: 6.1s
722:	learn: 0.1032842	total: 15.9s	remaining: 6.08s
723:	learn: 0.1032814	total: 15.9s	remaining: 6.06s
724:	learn: 0.1032766	total: 15.9s	remaining: 6.04s
725:	learn: 0.1032682	total: 15.9s	remaining: 6.01s
726:	learn: 0.1032665	total: 16s	remaining: 5.99s
727:	learn: 0.1032627	total: 16s	remaining: 5.97s
728:	learn: 0.1032595	total: 16s	remaining: 5.94s
729:	learn: 0.1032577	total: 16s	remaining: 5.92s
730:	learn: 0.1032561	total: 16s	remaining: 5.9s
731:	learn: 0.1032501	total: 16.1s	remaining: 5.88s
732:	learn: 0.1032460	total: 16.1s	remaining: 5.86s
733:	learn: 0.1032420	total: 16.1s	remaining: 5.83s
734:	learn: 0.1032402	total: 16.1s	remaining: 5.81s
735:	learn: 0.1032364	total: 16.1s	remaining: 5.79s
736:	learn: 0.1032352	total: 16.2s	remaining: 5.76s
737:	learn: 0.1032316	total: 16.2s	remaining: 5.74s
738:	learn: 0.1032261	total: 16.2s	remaining: 5.72s
739:	learn: 0.1032219	total: 16.2s	remaining: 5.7s
740:	learn: 0.1032173	total: 16.2s	remaining: 5.67s
741:	learn: 0.1032151	total: 16.3s	remaining: 5.65s
742:	learn: 0.1032138	total: 16.3s	remaining: 5.63s
743:	learn: 0.1032094	total: 16.3s	remaining: 5.6s
744:	learn: 0.1032072	total: 16.3s	remaining: 5.58s
745:	learn: 0.1032043	total: 16.3s	remaining: 5.56s



746:	learn: 0.1032024	total: 16.3s	remaining: 5.54s
747:	learn: 0.1032009	total: 16.4s	remaining: 5.51s
748:	learn: 0.1031966	total: 16.4s	remaining: 5.49s
749:	learn: 0.1031940	total: 16.4s	remaining: 5.47s
750:	learn: 0.1031926	total: 16.4s	remaining: 5.44s
751:	learn: 0.1031910	total: 16.4s	remaining: 5.42s
752:	learn: 0.1031893	total: 16.5s	remaining: 5.4s
753:	learn: 0.1031850	total: 16.5s	remaining: 5.37s
754:	learn: 0.1031817	total: 16.5s	remaining: 5.35s
755:	learn: 0.1031784	total: 16.5s	remaining: 5.33s
756:	learn: 0.1031763	total: 16.5s	remaining: 5.31s
757:	learn: 0.1031747	total: 16.5s	remaining: 5.28s
758:	learn: 0.1031729	total: 16.6s	remaining: 5.26s
759:	learn: 0.1031703	total: 16.6s	remaining: 5.24s
760:	learn: 0.1031666	total: 16.6s	remaining: 5.21s
761:	learn: 0.1031631	total: 16.6s	remaining: 5.19s
762:	learn: 0.1031593	total: 16.6s	remaining: 5.17s
763:	learn: 0.1031578	total: 16.7s	remaining: 5.14s
764:	learn: 0.1031516	total: 16.7s	remaining: 5.12s
765:	learn: 0.1031495	total: 16.7s	remaining: 5.1s
766:	learn: 0.1031458	total: 16.7s	remaining: 5.08s
767:	learn: 0.1031412	total: 16.7s	remaining: 5.05s
768:	learn: 0.1031364	total: 16.8s	remaining: 5.03s
769:	learn: 0.1031350	total: 16.8s	remaining: 5.01s
770:	learn: 0.1031317	total: 16.8s	remaining: 4.99s
771:	learn: 0.1031266	total: 16.8s	remaining: 4.96s
772:	learn: 0.1031234	total: 16.8s	remaining: 4.94s
773:	learn: 0.1031191	total: 16.8s	remaining: 4.92s
774:	learn: 0.1031160	total: 16.9s	remaining: 4.9s
775:	learn: 0.1031136	total: 16.9s	remaining: 4.87s
776:	learn: 0.1031104	total: 16.9s	remaining: 4.85s
777:	learn: 0.1031056	total: 16.9s	remaining: 4.83s
778:	learn: 0.1031037	total: 16.9s	remaining: 4.81s
779:	learn: 0.1031017	total: 17s	remaining: 4.79s
780:	learn: 0.1030967	total: 17s	remaining: 4.76s
781:	learn: 0.1030945	total: 17s	remaining: 4.74s
782:	learn: 0.1030915	total: 17s	remaining: 4.72s
783:	learn: 0.1030890	total: 17.1s	remaining: 4.7s
784:	learn: 0.1030867	total: 17.1s	remaining: 4.67s
785:	learn: 0.1030845	total: 17.1s	remaining: 4.65s
786:	learn: 0.1030824	total: 17.1s	remaining: 4.63s
787:	learn: 0.1030807	total: 17.1s	remaining: 4.61s
788:	learn: 0.1030790	total: 17.1s	remaining: 4.59s
789:	learn: 0.1030739	total: 17.2s	remaining: 4.56s
790:	learn: 0.1030698	total: 17.2s	remaining: 4.54s
791:	learn: 0.1030657	total: 17.2s	remaining: 4.52s
792:	learn: 0.1030631	total: 17.2s	remaining: 4.5s
793:	learn: 0.1030620	total: 17.2s	remaining: 4.47s
794:	learn: 0.1030598	total: 17.3s	remaining: 4.45s
795:	learn: 0.1030569	total: 17.3s	remaining: 4.43s
796:	learn: 0.1030513	total: 17.3s	remaining: 4.41s
797:	learn: 0.1030488	total: 17.3s	remaining: 4.38s
798:	learn: 0.1030465	total: 17.3s	remaining: 4.36s
799:	learn: 0.1030443	total: 17.4s	remaining: 4.34s
800:	learn: 0.1030401	total: 17.4s	remaining: 4.32s
801:	learn: 0.1030378	total: 17.4s	remaining: 4.29s
802:	learn: 0.1030349	total: 17.4s	remaining: 4.27s
803:	learn: 0.1030323	total: 17.4s	remaining: 4.25s
804:	learn: 0.1030304	total: 17.4s	remaining: 4.23s
805:	learn: 0.1030281	total: 17.5s	remaining: 4.21s
806:	learn: 0.1030184	total: 17.5s	remaining: 4.18s
807:	learn: 0.1030165	total: 17.5s	remaining: 4.16s
808:	learn: 0.1030143	total: 17.5s	remaining: 4.14s
809:	learn: 0.1030103	total: 17.5s	remaining: 4.12s

810:	learn: 0.1030077	total: 17.6s	remaining: 4.09s
811:	learn: 0.1030063	total: 17.6s	remaining: 4.07s
812:	learn: 0.1030038	total: 17.6s	remaining: 4.05s
813:	learn: 0.1029979	total: 17.6s	remaining: 4.03s
814:	learn: 0.1029962	total: 17.6s	remaining: 4s
815:	learn: 0.1029934	total: 17.7s	remaining: 3.98s
816:	learn: 0.1029919	total: 17.7s	remaining: 3.96s
817:	learn: 0.1029900	total: 17.7s	remaining: 3.94s
818:	learn: 0.1029857	total: 17.7s	remaining: 3.91s
819:	learn: 0.1029821	total: 17.7s	remaining: 3.89s
820:	learn: 0.1029795	total: 17.8s	remaining: 3.87s
821:	learn: 0.1029770	total: 17.8s	remaining: 3.85s
822:	learn: 0.1029726	total: 17.8s	remaining: 3.83s
823:	learn: 0.1029709	total: 17.8s	remaining: 3.8s
824:	learn: 0.1029694	total: 17.8s	remaining: 3.78s
825:	learn: 0.1029680	total: 17.8s	remaining: 3.76s
826:	learn: 0.1029661	total: 17.9s	remaining: 3.74s
827:	learn: 0.1029638	total: 17.9s	remaining: 3.71s
828:	learn: 0.1029597	total: 17.9s	remaining: 3.69s
829:	learn: 0.1029573	total: 17.9s	remaining: 3.67s
830:	learn: 0.1029539	total: 17.9s	remaining: 3.65s
831:	learn: 0.1029511	total: 18s	remaining: 3.63s
832:	learn: 0.1029487	total: 18s	remaining: 3.61s
833:	learn: 0.1029452	total: 18s	remaining: 3.59s
834:	learn: 0.1029409	total: 18s	remaining: 3.56s
835:	learn: 0.1029386	total: 18.1s	remaining: 3.54s
836:	learn: 0.1029370	total: 18.1s	remaining: 3.52s
837:	learn: 0.1029360	total: 18.1s	remaining: 3.5s
838:	learn: 0.1029346	total: 18.1s	remaining: 3.47s
839:	learn: 0.1029307	total: 18.1s	remaining: 3.45s
840:	learn: 0.1029292	total: 18.1s	remaining: 3.43s
841:	learn: 0.1029282	total: 18.2s	remaining: 3.41s
842:	learn: 0.1029254	total: 18.2s	remaining: 3.39s
843:	learn: 0.1029218	total: 18.2s	remaining: 3.36s
844:	learn: 0.1029176	total: 18.2s	remaining: 3.34s
845:	learn: 0.1029130	total: 18.2s	remaining: 3.32s
846:	learn: 0.1029092	total: 18.3s	remaining: 3.3s
847:	learn: 0.1029074	total: 18.3s	remaining: 3.27s
848:	learn: 0.1029064	total: 18.3s	remaining: 3.25s
849:	learn: 0.1029042	total: 18.3s	remaining: 3.23s
850:	learn: 0.1029012	total: 18.3s	remaining: 3.21s
851:	learn: 0.1028996	total: 18.3s	remaining: 3.19s
852:	learn: 0.1028914	total: 18.4s	remaining: 3.17s
853:	learn: 0.1028905	total: 18.4s	remaining: 3.14s
854:	learn: 0.1028889	total: 18.4s	remaining: 3.12s
855:	learn: 0.1028844	total: 18.4s	remaining: 3.1s
856:	learn: 0.1028819	total: 18.4s	remaining: 3.08s
857:	learn: 0.1028788	total: 18.5s	remaining: 3.06s
858:	learn: 0.1028769	total: 18.5s	remaining: 3.03s
859:	learn: 0.1028731	total: 18.5s	remaining: 3.01s
860:	learn: 0.1028711	total: 18.5s	remaining: 2.99s
861:	learn: 0.1028677	total: 18.5s	remaining: 2.97s
862:	learn: 0.1028653	total: 18.6s	remaining: 2.94s
863:	learn: 0.1028632	total: 18.6s	remaining: 2.92s
864:	learn: 0.1028612	total: 18.6s	remaining: 2.9s
865:	learn: 0.1028598	total: 18.6s	remaining: 2.88s
866:	learn: 0.1028587	total: 18.6s	remaining: 2.86s
867:	learn: 0.1028570	total: 18.6s	remaining: 2.83s
868:	learn: 0.1028537	total: 18.7s	remaining: 2.81s
869:	learn: 0.1028504	total: 18.7s	remaining: 2.79s
870:	learn: 0.1028474	total: 18.7s	remaining: 2.77s
871:	learn: 0.1028462	total: 18.7s	remaining: 2.75s
872:	learn: 0.1028438	total: 18.7s	remaining: 2.73s
873:	learn: 0.1028357	total: 18.8s	remaining: 2.7s

874:	learn: 0.1028344	total: 18.8s	remaining: 2.68s
875:	learn: 0.1028313	total: 18.8s	remaining: 2.66s
876:	learn: 0.1028295	total: 18.8s	remaining: 2.64s
877:	learn: 0.1028271	total: 18.8s	remaining: 2.62s
878:	learn: 0.1028257	total: 18.9s	remaining: 2.6s
879:	learn: 0.1028240	total: 18.9s	remaining: 2.57s
880:	learn: 0.1028184	total: 18.9s	remaining: 2.55s
881:	learn: 0.1028163	total: 18.9s	remaining: 2.53s
882:	learn: 0.1028150	total: 18.9s	remaining: 2.51s
883:	learn: 0.1028127	total: 19s	remaining: 2.49s
884:	learn: 0.1028103	total: 19s	remaining: 2.47s
885:	learn: 0.1028088	total: 19s	remaining: 2.44s
886:	learn: 0.1028075	total: 19s	remaining: 2.42s
887:	learn: 0.1028048	total: 19s	remaining: 2.4s
888:	learn: 0.1028039	total: 19.1s	remaining: 2.38s
889:	learn: 0.1028028	total: 19.1s	remaining: 2.36s
890:	learn: 0.1028011	total: 19.1s	remaining: 2.33s
891:	learn: 0.1027990	total: 19.1s	remaining: 2.31s
892:	learn: 0.1027964	total: 19.1s	remaining: 2.29s
893:	learn: 0.1027945	total: 19.1s	remaining: 2.27s
894:	learn: 0.1027931	total: 19.2s	remaining: 2.25s
895:	learn: 0.1027903	total: 19.2s	remaining: 2.23s
896:	learn: 0.1027882	total: 19.2s	remaining: 2.21s
897:	learn: 0.1027866	total: 19.2s	remaining: 2.18s
898:	learn: 0.1027835	total: 19.2s	remaining: 2.16s
899:	learn: 0.1027812	total: 19.3s	remaining: 2.14s
900:	learn: 0.1027799	total: 19.3s	remaining: 2.12s
901:	learn: 0.1027784	total: 19.3s	remaining: 2.1s
902:	learn: 0.1027766	total: 19.3s	remaining: 2.08s
903:	learn: 0.1027750	total: 19.3s	remaining: 2.05s
904:	learn: 0.1027739	total: 19.4s	remaining: 2.03s
905:	learn: 0.1027712	total: 19.4s	remaining: 2.01s
906:	learn: 0.1027696	total: 19.4s	remaining: 1.99s
907:	learn: 0.1027663	total: 19.4s	remaining: 1.97s
908:	learn: 0.1027627	total: 19.4s	remaining: 1.95s
909:	learn: 0.1027610	total: 19.4s	remaining: 1.92s
910:	learn: 0.1027589	total: 19.5s	remaining: 1.9s
911:	learn: 0.1027566	total: 19.5s	remaining: 1.88s
912:	learn: 0.1027553	total: 19.5s	remaining: 1.86s
913:	learn: 0.1027538	total: 19.5s	remaining: 1.84s
914:	learn: 0.1027527	total: 19.5s	remaining: 1.81s
915:	learn: 0.1027509	total: 19.6s	remaining: 1.79s
916:	learn: 0.1027498	total: 19.6s	remaining: 1.77s
917:	learn: 0.1027487	total: 19.6s	remaining: 1.75s
918:	learn: 0.1027472	total: 19.6s	remaining: 1.73s
919:	learn: 0.1027456	total: 19.6s	remaining: 1.71s
920:	learn: 0.1027446	total: 19.7s	remaining: 1.69s
921:	learn: 0.1027433	total: 19.7s	remaining: 1.66s
922:	learn: 0.1027423	total: 19.7s	remaining: 1.64s
923:	learn: 0.1027399	total: 19.7s	remaining: 1.62s
924:	learn: 0.1027383	total: 19.7s	remaining: 1.6s
925:	learn: 0.1027370	total: 19.7s	remaining: 1.58s
926:	learn: 0.1027337	total: 19.8s	remaining: 1.56s
927:	learn: 0.1027313	total: 19.8s	remaining: 1.53s
928:	learn: 0.1027297	total: 19.8s	remaining: 1.51s
929:	learn: 0.1027277	total: 19.8s	remaining: 1.49s
930:	learn: 0.1027256	total: 19.8s	remaining: 1.47s
931:	learn: 0.1027236	total: 19.9s	remaining: 1.45s
932:	learn: 0.1027214	total: 19.9s	remaining: 1.43s
933:	learn: 0.1027199	total: 19.9s	remaining: 1.41s
934:	learn: 0.1027182	total: 19.9s	remaining: 1.39s
935:	learn: 0.1027166	total: 20s	remaining: 1.36s
936:	learn: 0.1027149	total: 20s	remaining: 1.34s
937:	learn: 0.1027137	total: 20s	remaining: 1.32s

938:	learn: 0.1027123	total: 20.1s	remaining: 1.3s
939:	learn: 0.1027098	total: 20.1s	remaining: 1.28s
940:	learn: 0.1027084	total: 20.1s	remaining: 1.26s
941:	learn: 0.1027071	total: 20.2s	remaining: 1.24s
942:	learn: 0.1027060	total: 20.2s	remaining: 1.22s
943:	learn: 0.1027038	total: 20.2s	remaining: 1.2s
944:	learn: 0.1027020	total: 20.3s	remaining: 1.18s
945:	learn: 0.1026989	total: 20.3s	remaining: 1.16s
946:	learn: 0.1026954	total: 20.4s	remaining: 1.14s
947:	learn: 0.1026933	total: 20.4s	remaining: 1.12s
948:	learn: 0.1026895	total: 20.4s	remaining: 1.1s
949:	learn: 0.1026870	total: 20.5s	remaining: 1.08s
950:	learn: 0.1026846	total: 20.5s	remaining: 1.06s
951:	learn: 0.1026816	total: 20.6s	remaining: 1.04s
952:	learn: 0.1026804	total: 20.6s	remaining: 1.02s
953:	learn: 0.1026794	total: 20.7s	remaining: 997ms
954:	learn: 0.1026781	total: 20.7s	remaining: 976ms
955:	learn: 0.1026747	total: 20.7s	remaining: 954ms
956:	learn: 0.1026714	total: 20.8s	remaining: 934ms
957:	learn: 0.1026685	total: 20.8s	remaining: 913ms
958:	learn: 0.1026658	total: 20.9s	remaining: 893ms
959:	learn: 0.1026629	total: 20.9s	remaining: 872ms
960:	learn: 0.1026611	total: 21s	remaining: 851ms
961:	learn: 0.1026595	total: 21s	remaining: 830ms
962:	learn: 0.1026577	total: 21.1s	remaining: 809ms
963:	learn: 0.1026562	total: 21.1s	remaining: 788ms
964:	learn: 0.1026536	total: 21.2s	remaining: 767ms
965:	learn: 0.1026497	total: 21.2s	remaining: 746ms
966:	learn: 0.1026470	total: 21.2s	remaining: 724ms
967:	learn: 0.1026453	total: 21.3s	remaining: 703ms
968:	learn: 0.1026441	total: 21.3s	remaining: 681ms
969:	learn: 0.1026398	total: 21.3s	remaining: 660ms
970:	learn: 0.1026361	total: 21.4s	remaining: 638ms
971:	learn: 0.1026334	total: 21.4s	remaining: 616ms
972:	learn: 0.1026327	total: 21.4s	remaining: 595ms
973:	learn: 0.1026303	total: 21.5s	remaining: 573ms
974:	learn: 0.1026282	total: 21.5s	remaining: 552ms
975:	learn: 0.1026266	total: 21.6s	remaining: 530ms
976:	learn: 0.1026238	total: 21.6s	remaining: 508ms
977:	learn: 0.1026226	total: 21.6s	remaining: 486ms
978:	learn: 0.1026214	total: 21.6s	remaining: 464ms
979:	learn: 0.1026193	total: 21.7s	remaining: 442ms
980:	learn: 0.1026181	total: 21.7s	remaining: 420ms
981:	learn: 0.1026167	total: 21.8s	remaining: 399ms
982:	learn: 0.1026137	total: 21.8s	remaining: 377ms
983:	learn: 0.1026120	total: 21.8s	remaining: 355ms
984:	learn: 0.1026099	total: 21.8s	remaining: 333ms
985:	learn: 0.1026049	total: 21.9s	remaining: 311ms
986:	learn: 0.1026029	total: 21.9s	remaining: 289ms
987:	learn: 0.1026015	total: 22s	remaining: 267ms
988:	learn: 0.1025985	total: 22s	remaining: 245ms
989:	learn: 0.1025958	total: 22.1s	remaining: 223ms
990:	learn: 0.1025937	total: 22.1s	remaining: 201ms
991:	learn: 0.1025910	total: 22.1s	remaining: 179ms
992:	learn: 0.1025888	total: 22.2s	remaining: 156ms
993:	learn: 0.1025874	total: 22.2s	remaining: 134ms
994:	learn: 0.1025850	total: 22.3s	remaining: 112ms
995:	learn: 0.1025836	total: 22.3s	remaining: 89.7ms
996:	learn: 0.1025827	total: 22.4s	remaining: 67.3ms
997:	learn: 0.1025803	total: 22.4s	remaining: 44.9ms
998:	learn: 0.1025787	total: 22.5s	remaining: 22.5ms
999:	learn: 0.1025761	total: 22.5s	remaining: 0us

In [108... df\_results.sort\_values('MSE', ascending=False)

Out[108]:

	Model	MSE	MAE	R-squared
2	LassoRegression	0.032220	0.141360	-0.000058
3	ElasticNetRegression	0.032220	0.141360	-0.000058
8	AdaBoostRegressor	0.018235	0.109502	0.434018
5	ElasticNetCVRegression	0.015391	0.092497	0.522296
4	LassoCVRegression	0.015391	0.092497	0.522297
0	LinearRegression	0.015390	0.092497	0.522312
1	RidgeRegression	0.015390	0.092497	0.522312
10	GradientBoostingRegressor	0.013251	0.085691	0.588704
6	DecisionTreeRegressor	0.011838	0.079320	0.632584
9	LGBMRegressor	0.011461	0.079813	0.644287
7	RandomForestRegressor	0.011427	0.078165	0.645335
11	XGBoostRegression	0.010843	0.076751	0.663458
12	CatBoostRegressor	0.010807	0.076598	0.664569

## Conclusion

If we consider MSE and MAE, a **lower value for MSE and MAE indicates better model performance**. In this case, **CatBoost Regressor model has the lowest MSE and MAE value and the highest R-squared value** compared to other models. Based on these metrics, Catboost Regressor is considered the best model in terms of prediction accuracy.

However, if we consider R-squared, a value closer to 1 indicates a better model in explaining the variation in the data. In this case, **CatBoost Regressor also has the highest R-squared value** compared to other models. **This indicates that the CatBoost Regressor model is better at explaining the variation in the data compared to other models.**

Based on the evaluation of MSE, MAE, and R-squared, it can be concluded that the **CatBoost Regression model is the best model in terms of prediction accuracy and its ability to explain the variation in the data.**

In [109...]

```
# #@title Convert ipynb to HTML in Colab
# # Upload ipynb
# from google.colab import files
# f = files.upload()

# # Convert ipynb to html
# import subprocess
# file0 = list(f.keys())[0]
# _ = subprocess.run(["pip", "install", "nbconvert"])
# _ = subprocess.run(["jupyter", "nbconvert", file0, "--to", "html"])

# # download the html
# files.download(file0[:-5]+"html")
```