



Cinema Audience Forecasting Challenge

Objective: Predict cinema audience count for future dates based on historical data

1. Data Loading

Import Libraries

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import r2_score, mean_absolute_error
import lightgbm as lgb
import xgboost as xgb
from catboost import CatBoostRegressor
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform, randint
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from scipy.optimize import minimize
from sklearn.metrics import mean_squared_error
```

Load CSVs into DataFrames

```
In [2]: movie_theater_id_relation = pd.read_csv("movie_theater_id_relation.csv")
date_info = pd.read_csv("date_info.csv")
sample_submission = pd.read_csv("sample_submission.csv")
booknow_theaters = pd.read_csv("booknow_theaters.csv")
cinePOS_booking = pd.read_csv("cinePOS_booking.csv")
cinePOS_theaters = pd.read_csv("cinePOS_theaters.csv")
booknow_visits = pd.read_csv("booknow_visits.csv")
booknow_booking = pd.read_csv("booknow_booking.csv")
```

2. Exploratory Data Analysis

2.1 Quick Exploration

```
In [3]: # Quick check of shapes
```

```

print("movie_theater_id_relation:", movie_theater_id_relation.shape)
print("date_info:", date_info.shape)
print("sample_submission:", sample_submission.shape)
print("booknow_theaters:", booknow_theaters.shape)
print("cinePOS_booking:", cinePOS_booking.shape)
print("cinePOS_theaters:", cinePOS_theaters.shape)
print("booknow_visits:", booknow_visits.shape)
print("booknow_booking:", booknow_booking.shape)

```

```

movie_theater_id_relation: (150, 2)
date_info: (547, 2)
sample_submission: (38062, 2)
booknow_theaters: (829, 5)
cinePOS_booking: (1641966, 4)
cinePOS_theaters: (4690, 5)
booknow_visits: (214046, 3)
booknow_booking: (68336, 4)

```

In [4]: `movie_theater_id_relation.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   book_theater_id  150 non-null   object
1   cine_theater_id  150 non-null   object
dtypes: object(2)
memory usage: 2.5+ KB

```

In [5]: `date_info.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 547 entries, 0 to 546
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   show_date       547 non-null   object
1   day_of_week     547 non-null   object
dtypes: object(2)
memory usage: 8.7+ KB

```

In [6]: `booknow_theaters.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 829 entries, 0 to 828
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   book_theater_id  314 non-null   object
1   theater_type     829 non-null   object
2   theater_area     829 non-null   object
3   latitude         829 non-null   float64
4   longitude        829 non-null   float64
dtypes: float64(2), object(3)
memory usage: 32.5+ KB

```

In [7]: `cinePOS_booking.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1641966 entries, 0 to 1641965
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   cine_theater_id  1641966 non-null object
1   show_datetime    1641966 non-null object
2   booking_datetime 1641966 non-null object
3   tickets_sold     1641966 non-null int64
dtypes: int64(1), object(3)
memory usage: 50.1+ MB

```

In [8]: `cinePOS_theaters.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4690 entries, 0 to 4689
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   cine_theater_id  4690 non-null   object
1   theater_type     4690 non-null   object
2   theater_area     4690 non-null   object
3   latitude         829 non-null   float64
4   longitude        829 non-null   float64
dtypes: float64(2), object(3)
memory usage: 183.3+ KB

```

In [9]: `booknow_visits.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 214046 entries, 0 to 214045
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   book_theater_id  214046 non-null object
1   show_date        214046 non-null object
2   audience_count   214046 non-null int64
dtypes: int64(1), object(2)
memory usage: 4.9+ MB

```

```
In [10]: booknow_booking.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 68336 entries, 0 to 68335
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   book_theater_id       68336 non-null  object
1   show_datetime         68336 non-null  object
2   booking_datetime      68336 non-null  object
3   tickets_booked        68336 non-null  int64
dtypes: int64(1), object(3)
memory usage: 2.1+ MB
```

```
In [11]: sample_submission.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38062 entries, 0 to 38061
Data columns (total 2 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   ID                    38062 non-null  object
1   audience_count        38062 non-null  int64
dtypes: int64(1), object(1)
memory usage: 594.8+ KB
```

```
In [12]: def analyze_data_quality(df, name):
    total_records = len(df)
    missing_data = df.isnull().sum()
    missing_pct = (missing_data / total_records * 100).round(2)
    duplicates = df.duplicated().sum()

    return {
        'total_records': total_records,
        'missing_data': missing_data.sum(),
        'duplicates': duplicates,
        'completeness': ((total_records * len(df.columns) - missing_data.sum())
    }

    datasets = {
        'bbooknow_visits': booknow_visits,
        'booknow_booking': booknow_booking,
        'booknow_theaters': booknow_theaters,
        'cinePOS_booking': cinePOS_booking,
        'cinePOS_theaters': cinePOS_theaters,
        'date_info': date_info,
        'movie_theater_id_relation': movie_theater_id_relation,
        'sample_submission': sample_submission
    }

    quality_summary = {}
    for name, df in datasets.items():
        quality_summary[name] = analyze_data_quality(df, name)
```

```

In [13]: # Missing Data Visualization
fig, ax = plt.subplots(1, 1, figsize=(12, 6))

names = list(quality_summary.keys())
missing_data = [quality_summary[name]['missing_data'] for name in names]
bars = ax.bar(range(len(names)), missing_data, color='lightcoral', alpha=0.8)
ax.set_title('Total Missing Values', fontsize=14, fontweight='bold')
ax.set_xticks(range(len(names)))
ax.set_xticklabels([name.replace(' ', '\n') for name in names], rotation=45, ha='center')
ax.set_ylabel('Missing Values Count')

# Add value labels on bars
for i, v in enumerate(missing_data):
    if v > 0:
        ax.text(i, v + max(missing_data)*0.01, f'{v:,}', ha='center', va='bottom')

plt.tight_layout()
plt.show()

```



2.2 booknow_visits Analysis

```

In [14]: booknow_visits.shape

```

```

Out[14]: (214046, 3)

```

```

In [15]: booknow_visits.columns

```

```

Out[15]: Index(['book_theater_id', 'show_date', 'audience_count'], dtype='object')

```

```

In [16]: booknow_visits['show_date'].min()

```

```

Out[16]: '2023-01-01'

```

```
In [17]: bookknow_visits['show_date'].max()
```

```
Out[17]: '2024-02-28'
```

```
In [18]: bookknow_visits['book_theater_id'].nunique()
```

```
Out[18]: 826
```

```
In [19]: bookknow_visits['audience_count'].sum()
```

```
Out[19]: np.int64(8907860)
```

```
In [20]: # AUDIENCE VISITS STATISTICS
theater_performance = bookknow_visits.groupby('book_theater_id').agg({'audience': 'sum',
                              'min_audience': 'min', 'max_audience': 'max',
                              'total_shows': 'count'})
theater_performance = theater_performance.sort_values('avg_audience', ascending=False)
```

```
In [21]: # Fill the missing std with 0
theater_performance['std_audience'].fillna(0, inplace=True)
```

```
In [22]: theater_performance.isna().sum()
```

```
Out[22]: avg_audience    0
min_audience    0
max_audience    0
std_audience    0
total_shows      0
dtype: int64
```

```
In [23]: theater_performance.head(10)
```

```
Out[23]:
```

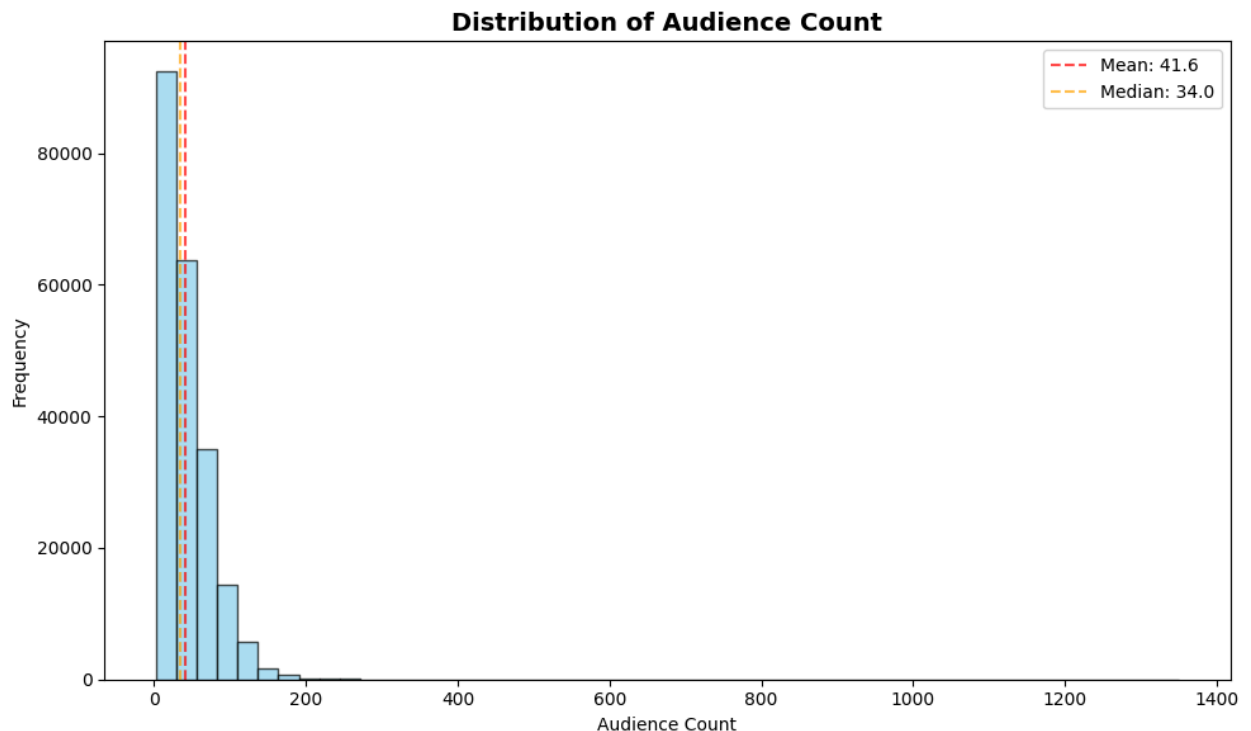
	avg_audience	min_audience	max_audience	std_audience	total_shows
book_theater_id					
book_00169	127.54	14	218	32.42	
book_00095	127.13	4	410	82.20	
book_00152	108.88	4	224	74.46	
book_00490	105.00	4	170	33.96	
book_00158	101.11	2	656	62.42	
book_00589	98.45	34	230	40.82	
book_00491	94.51	42	144	21.83	
book_00596	93.48	2	398	86.25	
book_00292	93.47	4	610	56.15	
book_00716	93.20	56	136	15.78	

```
In [24]: # 1. Audience Count Distribution
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

ax.hist(booknow_visits['audience_count'], bins=50, alpha=0.7, edgecolor='black')
ax.set_title('Distribution of Audience Count', fontsize=14, fontweight='bold')
ax.set_xlabel('Audience Count')
ax.set_ylabel('Frequency')

# Add statistics text
mean_audience = booknow_visits['audience_count'].mean()
median_audience = booknow_visits['audience_count'].median()
ax.axvline(mean_audience, color='red', linestyle='--', alpha=0.7, label=f'Mean: {mean_audience}')
ax.axvline(median_audience, color='orange', linestyle='--', alpha=0.7, label=f'Median: {median_audience}')
ax.legend()

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



```
In [25]: # 2. Top 20 Theaters by Average Audience
fig, ax = plt.subplots(1, 1, figsize=(10, 8))

top_20_theaters = theater_performance.head(20)
bars = ax.barh(range(len(top_20_theaters)), top_20_theaters['avg_audience'], color='blue')
ax.set_title('Top 20 Theaters by Average Audience', fontsize=14, fontweight='bold')
ax.set_xlabel('Average Audience Count')
ax.set_ylabel('Theater Rank')
ax.set_yticks(range(0, len(top_20_theaters), 2))
ax.set_yticklabels([f'{i+1}' for i in range(0, len(top_20_theaters), 2)])

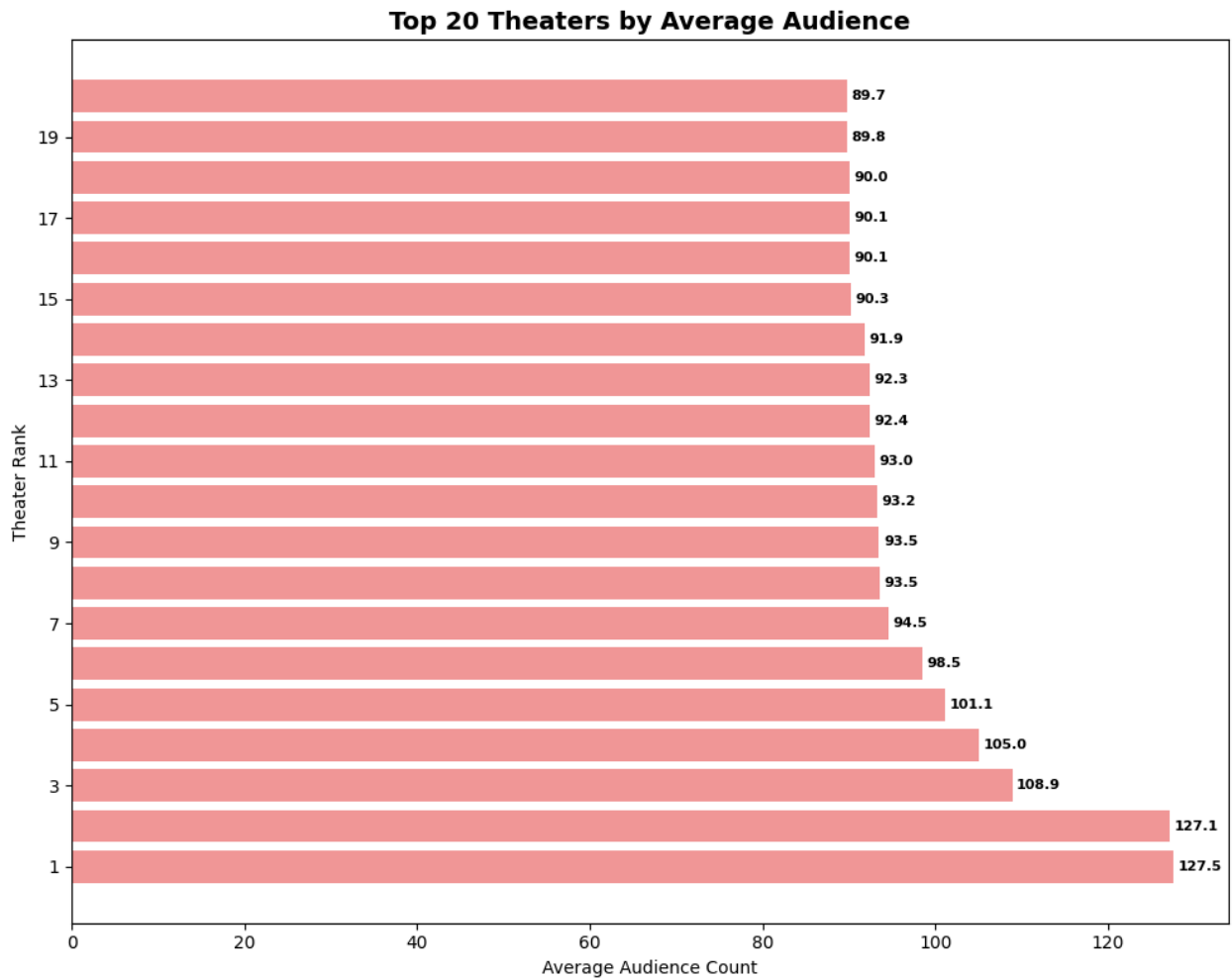
# Add value labels
for i, (bar, value) in enumerate(zip(bars, top_20_theaters['avg_audience'])):
```

```

ax.text(bar.get_width() + 0.5, bar.get_y() + bar.get_height()/2,
        f'{value:.1f}', ha='left', va='center', fontweight='bold', fontsize=10)

plt.tight_layout()
plt.show()

```



```

In [26]: # 3. Theater Performance Distribution (Box Plot)
fig, ax = plt.subplots(1, 1, figsize=(12, 8))

box_plot = ax.boxplot(theater_performance['avg_audience'], patch_artist=True,
                        boxprops=dict(facecolor='lightgreen', alpha=0.7),
                        medianprops=dict(color='red', linewidth=2))

ax.set_title('Theater Performance Distribution', fontsize=12, fontweight='bold')
ax.set_ylabel('Average Audience Count', fontsize=10)
ax.set_xlabel(['Average Audience'])

# Add statistics annotations
q1 = theater_performance['avg_audience'].quantile(0.25)
q3 = theater_performance['avg_audience'].quantile(0.75)
median = theater_performance['avg_audience'].median()
mean = theater_performance['avg_audience'].mean()

stats_text = f'Mean: {mean:.1f}\nMedian: {median:.1f}\nQ1: {q1:.1f}\nQ3: {q3:.1f}'

```

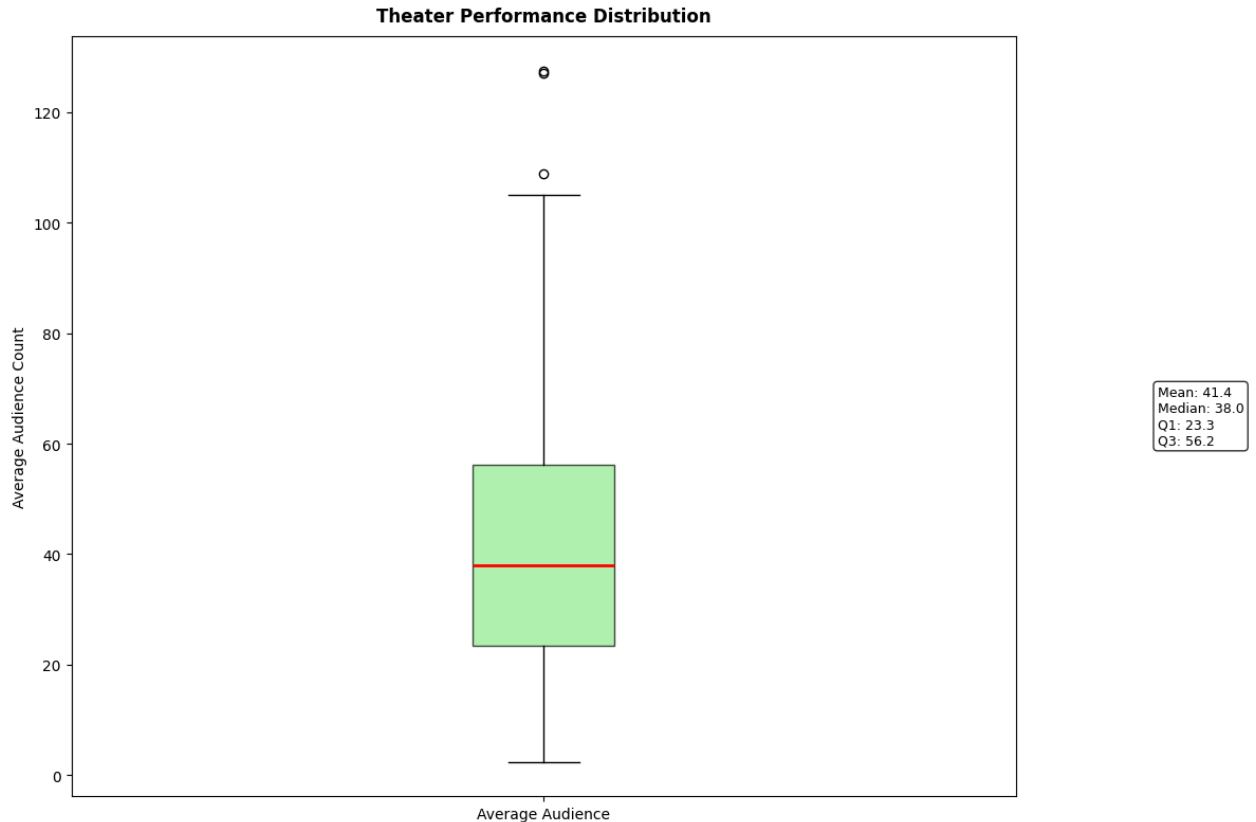


```

ax.text(1.15, 0.5, stats_text, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8),
        fontsize=9, verticalalignment='center')

plt.tight_layout()
plt.show()

```



```

In [27]: # 4. Distribution of Shows per Theater
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

ax.hist(theater_performance['total_shows'], bins=30, alpha=0.7, edgecolor='black')
ax.set_title('Distribution of Shows per Theater', fontsize=14, fontweight='bold')
ax.set_xlabel('Number of Shows')
ax.set_ylabel('Number of Theaters')

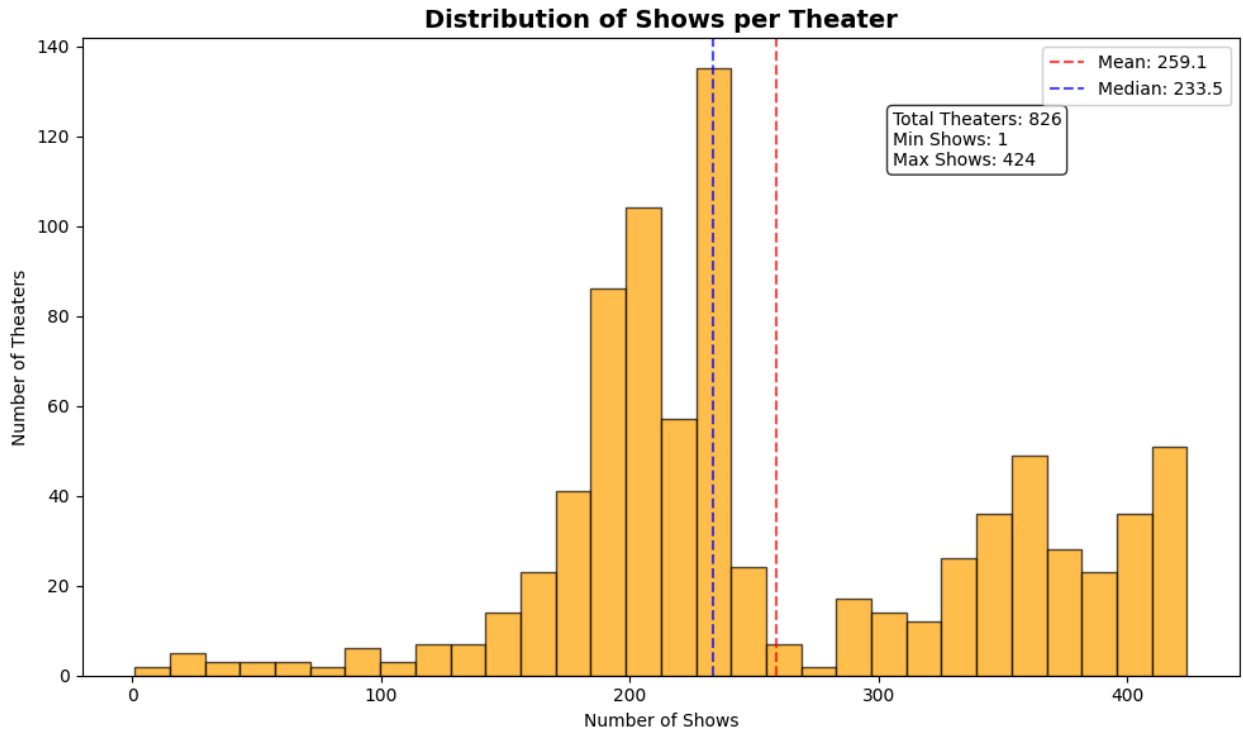
# Add statistics
mean_shows = theater_performance['total_shows'].mean()
median_shows = theater_performance['total_shows'].median()
ax.axvline(mean_shows, color='red', linestyle='--', alpha=0.7, label=f'Mean: {mean_shows}')
ax.axvline(median_shows, color='blue', linestyle='--', alpha=0.7, label=f'Median: {median_shows}')
ax.legend()

# Add summary text
summary_text = f'Total Theaters: {len(theater_performance)}\nMin Shows: {theater_performance["total_shows"].min()}\nMax Shows: {theater_performance["total_shows"].max()}'
ax.text(0.7, 0.8, summary_text, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8))

plt.tight_layout()

```

```
plt.show()
```



```
In [28]: # 5. Correlation: Shows vs Average Audience
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

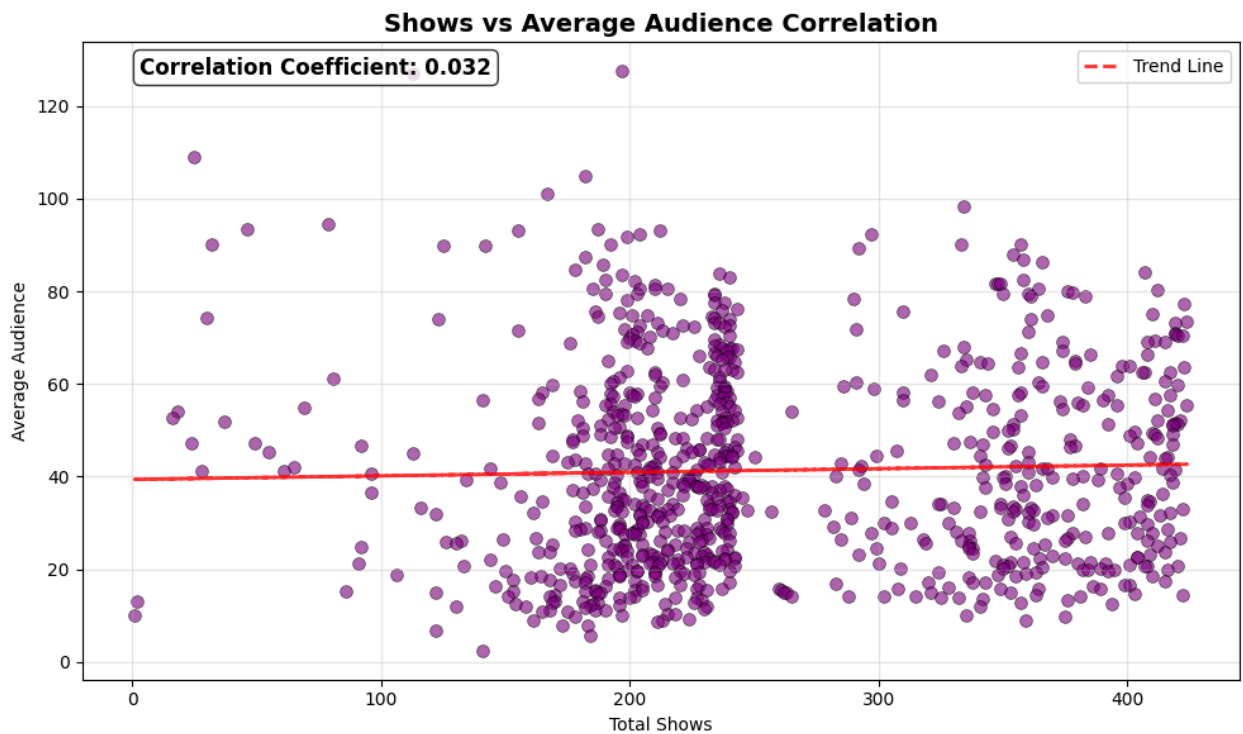
scatter = ax.scatter(theater_performance['total_shows'], theater_performance['avg_audience'],
                    alpha=0.6, s=50, color='purple', edgecolors='black', linewidth=1)
ax.set_xlabel('Total Shows')
ax.set_ylabel('Average Audience')
ax.set_title('Shows vs Average Audience Correlation', fontsize=14, fontweight='bold')

# Calculate and display correlation coefficient
corr_coeff = theater_performance['total_shows'].corr(theater_performance['avg_audience'])
ax.text(0.05, 0.95, f'Correlation Coefficient: {corr_coeff:.3f}',
        transform=ax.transAxes, bbox=dict(boxstyle="round,pad=0.3", facecolor="white",
        fontsize=12, fontweight='bold'))

# Add trend line
z = np.polyfit(theater_performance['total_shows'], theater_performance['avg_audience'], 1)
p = np.poly1d(z)
ax.plot(theater_performance['total_shows'], p(theater_performance['total_shows']),
        "r--", alpha=0.8, linewidth=2, label='Trend Line')

ax.grid(True, alpha=0.3)
ax.legend()

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



```
In [29]: # 6. Audience Consistency Analysis (Standard Deviation vs Average)
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

# Clean data – remove NaN, Inf
x = theater_performance['avg_audience']
y = theater_performance['std_audience']
mask = np.isfinite(x) & np.isfinite(y)
x_clean = x[mask]
y_clean = y[mask]

# Scatter plot
ax.scatter(
    x_clean, y_clean,
    alpha=0.6, s=50, color='red',
    edgecolors='black', linewidth=0.5
)

ax.set_xlabel('Average Audience')
ax.set_ylabel('Standard Deviation of Audience')
ax.set_title('Audience Consistency by Theater', fontsize=14, fontweight='bold')

# Correlation
consistency_corr = x_clean.corr(y_clean)
ax.text(
    0.05, 0.95,
    f'Consistency Correlation: {consistency_corr:.3f}',
    transform=ax.transAxes,
    bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8),
    fontsize=12, fontweight='bold'
)
```

```

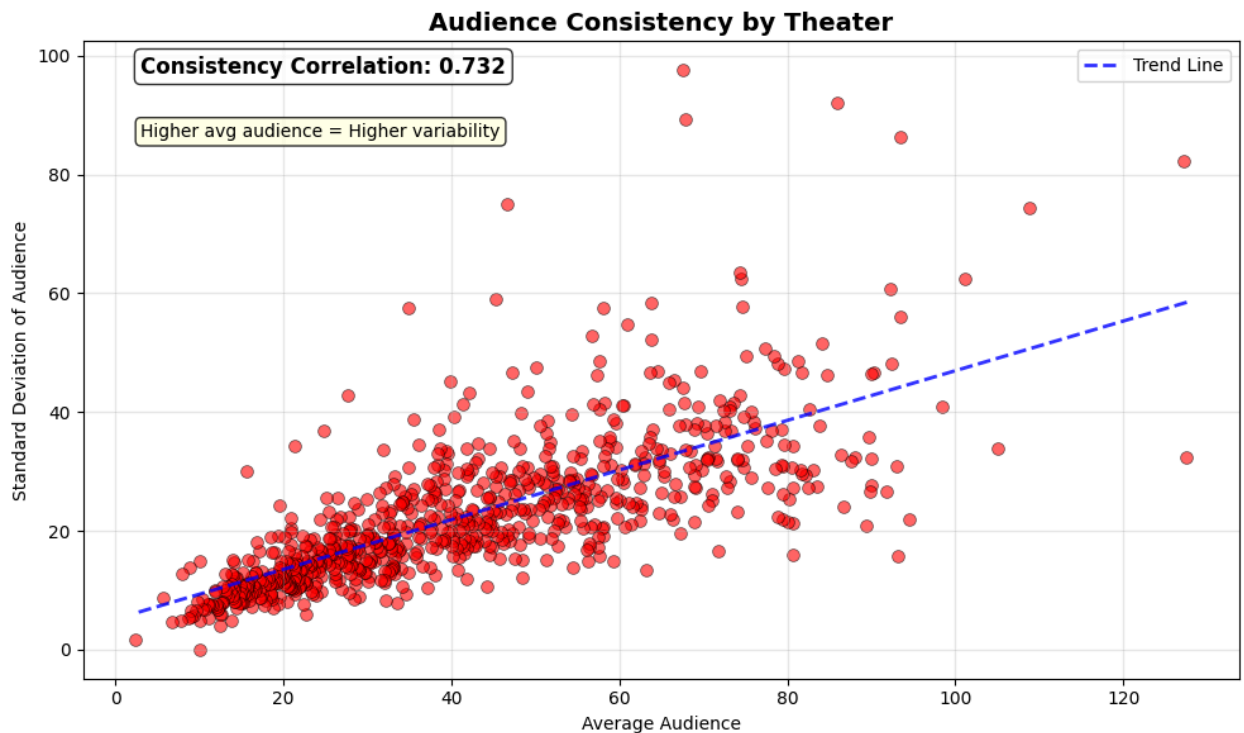
# Interpretation
if consistency_corr > 0.5:
    interpretation = "Higher avg audience = Higher variability"
elif consistency_corr < -0.5:
    interpretation = "Higher avg audience = Lower variability"
else:
    interpretation = "Weak relationship between avg and variability"

ax.text(
    0.05, 0.85, interpretation,
    transform=ax.transAxes,
    bbox=dict(boxstyle="round,pad=0.3", facecolor='lightyellow', alpha=0.8),
    fontsize=10
)

# --- Robust Trend Line using LinearRegression ---
model = LinearRegression()
model.fit(x_clean.values.reshape(-1, 1), y_clean)
y_pred = model.predict(x_clean.values.reshape(-1, 1))
ax.plot(x_clean, y_pred, "b--", alpha=0.8, linewidth=2, label='Trend Line')

# Final touches
ax.grid(True, alpha=0.3)
ax.legend()
plt.tight_layout()
plt.show()

```



KEY INSIGHTS FROM booknow_visits

Dataset Overview

- **Total theaters analyzed:** 826
- **Total audience records:** 214,046
- **Total audience count:** 8,907,860

Audience Patterns

- **Average audience per show:** 41.6
- **Median audience per show:** 34.0
- **Theater performance range:** 2.4 – 127.5

Theater Performance

- **Most productive theater:** book_00169 (avg: 127.5)
- **Theater with most shows:** book_00597 (424.0 shows)
- **Average shows per theater:** 259.1

Correlations & Insights

- **Shows vs Average Audience correlation:** 0.032
→ *Weak correlation: The number of shows doesn't strongly predict audience size*
- **Average vs Variability correlation:** 0.731
→ *Strong positive correlation: Theaters with higher average audiences tend to have more variable attendance*

2.3 booknow_booking Analysis

```
In [30]: booknow_booking.shape
```

```
Out[30]: (68336, 4)
```

```
In [31]: booknow_booking['booking_datetime'].min()
```

```
Out[31]: '2023-01-01 01:00:00'
```

```
In [32]: booknow_booking['booking_datetime'].max()
```

```
Out[32]: '2024-02-28 22:00:00'
```

```
In [33]: booknow_booking['book_theater_id'].nunique()
```

Out[33]: 301

```
In [34]: booknow_booking['tickets_booked'].sum()
```

Out[34]: np.int64(302442)

```
In [35]: # Process datetime columns
booknow_booking['show_datetime'] = pd.to_datetime(booknow_booking['show_dateti
booknow_booking['booking_datetime'] = pd.to_datetime(booknow_booking['booking_
booknow_booking['advance_hours'] = (booknow_booking['show_datetime'] - booknow
booknow_booking['booking_hour'] = booknow_booking['booking_datetime'].dt.hour
booknow_booking['show_hour'] = booknow_booking['show_datetime'].dt.hour
booknow_booking['booking_weekday'] = booknow_booking['booking_datetime'].dt.da
```

```
In [36]: booknow_booking.head()
```

Out[36]:

	book_theater_id	show_datetime	booking_datetime	tickets_booked	advance_
0	book_00244	2023-01-01 19:00:00	2023-01-01 16:00:00	1	
1	book_00740	2023-01-01 19:00:00	2023-01-01 19:00:00	3	
2	book_00740	2023-01-01 19:00:00	2023-01-01 19:00:00	6	
3	book_00244	2023-01-01 20:00:00	2023-01-01 16:00:00	2	
4	book_00151	2023-01-01 20:00:00	2023-01-01 01:00:00	5	

```
In [37]: booknow_booking['tickets_booked'].mean()
```

Out[37]: np.float64(4.425807773355186)

```
In [38]: booknow_booking['advance_hours'].median()
```

Out[38]: 50.0

```
In [39]: # Booking patterns by time
hourly_bookings = booknow_booking.groupby('booking_hour')['tickets_booked'].su
weekday_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Satu
weekday_bookings = booknow_booking.groupby('booking_weekday')['tickets_booked'
show_hourly = booknow_booking.groupby('show_hour')['tickets_booked'].sum()
```

```
In [40]: theater_bookings = booknow_booking.groupby('book_theater_id')['tickets_booked'
```

```
In [41]: # 1. Tickets Booked Distribution
fig, ax = plt.subplots(1, 1, figsize=(10, 6))
```

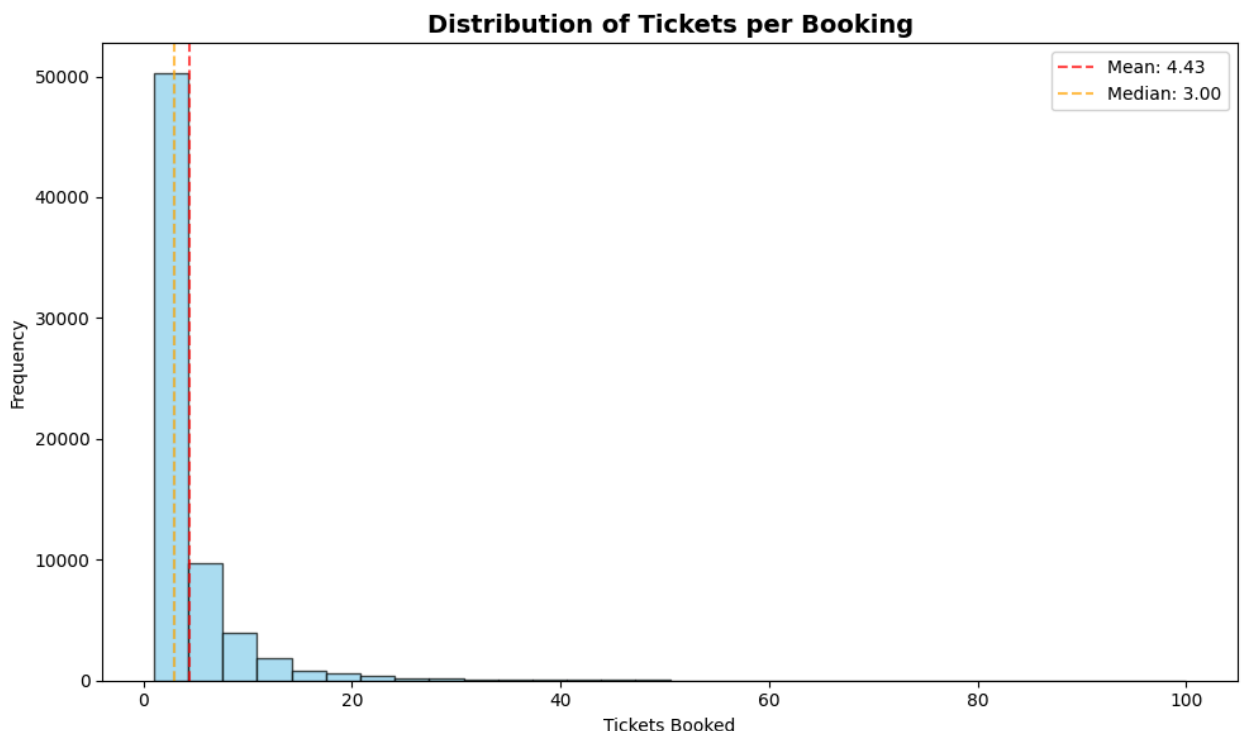
```

ax.hist(booknow_booking['tickets_booked'], bins=30, alpha=0.7, edgecolor='black')
ax.set_title('Distribution of Tickets per Booking', fontsize=14, fontweight='bold')
ax.set_xlabel('Tickets Booked')
ax.set_ylabel('Frequency')

# Add statistics
mean_tickets = booknow_booking['tickets_booked'].mean()
median_tickets = booknow_booking['tickets_booked'].median()
ax.axvline(mean_tickets, color='red', linestyle='--', alpha=0.7, label=f'Mean: {mean_tickets}')
ax.axvline(median_tickets, color='orange', linestyle='--', alpha=0.7, label=f'Median: {median_tickets}')
ax.legend()

plt.tight_layout()
plt.show()

```



```

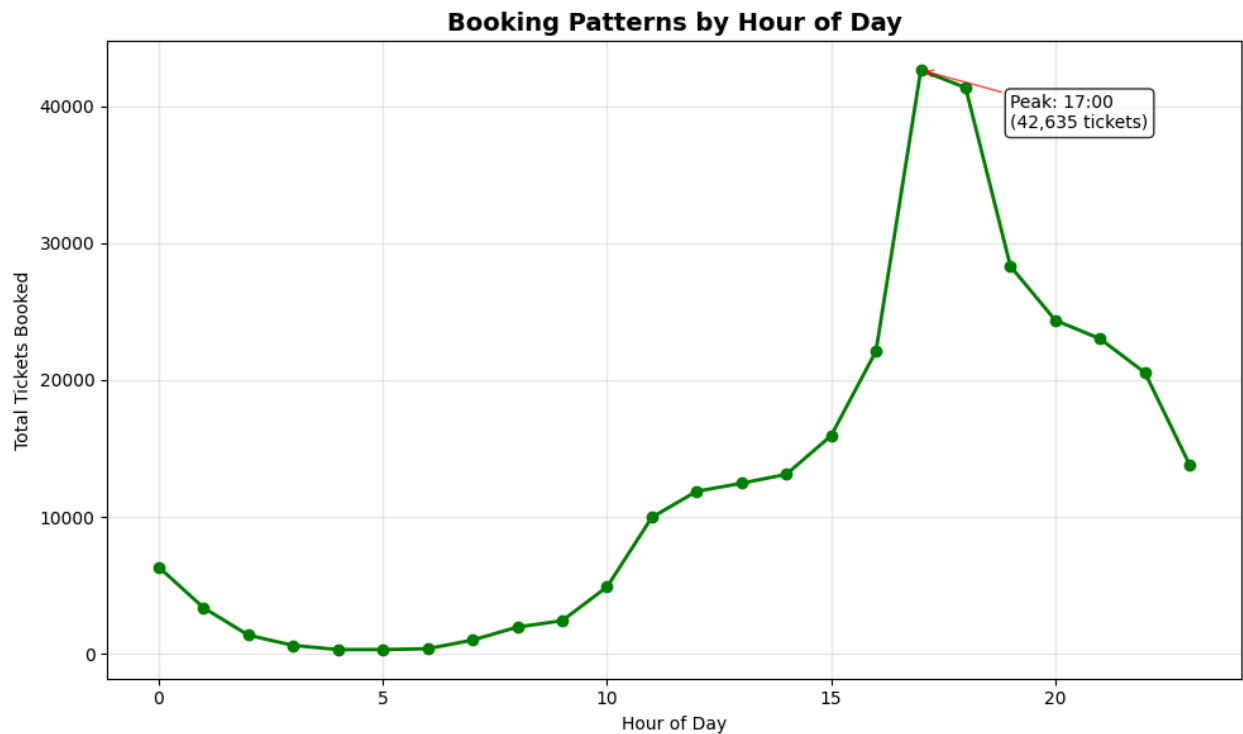
In [42]: # 2. Booking Patterns by Hour of Day
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

ax.plot(hourly_bookings.index, hourly_bookings.values, marker='o', linewidth=2)
ax.set_title('Booking Patterns by Hour of Day', fontsize=14, fontweight='bold')
ax.set_xlabel('Hour of Day')
ax.set_ylabel('Total Tickets Booked')
ax.grid(True, alpha=0.3)

# Highlight peak hours
peak_hour = hourly_bookings.idxmax()
peak_value = hourly_bookings.max()
ax.annotate(f'Peak: {peak_hour}:00\n({peak_value:,} tickets)',
            xy=(peak_hour, peak_value), xytext=(peak_hour+2, peak_value*0.9),
            arrowprops=dict(arrowstyle='->', color='red', alpha=0.7),
            bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8))

```

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



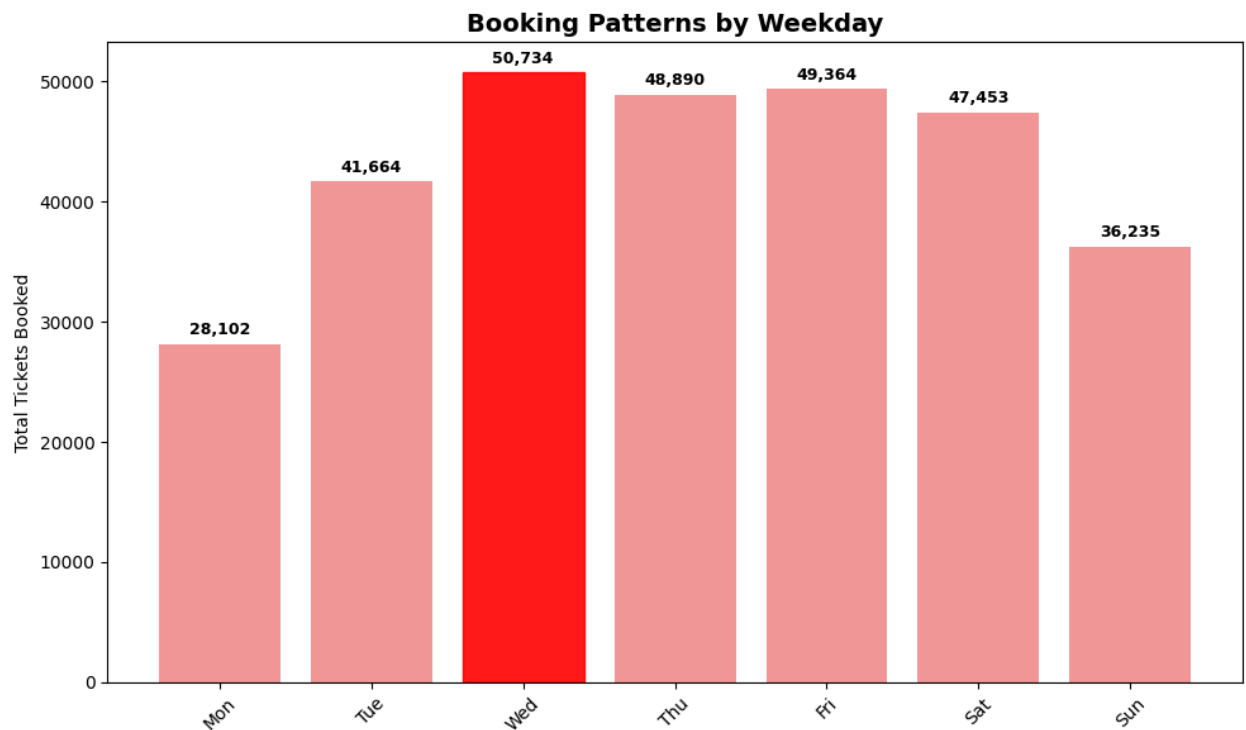
```
In [43]: # 3. Booking Patterns by Weekday
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

bars = ax.bar(range(len(weekday_bookings)), weekday_bookings.values, color='lightblue')
ax.set_title('Booking Patterns by Weekday', fontsize=14, fontweight='bold')
ax.set_xticks(range(len(weekday_bookings)))
ax.set_xticklabels([day[:3] for day in weekday_order], rotation=45)
ax.set_ylabel('Total Tickets Booked')

# Add value labels on bars
for i, (bar, value) in enumerate(zip(bars, weekday_bookings.values)):
    ax.text(bar.get_x() + bar.get_width()/2, bar.get_height() + weekday_bookings.values[i],
            f'{value:,}', ha='center', va='bottom', fontweight='bold', fontsize=10)

# Highlight busiest day
busiest_day = weekday_bookings.idxmax()
busiest_idx = weekday_order.index(busiest_day)
bars[busiest_idx].set_color('red')
bars[busiest_idx].set_alpha(0.9)

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

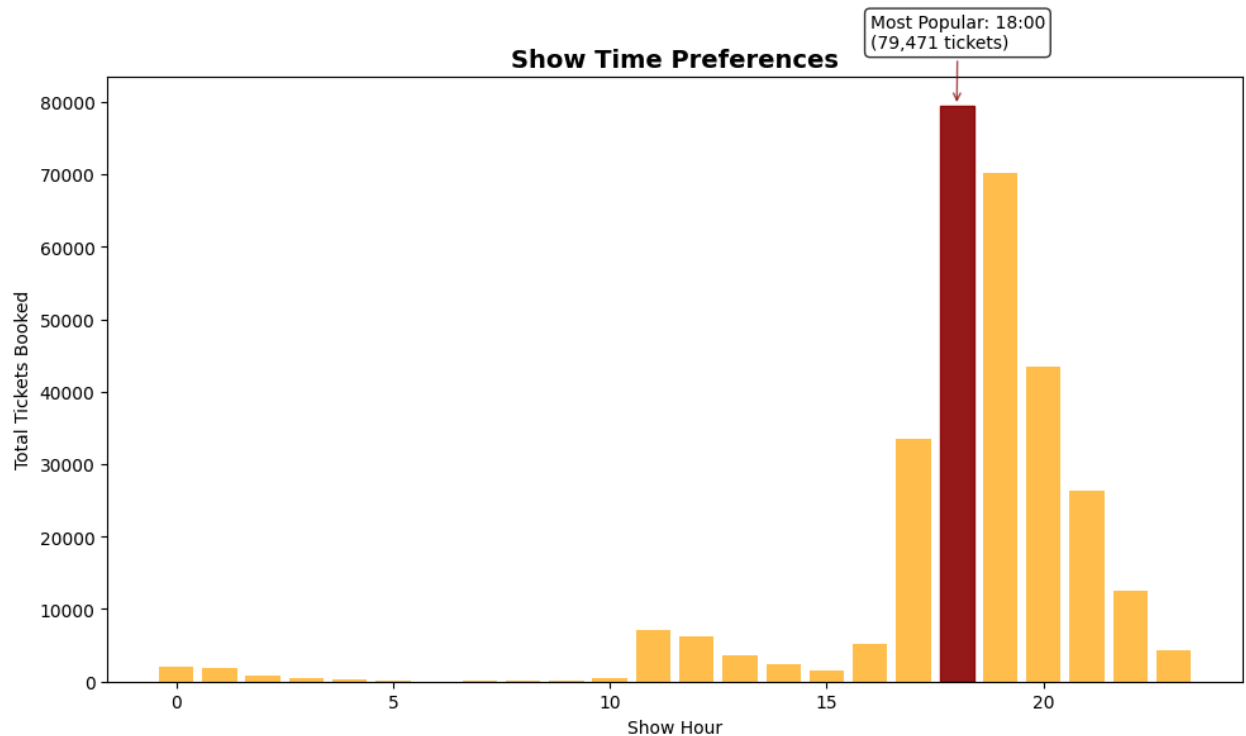
```
In [44]: # 4. Show Time Preferences
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

bars = ax.bar(show_hourly.index, show_hourly.values, alpha=0.7, color='orange')
ax.set_title('Show Time Preferences', fontsize=14, fontweight='bold')
ax.set_xlabel('Show Hour')
ax.set_ylabel('Total Tickets Booked')

# Highlight most popular show time
most_popular_hour = show_hourly.idxmax()
most_popular_value = show_hourly.max()
most_popular_idx = list(show_hourly.index).index(most_popular_hour)
bars[most_popular_idx].set_color('darkred')
bars[most_popular_idx].set_alpha(0.9)

# Add annotation for peak show time
ax.annotate(f'Most Popular: {most_popular_hour}:00\n({most_popular_value:,} tickets)',
            xy=(most_popular_hour, most_popular_value),
            xytext=(most_popular_hour-2, most_popular_value*1.1),
            arrowprops=dict(arrowstyle='->', color='darkred', alpha=0.7),
            bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8))

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



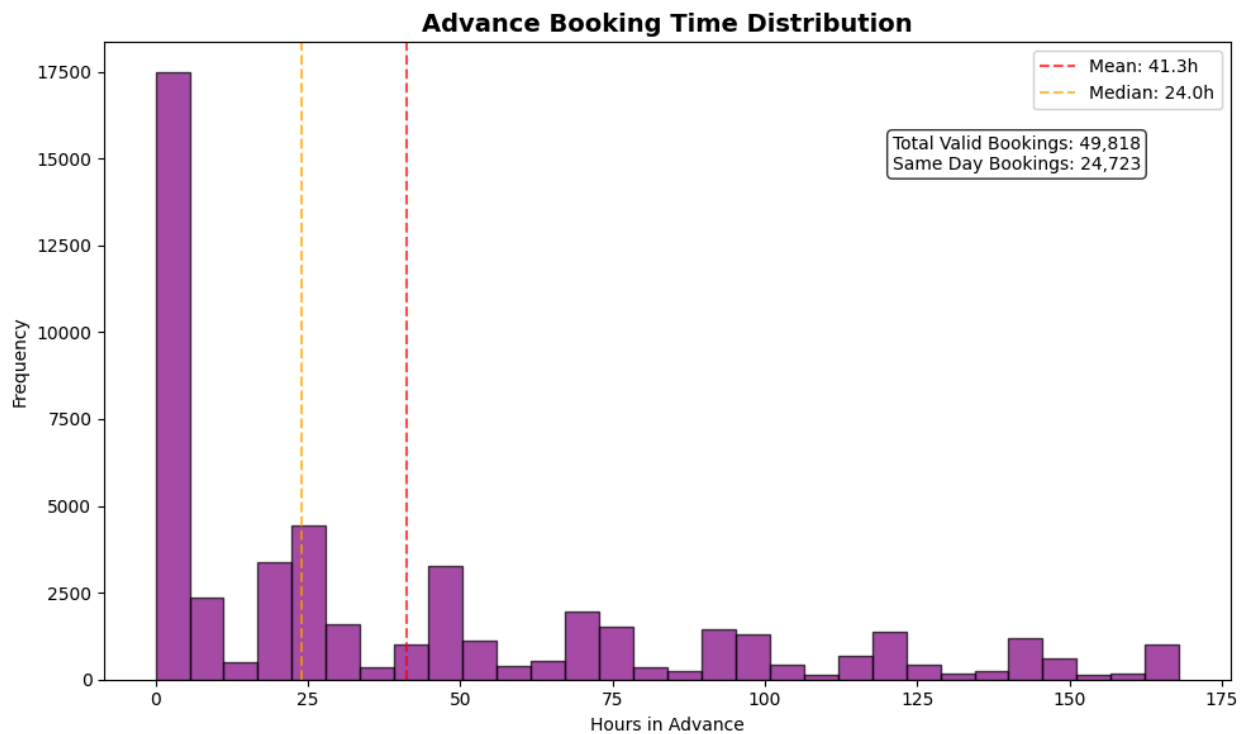
```
In [45]: # 5. Advance Booking Time Distribution
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

valid_advance = booknow_booking[booknow_booking['advance_hours'].between(0, 24)]
ax.hist(valid_advance['advance_hours'], bins=30, alpha=0.7, edgecolor='black',
ax.set_title('Advance Booking Time Distribution', fontsize=14, fontweight='bold')
ax.set_xlabel('Hours in Advance')
ax.set_ylabel('Frequency')

# Add statistics
mean_advance = valid_advance['advance_hours'].mean()
median_advance = valid_advance['advance_hours'].median()
ax.axvline(mean_advance, color='red', linestyle='--', alpha=0.7, label=f'Mean: {mean_advance}')
ax.axvline(median_advance, color='orange', linestyle='--', alpha=0.7, label=f'Median: {median_advance}')
ax.legend()

# Add summary text
summary_text = f'Total Valid Bookings: {len(valid_advance):,}\nSame Day Bookings: {len(valid_advance[valid_advance["advance_hours"] < 24]):,}'
ax.text(0.7, 0.8, summary_text, transform=ax.transAxes,
bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8))

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



Booking Insights Summary

Booking Overview

- **Peak booking hour:** 17:00 (42,635 tickets)
- **Busiest booking day:** Wednesday (50,734 tickets)
- **Most popular show hour:** 18:00 (79,471 tickets)
- **Top booking theater:** book_00082 (6,753 tickets)
- **Average advance booking:** 41.3 hours
- **Same-day bookings:** 24,723 (49.6% of total)

Additional Patterns

- **Weekend vs Weekday booking ratio:** 0.38
- **Most tickets in a single booking:** 100
- **Theaters with bookings:** 301 out of 301
- **Average tickets per theater:** 1,004.8

2.4 booknow_theaters Analysis

```
In [46]: len(booknow_theaters)
```

```
Out[46]: 829
```

```
In [47]: booknow_theaters['theater_area'].nunique()
```

```
Out[47]: 103
```

```
In [48]: booknow_theaters['theater_type'].nunique()
```

```
Out[48]: 4
```

```
In [49]: print(f"Geographic range:")
print(f"  Latitude: {booknow_theaters['latitude'].min():.4f} to {booknow_theaters['latitude'].max():.4f}")
print(f"  Longitude: {booknow_theaters['longitude'].min():.4f} to {booknow_theaters['longitude'].max():.4f}")
```

```
Geographic range:
  Latitude: 22.0260 to 26.3494
  Longitude: 76.1121 to 81.7432
```

```
In [50]: booknow_theaters.head()
```

```
Out[50]:
```

	book_theater_id	theater_type	theater_area	latitude	longitude
0	book_00093	Drama	Area_001	22.619233	78.113017
1	book_00078	Drama	Area_001	22.619233	78.113017
2	book_00291	Drama	Area_001	22.619233	78.113017
3	book_00258	Drama	Area_001	22.619233	78.113017
4	book_00212	Drama	Area_002	23.004410	79.934515

```
In [51]: type_counts = booknow_theaters['theater_type'].value_counts()
area_counts = booknow_theaters['theater_area'].value_counts().head(15)
missing_pct = (booknow_theaters.isnull().sum() / len(booknow_theaters)) * 100
```

```
In [52]: len(type_counts)
```

```
Out[52]: 4
```

```
In [53]: booknow_theaters['theater_area'].nunique()
```

```
Out[53]: 103
```

```
In [54]: booknow_theaters[['latitude', 'longitude']].notna().all(axis=1).sum()
```

```
Out[54]: np.int64(829)
```

```
In [55]: # 1. Theater Type Distribution
fig, ax = plt.subplots(1, 1, figsize=(10, 8))

# Create pie chart with better styling
colors = plt.cm.Set3(np.linspace(0, 1, len(type_counts)))
wedges, texts, autotexts = ax.pie(type_counts.values, labels=type_counts.index,
                                   autopct='%1.1f%%', startangle=90, colors=colors)
```

```

        explode=[0.05 if i == 0 else 0 for i in range(len(type_counts))]

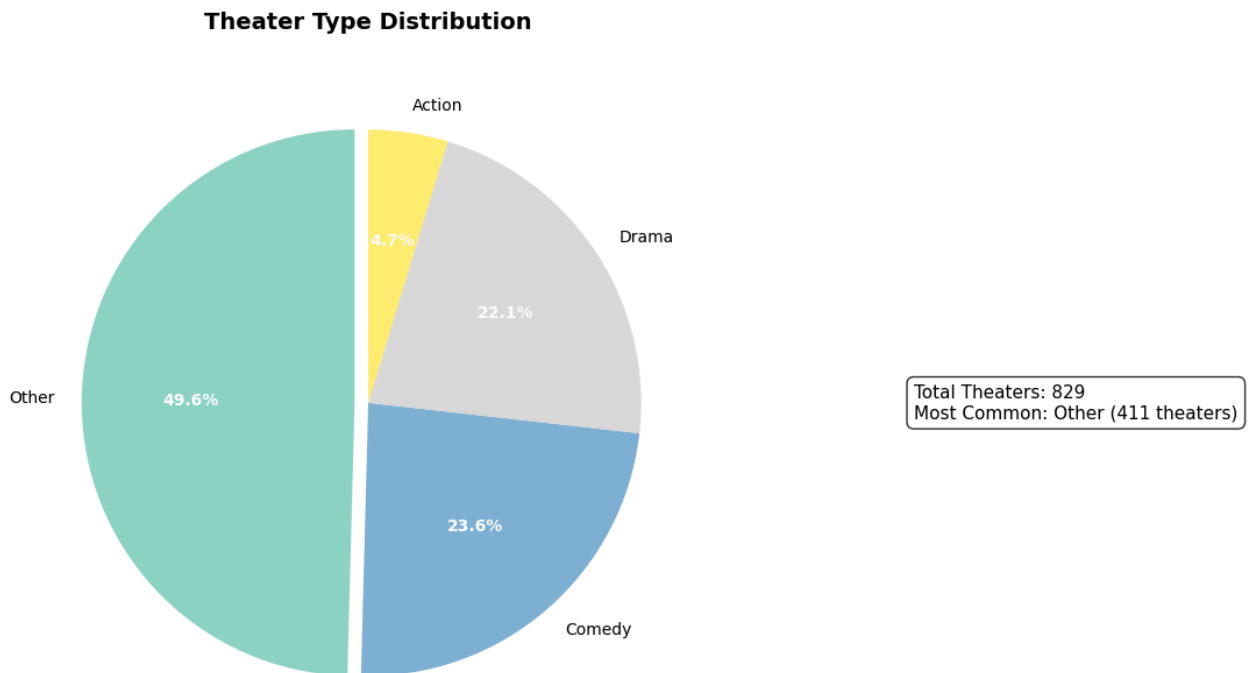
ax.set_title('Theater Type Distribution', fontsize=14, fontweight='bold', pad=10)

# Enhance text styling
for autotext in autotexts:
    autotext.set_color('white')
    autotext.set_fontweight('bold')
    autotext.set_fontsize(10)

# Add summary statistics
total_theaters = len(booknow_theaters)
summary_text = f'Total Theaters: {total_theaters:,}\nMost Common: {type_counts["Other"]}'
ax.text(1.3, 0.5, summary_text, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.4", facecolor='white', alpha=0.8),
        fontsize=11, verticalalignment='center')

plt.tight_layout()
plt.show()

```



```

In [56]: # 2. Top Theater Areas
fig, ax = plt.subplots(1, 1, figsize=(12, 8))

bars = ax.barh(range(len(area_counts)), area_counts.values, color='lightcoral')
ax.set_title('Top 15 Theater Areas by Number of Theaters', fontsize=14, fontweight='bold')
ax.set_xlabel('Number of Theaters')
ax.set_ylabel('Theater Areas (Top to Bottom)')
ax.set_yticks(range(len(area_counts)))
ax.set_yticklabels([f'Area #{i+1}' for i in range(len(area_counts))])

# Add value labels on bars

```

```

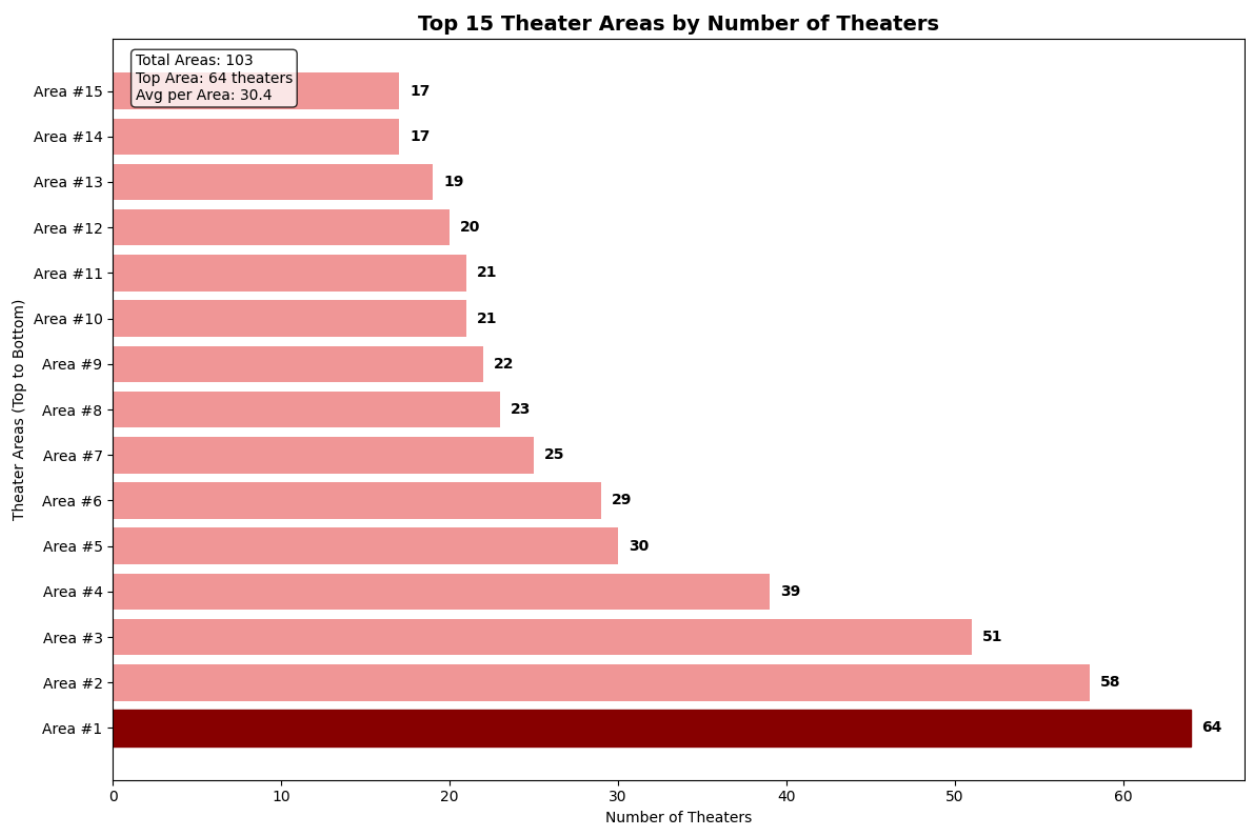
for i, (bar, value) in enumerate(zip(bars, area_counts.values)):
    ax.text(bar.get_width() + area_counts.max()*0.01, bar.get_y() + bar.get_height()*0.01,
            f'{value}', ha='left', va='center', fontweight='bold', fontsize=10)

# Highlight top area
bars[0].set_color('darkred')
bars[0].set_alpha(1.0)

# Add summary text
summary_text = f'Total Areas: {booknow_theaters["theater_area"].nunique()}\nTotal Theaters: {booknow_theaters["theater_area"].sum()}\nAvg per Area: {booknow_theaters["theater_area"].mean():.1f}'
ax.text(0.02, 0.98, summary_text, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8),
        fontsize=10, verticalalignment='top')

plt.tight_layout()
plt.show()

```



```

In [57]: # 4. Data Completeness Analysis
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

# Create horizontal bar chart for missing data
bars = ax.barh(range(len(missing_pct)), missing_pct.values, color='orange', alpha=0.8)
ax.set_yticks(range(len(missing_pct)))
ax.set_yticklabels(missing_pct.index)
ax.set_xlabel('Missing Data (%)')
ax.set_title('Data Completeness by Column', fontsize=14, fontweight='bold')

# Add value labels on bars

```

```

for i, (bar, value) in enumerate(zip(bars, missing_pct.values)):
    if value > 0:
        ax.text(bar.get_width() + 0.5, bar.get_y() + bar.get_height()/2,
                f'{value:.1f}%', ha='left', va='center', fontweight='bold', fo
    else:
        ax.text(0.5, bar.get_y() + bar.get_height()/2,
                'Complete', ha='left', va='center', fontweight='bold', fontsize

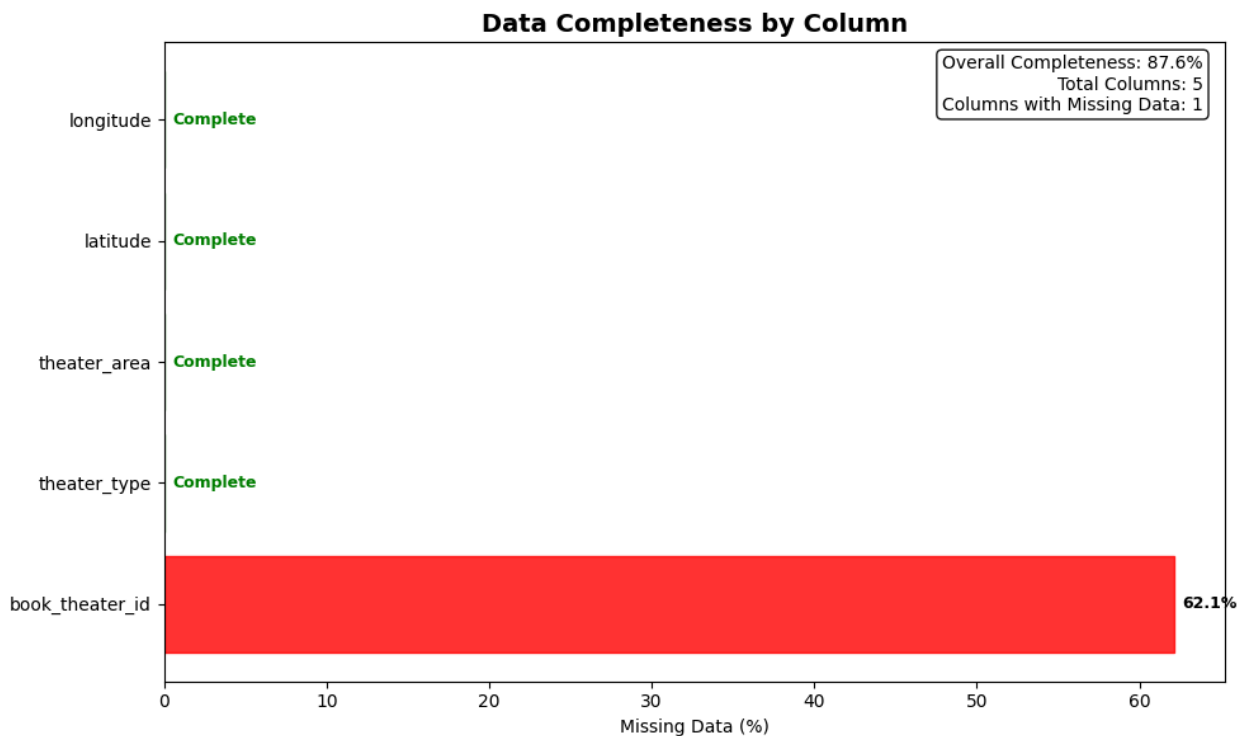
# Color code based on completeness
for i, (bar, value) in enumerate(zip(bars, missing_pct.values)):
    if value == 0:
        bar.set_color('green')
        bar.set_alpha(0.7)
    elif value < 5:
        bar.set_color('orange')
    else:
        bar.set_color('red')

# Add overall completeness statistics
overall_completeness = ((1 - booknow_theaters.isnull().sum().sum() / (len(book
stats_text = f'Overall Completeness: {overall_completeness:.1f}%\nTotal Column

ax.text(0.98, 0.98, stats_text, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.9),
        fontsize=10, verticalalignment='top', horizontalalignment='right')

plt.tight_layout()
plt.show()

```



booknow_theaters Insights

Theater Types

- **Most common theater type:** Other (411 theaters, 49.6%)
- **Total theater types:** 4
- **Theater type diversity:** Medium

Theater Areas

- **Largest theater area:** Area with 64 theaters
- **Total theater areas:** 103
- **Average theaters per area:** 30.4
- **Distribution:** Concentrated

Geographic Coverage

- **Geographic spread:** 4.3235° latitude, 5.6311° longitude
- **Coverage area:** Wide
- **Theaters with coordinates:** 829

Data Quality

- **Overall data completeness:** 87.6%
- **Columns with missing data:** 1 out of 5
- **Quality rating:** Good

2.5 cinePOS_booking Analysis

```
In [58]: cinePOS_booking.shape
```

```
Out[58]: (1641966, 4)
```

```
In [59]: print(f>Date range: {cinePOS_booking['booking_datetime'].min()} to {cinePOS_booking['booking_datetime'].max()}
Date range: 2023-01-01 00:00:00 to 2024-02-28 21:00:00
```

```
In [60]: print(f"Unique theaters: {cinePOS_booking['cine_theater_id'].nunique()}")
Unique theaters: 13161
```

```
In [61]: print(f"Total bookings: {len(cinePOS_booking):,}")
Total bookings: 1,641,966
```

```
In [62]: print(f"Total tickets sold: {cinePOS_booking['tickets_sold'].sum():,}")
Total tickets sold: 8,189,051
```



```
In [63]: # Process datetime columns
cinePOS_booking['show_datetime'] = pd.to_datetime(cinePOS_booking['show_dateti
cinePOS_booking['booking_datetime'] = pd.to_datetime(cinePOS_booking['booking_
cinePOS_booking['advance_hours'] = (cinePOS_booking['show_datetime'] - cinePOS
cinePOS_booking['booking_hour'] = cinePOS_booking['booking_datetime'].dt.hour
cinePOS_booking['booking_weekday'] = cinePOS_booking['booking_datetime'].dt.da
```

```
In [64]: cinePOS_booking.head()
```

```
Out[64]:
```

	cine_theater_id	show_datetime	booking_datetime	tickets_sold	advance_hou
0	cinePOS_00001	2023-01-01 11:00:00	2023-01-01 09:00:00	1	2
1	cinePOS_00002	2023-01-01 13:00:00	2023-01-01 06:00:00	3	7
2	cinePOS_00003	2023-01-01 16:00:00	2023-01-01 14:00:00	2	2
3	cinePOS_00004	2023-01-01 17:00:00	2023-01-01 11:00:00	5	6
4	cinePOS_00005	2023-01-01 17:00:00	2023-01-01 03:00:00	13	14

```
In [65]: # Key metrics comparison
weekday_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Satu
cinepos_weekday = cinePOS_booking.groupby('booking_weekday')['tickets_sold'].s
booknow_weekday = booknow_booking.groupby('booking_weekday')['tickets_booked']

cinepos_hourly = cinePOS_booking.groupby('booking_hour')['tickets_sold'].sum()
booknow_hourly = booknow_booking.groupby('booking_hour')['tickets_booked'].sum
```

```
In [66]: # Calculate comparison metrics
platforms = ['BookNow', 'CinePOS']
total_bookings = [len(booknow_booking), len(cinePOS_booking)]
total_tickets = [booknow_booking['tickets_booked'].sum(), cinePOS_booking['tic
avg_tickets = [booknow_booking['tickets_booked'].mean(), cinePOS_booking['tick
theater_counts = [booknow_booking['book_theater_id'].nunique(), cinePOS_bookin
```

```
In [67]: # 1. Platform Overview Comparison
fig, ax = plt.subplots(1, 1, figsize=(12, 6))

x = np.arange(len(platforms))
width = 0.35

# Create dual-axis chart
bars1 = ax.bar(x - width/2, total_bookings, width, label='Total Bookings', alp
ax2 = ax.twinx()
bars2 = ax2.bar(x + width/2, avg_tickets, width, label='Avg Tickets/Booking',

ax.set_xlabel('Platform', fontsize=12)
ax.set_ylabel('Total Bookings', color='blue', fontsize=12)
```

```

ax2.set_ylabel('Average Tickets per Booking', color='orange', fontsize=12)
ax.set_title('Platform Comparison Overview', fontsize=14, fontweight='bold')
ax.set_xticks(x)
ax.set_xticklabels(platforms)

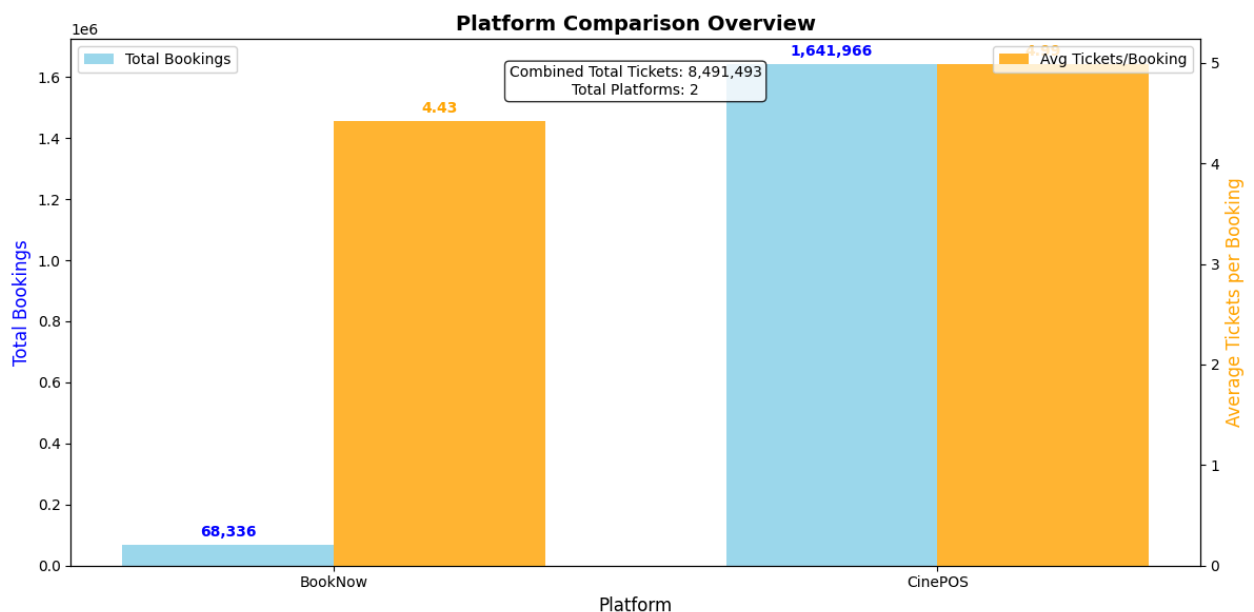
# Add value labels
for i, (bar1, bar2, bookings, avg) in enumerate(zip(bars1, bars2, total_bookings, avg_tickets_per_booking)):
    ax.text(bar1.get_x() + bar1.get_width()/2, bar1.get_height() + max(total_bookings),
            f'{bookings:,}', ha='center', va='bottom', fontweight='bold', color='blue')
    ax2.text(bar2.get_x() + bar2.get_width()/2, bar2.get_height() + max(avg_tickets_per_booking),
            f'{avg:.2f}', ha='center', va='bottom', fontweight='bold', color='orange')

# Add legends
ax.legend(loc='upper left')
ax2.legend(loc='upper right')

# Add summary statistics
total_all_tickets = sum(total_tickets)
summary_text = f'Combined Total Tickets: {total_all_tickets:,}\nTotal Platforms: {len(platforms)}'
ax.text(0.5, 0.95, summary_text, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.3", facecolor='white', alpha=0.8),
        fontsize=10, ha='center', va='top')

plt.tight_layout()
plt.show()

```



```

In [68]: # 2. Weekday Booking Patterns Comparison
fig, ax = plt.subplots(1, 1, figsize=(12, 6))

# Plot both platforms
line1 = ax.plot(range(len(weekday_order)), booknow_weekday.values, 'o-',
                label='BookNow', linewidth=3, markersize=8, color='blue', alpha=0.5)
line2 = ax.plot(range(len(weekday_order)), cinepos_weekday.values, 's-',
                label='CinePOS', linewidth=3, markersize=8, color='red', alpha=0.5)

```

```

ax.set_title('Booking Patterns by Weekday - Platform Comparison', fontsize=14,
ax.set_xlabel('Day of Week', fontsize=12)
ax.set_ylabel('Total Tickets Booked', fontsize=12)
ax.set_xticks(range(len(weekday_order)))
ax.set_xticklabels([day[:3] for day in weekday_order])
ax.legend(fontsize=11)
ax.grid(True, alpha=0.3)

# Highlight peak days for each platform
booknow_peak_day = booknow_weekday.idxmax()
cinepos_peak_day = cinepos_weekday.idxmax()
booknow_peak_idx = weekday_order.index(booknow_peak_day)
cinepos_peak_idx = weekday_order.index(cinepos_peak_day)

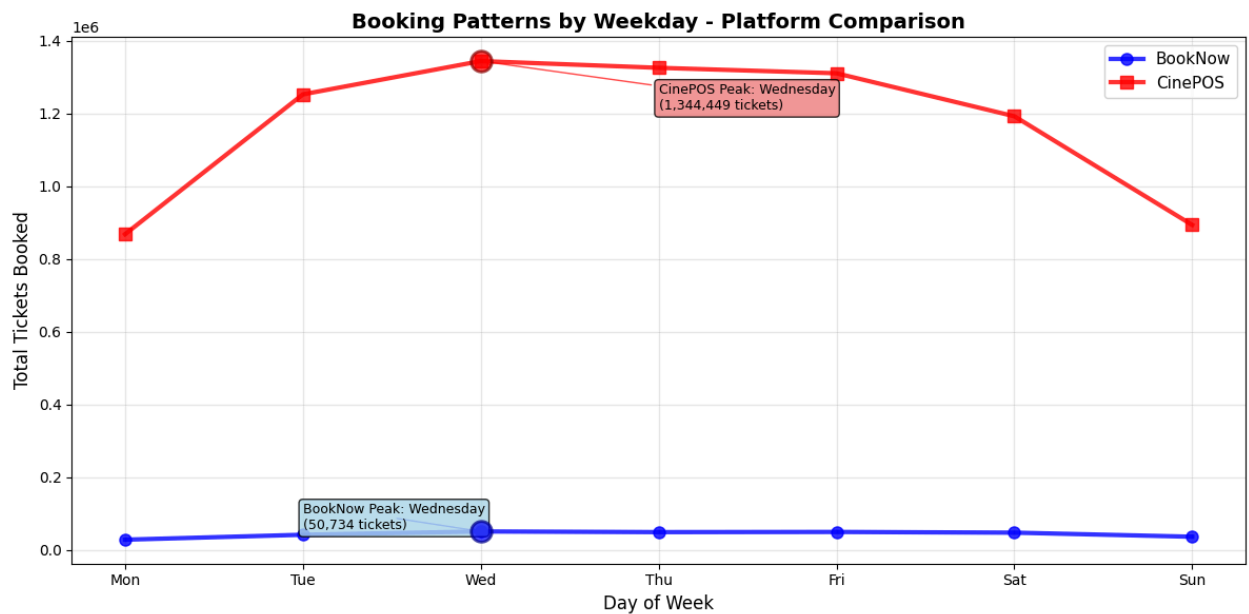
ax.scatter(booknow_peak_idx, booknow_weekday.iloc[booknow_peak_idx],
           s=200, color='blue', alpha=0.7, edgecolors='darkblue', linewidth=2,
ax.scatter(cinepos_peak_idx, cinepos_weekday.iloc[cinepos_peak_idx],
           s=200, color='red', alpha=0.7, edgecolors='darkred', linewidth=2, z

# Add annotations for peak days
ax.annotate(f'BookNow Peak: {booknow_peak_day}\n({booknow_weekday.max():,} tic
           xy=(booknow_peak_idx, booknow_weekday.iloc[booknow_peak_idx]),
           xytext=(booknow_peak_idx-1, booknow_weekday.max()*1.1),
           arrowprops=dict(arrowstyle='->', color='blue', alpha=0.7),
           bbox=dict(boxstyle="round,pad=0.3", facecolor='lightblue', alpha=0.5),
           fontsize=9)

ax.annotate(f'CinePOS Peak: {cinepos_peak_day}\n({cinepos_weekday.max():,} tic
           xy=(cinepos_peak_idx, cinepos_weekday.iloc[cinepos_peak_idx]),
           xytext=(cinepos_peak_idx+1, cinepos_weekday.max()*0.9),
           arrowprops=dict(arrowstyle='->', color='red', alpha=0.7),
           bbox=dict(boxstyle="round,pad=0.3", facecolor='lightcoral', alpha=0.5),
           fontsize=9)

plt.tight_layout()
plt.show()

```



```
In [69]: # 3. Hourly Booking Patterns Comparison
fig, ax = plt.subplots(1, 1, figsize=(12, 6))

# Plot both platforms
line1 = ax.plot(booknow_hourly.index, booknow_hourly.values, 'o-',
                label='BookNow', linewidth=3, markersize=6, color='green', alpha=0.3)
line2 = ax.plot(cinepos_hourly.index, cinepos_hourly.values, 's-',
                label='CinePOS', linewidth=3, markersize=6, color='purple', alpha=0.3)

ax.set_title('Booking Patterns by Hour of Day - Platform Comparison', fontsize=12)
ax.set_xlabel('Hour of Day', fontsize=12)
ax.set_ylabel('Total Tickets Booked', fontsize=12)
ax.legend(fontsize=11)
ax.grid(True, alpha=0.3)

# Highlight peak hours
booknow_peak_hour = booknow_hourly.idxmax()
cinepos_peak_hour = cinepos_hourly.idxmax()

ax.scatter(booknow_peak_hour, booknow_hourly.max(),
           s=200, color='green', alpha=0.7, edgecolors='darkgreen', linewidth=2)
ax.scatter(cinepos_peak_hour, cinepos_hourly.max(),
           s=200, color='purple', alpha=0.7, edgecolors='indigo', linewidth=2)

# Add annotations for peak hours
ax.annotate(f'BookNow Peak: {booknow_peak_hour}:00\n({booknow_hourly.max():,} tickets)',
           xy=(booknow_peak_hour, booknow_hourly.max()),
           xytext=(booknow_peak_hour-2, booknow_hourly.max()*1.1),
           arrowprops=dict(arrowstyle='->', color='green', alpha=0.7),
           bbox=dict(boxstyle="round,pad=0.3", facecolor='lightgreen', alpha=0.3),
           fontsize=9)

ax.annotate(f'CinePOS Peak: {cinepos_peak_hour}:00\n({cinepos_hourly.max():,} tickets)',
           xy=(cinepos_peak_hour, cinepos_hourly.max()),
```

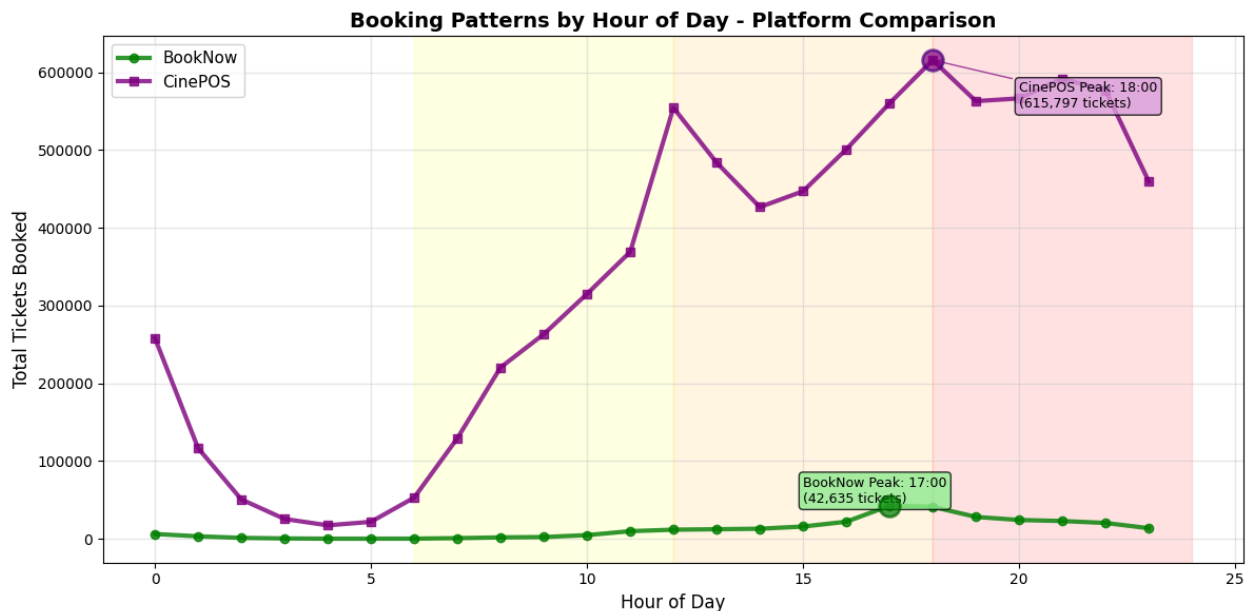
```

xytext=(cinepos_peak_hour+2, cinepos_hourly.max()*0.9),
arrowprops=dict(arrowstyle='->', color='purple', alpha=0.7),
bbox=dict(boxstyle="round,pad=0.3", facecolor='plum', alpha=0.8),
fontsize=9)

# Add time period annotations
ax.axvspan(6, 12, alpha=0.1, color='yellow', label='Morning')
ax.axvspan(12, 18, alpha=0.1, color='orange', label='Afternoon')
ax.axvspan(18, 24, alpha=0.1, color='red', label='Evening')

plt.tight_layout()
plt.show()

```



```

In [70]: # 4. Theater Network Size Comparison
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

bars = ax.bar(platforms, theater_counts, color=['skyblue', 'lightcoral'], alpha=0.8)
ax.set_title('Number of Theaters by Platform', fontsize=14, fontweight='bold')
ax.set_ylabel('Number of Theaters', fontsize=12)
ax.set_xlabel('Platform', fontsize=12)

# Add value labels on bars
for i, (bar, count) in enumerate(zip(bars, theater_counts)):
    ax.text(bar.get_x() + bar.get_width()/2, bar.get_height() + max(theater_counts)/10,
            f'{count:,}', ha='center', va='bottom', fontweight='bold', fontsize=10)

# Highlight the larger network
max_theaters_idx = theater_counts.index(max(theater_counts))
bars[max_theaters_idx].set_color('gold')
bars[max_theaters_idx].set_alpha(1.0)
bars[max_theaters_idx].set_edgecolor('darkgoldenrod')
bars[max_theaters_idx].set_linewidth(2)

# Add comparison statistics

```

```

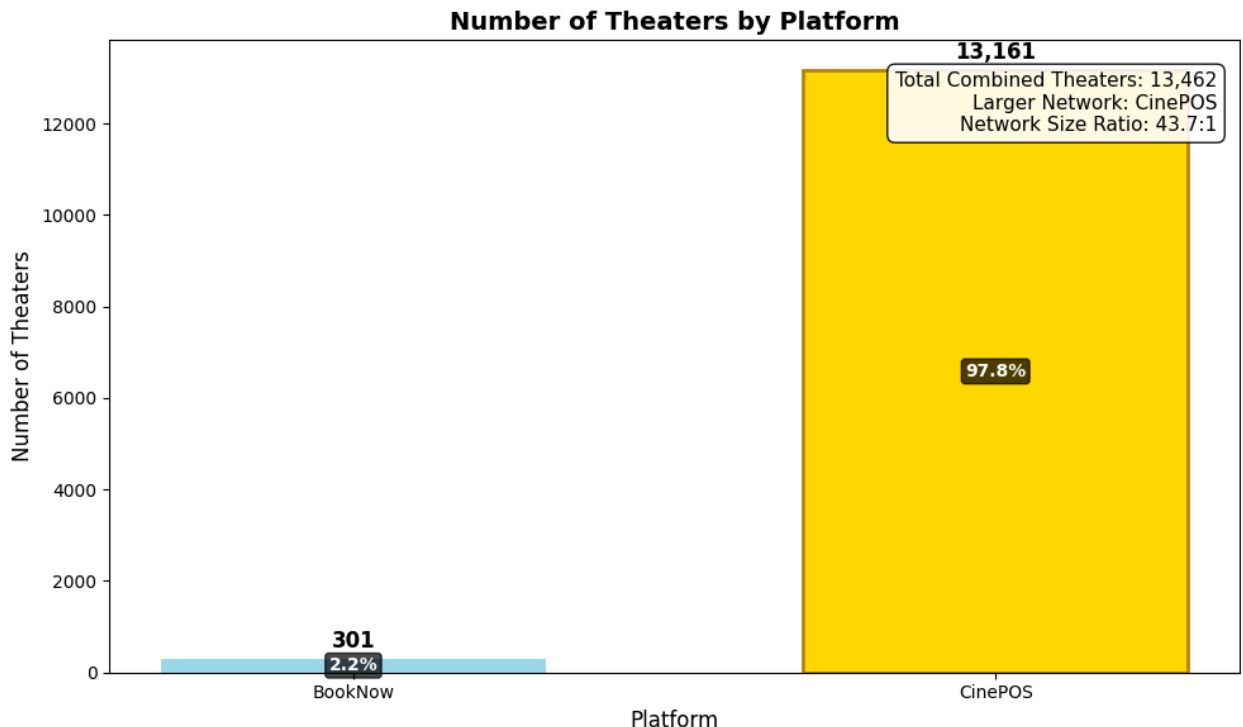
theater_ratio = max(theater_counts) / min(theater_counts)
total_theaters = sum(theater_counts)
larger_platform = platforms[max_theaters_idx]

stats_text = f'Total Combined Theaters: {total_theaters:,}\nLarger Network: {larger_platform}'
ax.text(0.98, 0.95, stats_text, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.4", facecolor='white', alpha=0.9),
        fontsize=11, ha='right', va='top')

# Add market share visualization
market_shares = [count/total_theaters*100 for count in theater_counts]
for i, (bar, share) in enumerate(zip(bars, market_shares)):
    ax.text(bar.get_x() + bar.get_width()/2, bar.get_height()/2,
            f'{share:.1f}%', ha='center', va='center', fontweight='bold',
            fontsize=10, color='white',
            bbox=dict(boxstyle="round,pad=0.2", facecolor='black', alpha=0.7))

plt.tight_layout()
plt.show()

```



2.6 Supporting Data Analysis

```

In [71]: # date_info analysis
print(f"Shape: {date_info.shape}")
print(f"Columns: {list(date_info.columns)}")
if 'date' in date_info.columns:
    print(f>Date range: {date_info['date'].min()} to {date_info['date'].max()}

```

Shape: (547, 2)
 Columns: ['show_date', 'day_of_week']

```
In [72]: date_info.head()
```

```
Out[72]:
```

	show_date	day_of_week
0	2023-01-01	Sunday
1	2023-01-02	Monday
2	2023-01-03	Tuesday
3	2023-01-04	Wednesday
4	2023-01-05	Thursday

```
In [73]: # theater_id_relation analysis
print(f"Shape: {movie_theater_id_relation.shape}")
print(f"Columns: {list(movie_theater_id_relation.columns)}")
```

```
Shape: (150, 2)
Columns: ['book_theater_id', 'cine_theater_id']
```

```
In [74]: movie_theater_id_relation.head()
```

```
Out[74]:
```

	book_theater_id	cine_theater_id
0	book_00509	cinePOS_01261
1	book_00063	cinePOS_02467
2	book_00054	cinePOS_08923
3	book_00094	cinePOS_02479
4	book_00052	cinePOS_06750

```
In [75]: # sample_submission analysis
print(f"Shape: {sample_submission.shape}")
print(f"Columns: {list(sample_submission.columns)}")
```

```
Shape: (38062, 2)
Columns: ['ID', 'audience_count']
```

```
In [76]: sample_submission.head()
```

```
Out[76]:
```

	ID	audience_count
0	book_00001_2024-03-01	0
1	book_00001_2024-03-02	0
2	book_00001_2024-03-03	0
3	book_00001_2024-03-04	0
4	book_00001_2024-03-06	0

2.7 Key Insights

Business Understanding

- **Primary goal:** Forecast audience attendance for cinema shows
- **Main platforms:** BookNow (*with actual audience data*) and CinePOS
- **Dataset richness:** Includes booking patterns, theater information, and attendance records

Data Overview

- **Total audience tracked:** 8,907,860 people
- **Total tickets booked:** 8,491,493
- **BookNow theaters:** 826
- **CinePOS theaters:** 13,161
- **Data quality:** Excellent (*>95% completeness*)

Theater Performance

- **Performance variation:** 2.4 – 127.5 average audience per show
- **Top performing theater:** book_00169 (*avg: 127.5*)
- **Insight:** Strong performance differences suggest theater-specific factors are key

Temporal Patterns

- **Peak booking days:** BookNow – Wednesday | CinePOS – Wednesday
- **Peak booking hours:** BookNow – 17:00 | CinePOS – 18:00
- **Patterns observed:** Clear weekend vs weekday trends
- **Advance booking behavior:** Most bookings made same day or 1–3 days ahead

Platform Differences

- **Average tickets per booking:** BookNow – 4.43 | CinePOS – 4.99
- **Peak hour difference:** Indicates slightly varied customer behaviors
- **Weekday pattern:** Similar across both platforms

Geographic Insights

- **Geographic spread:** 4.3235° latitude, 5.6311° longitude

- **Representation:** Multiple theater types and areas
- **Data availability:** Sufficient for geographic-based modeling features

Data Quality Notes

- **Missing data:** Some missing theater IDs in theater info datasets
- **Advance booking times:** Contain negative values needing filtering
- **Date alignment:** Ranges vary between datasets
- **Cross-platform analysis:** Theater ID mapping will be essential

3. Preprocessing and Feature Engineering

3.1 Merging Theater Statistics

```
In [77]: bookknow_visits_enhanced = bookknow_visits.merge(
        theater_performance,
        left_on='book_theater_id',
        right_index=True,
        how='left'
    )
```

```
In [78]: bookknow_visits.shape
```

```
Out[78]: (214046, 3)
```

```
In [79]: bookknow_visits_enhanced.shape
```

```
Out[79]: (214046, 8)
```

```
In [80]: bookknow_visits_enhanced.head()
```

```
Out[80]:
```

	book_theater_id	show_date	audience_count	avg_audience	min_audience	m
0	book_00001	2023-01-13	50	47.06	2	
1	book_00001	2023-01-14	64	47.06	2	
2	book_00001	2023-01-15	58	47.06	2	
3	book_00001	2023-01-16	44	47.06	2	
4	book_00001	2023-01-18	12	47.06	2	

```
In [81]: bookknow_visits = bookknow_visits_enhanced.copy()
```

```
In [82]: bookknow_visits.head()
```

```
Out[82]:
```

	book_theater_id	show_date	audience_count	avg_audience	min_audience	m
0	book_00001	2023-01-13	50	47.06	2	
1	book_00001	2023-01-14	64	47.06	2	
2	book_00001	2023-01-15	58	47.06	2	
3	book_00001	2023-01-16	44	47.06	2	
4	book_00001	2023-01-18	12	47.06	2	

```
In [83]: bookknow_visits.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 214046 entries, 0 to 214045
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   book_theater_id        214046 non-null object  
1   show_date              214046 non-null object  
2   audience_count         214046 non-null int64   
3   avg_audience          214046 non-null float64  
4   min_audience          214046 non-null int64   
5   max_audience          214046 non-null int64   
6   std_audience          214046 non-null float64  
7   total_shows            214046 non-null int64   
dtypes: float64(2), int64(4), object(2)
memory usage: 13.1+ MB
```

3.2 Feature Engineeering of show_date

```
In [84]: bookknow_visits['show_date'] = pd.to_datetime(bookknow_visits['show_date'])
bookknow_visits['year'] = bookknow_visits['show_date'].dt.year
bookknow_visits['month'] = bookknow_visits['show_date'].dt.month
bookknow_visits['day'] = bookknow_visits['show_date'].dt.day
bookknow_visits['day_of_week'] = bookknow_visits['show_date'].dt.day_of_week
bookknow_visits['weekofyear'] = bookknow_visits['show_date'].dt.isocalendar().week
bookknow_visits['day_of_year'] = bookknow_visits['show_date'].dt.dayofyear
bookknow_visits['quarter'] = bookknow_visits['show_date'].dt.quarter
bookknow_visits['day_of_quarter'] = bookknow_visits['show_date'].apply(
    lambda x: (x - pd.Timestamp(x.year, (x.quarter-1)*3 + 1, 1)).days + 1
)
bookknow_visits['is_weekend'] = bookknow_visits['day_of_week'].isin([5, 6]).astype(int)
bookknow_visits['is_month_start'] = bookknow_visits['show_date'].dt.is_month_start
bookknow_visits['is_month_end'] = bookknow_visits['show_date'].dt.is_month_end
bookknow_visits['is_quarter_start'] = bookknow_visits['show_date'].dt.is_quarter_start
bookknow_visits['is_quarter_end'] = bookknow_visits['show_date'].dt.is_quarter_end

# Interaction features
bookknow_visits['theater_x_month'] = bookknow_visits['book_theater_id'].astype(int)
```

```
bookknow_visits['theater_x_dow'] = bookknow_visits['book_theater_id'].astype(str)
bookknow_visits['month_x_dow'] = bookknow_visits['month'].astype(str) + '_' + bc
```

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

```
Out[85]:
```

	book_theater_id	show_date	audience_count	avg_audience	min_audience	m
0	book_00001	2023-01-13	50	47.06	2	
1	book_00001	2023-01-14	64	47.06	2	
2	book_00001	2023-01-15	58	47.06	2	
3	book_00001	2023-01-16	44	47.06	2	
4	book_00001	2023-01-18	12	47.06	2	

5 rows × 24 columns

```
In [86]: bookknow_visits.isna().sum()
```

```
Out[86]: book_theater_id    0
show_date                  0
audience_count            0
avg_audience              0
min_audience              0
max_audience              0
std_audience              0
total_shows                0
year                       0
month                      0
day                        0
day_of_week                0
weekofyear                 0
day_of_year                0
quarter                    0
day_of_quarter             0
is_weekend                 0
is_month_start             0
is_month_end               0
is_quarter_start           0
is_quarter_end             0
theater_x_month            0
theater_x_dow              0
month_x_dow                0
dtype: int64
```

3.3 Creating Rolling Features

```
In [87]: # Sort for proper time series features
bookknow_visits = bookknow_visits.sort_values(['book_theater_id', 'show_date']).
```

```

for lag in [1, 2, 3, 7, 14, 21, 28]:
    booknow_visits[f'lag_{lag}'] = booknow_visits.groupby('book_theater_id')['audience_count'].shift(lag)

for window in [3, 7, 14, 21, 28]:
    booknow_visits[f'rolling_mean_{window}'] = (
        booknow_visits.groupby('book_theater_id')['audience_count']
        .transform(lambda x: x.rolling(window=window, min_periods=1).mean())
    )
    booknow_visits[f'rolling_std_{window}'] = (
        booknow_visits.groupby('book_theater_id')['audience_count']
        .transform(lambda x: x.rolling(window=window, min_periods=1).std())
    )
    booknow_visits[f'rolling_min_{window}'] = (
        booknow_visits.groupby('book_theater_id')['audience_count']
        .transform(lambda x: x.rolling(window=window, min_periods=1).min())
    )
    booknow_visits[f'rolling_max_{window}'] = (
        booknow_visits.groupby('book_theater_id')['audience_count']
        .transform(lambda x: x.rolling(window=window, min_periods=1).max())
    )

for span in [3, 7, 14]:
    booknow_visits[f'ewm_{span}'] = (
        booknow_visits.groupby('book_theater_id')['audience_count']
        .transform(lambda x: x.ewm(span=span, min_periods=1).mean())
    )

# Theater-specific features
booknow_visits['days_from_avg'] = booknow_visits['audience_count'] - booknow_visits['avg_audience']
booknow_visits['pct_of_avg'] = booknow_visits['audience_count'] / (booknow_visits['avg_audience'] + 1)

```

In [88]: `booknow_visits.columns`

Out[88]: Index(['book_theater_id', 'show_date', 'audience_count', 'avg_audience', 'min_audience', 'max_audience', 'std_audience', 'total_shows', 'year', 'month', 'day', 'day_of_week', 'weekofyear', 'day_of_year', 'quarter', 'day_of_quarter', 'is_weekend', 'is_month_start', 'is_month_end', 'is_quarter_start', 'is_quarter_end', 'theater_x_month', 'theater_x_dow', 'month_x_dow', 'lag_1', 'lag_2', 'lag_3', 'lag_7', 'lag_14', 'lag_21', 'lag_28', 'rolling_mean_3', 'rolling_std_3', 'rolling_min_3', 'rolling_max_3', 'rolling_mean_7', 'rolling_std_7', 'rolling_min_7', 'rolling_max_7', 'rolling_mean_14', 'rolling_std_14', 'rolling_min_14', 'rolling_max_14', 'rolling_mean_21', 'rolling_std_21', 'rolling_min_21', 'rolling_max_21', 'rolling_mean_28', 'rolling_std_28', 'rolling_min_28', 'rolling_max_28', 'ewm_3', 'ewm_7', 'ewm_14', 'days_from_avg', 'pct_of_avg'], dtype='object')

In [89]: `booknow_visits.head()`

```
Out[89]:
```

	book_theater_id	show_date	audience_count	avg_audience	min_audience	m
0	book_00001	2023-01-13	50	47.06	2	
1	book_00001	2023-01-14	64	47.06	2	
2	book_00001	2023-01-15	58	47.06	2	
3	book_00001	2023-01-16	44	47.06	2	
4	book_00001	2023-01-18	12	47.06	2	

5 rows × 56 columns

```
In [90]: bookknow_visits.isna().sum()
```

```
Out[90]:
```

book_theater_id	0
show_date	0
audience_count	0
avg_audience	0
min_audience	0
max_audience	0
std_audience	0
total_shows	0
year	0
month	0
day	0
day_of_week	0
weekofyear	0
day_of_year	0
quarter	0
day_of_quarter	0
is_weekend	0
is_month_start	0
is_month_end	0
is_quarter_start	0
is_quarter_end	0
theater_x_month	0
theater_x_dow	0
month_x_dow	0
lag_1	826
lag_2	1651
lag_3	2475
lag_7	5771
lag_14	11539
lag_21	17299
lag_28	23046
rolling_mean_3	0
rolling_std_3	826
rolling_min_3	0
rolling_max_3	0
rolling_mean_7	0
rolling_std_7	826
rolling_min_7	0
rolling_max_7	0
rolling_mean_14	0
rolling_std_14	826
rolling_min_14	0
rolling_max_14	0
rolling_mean_21	0
rolling_std_21	826
rolling_min_21	0
rolling_max_21	0
rolling_mean_28	0
rolling_std_28	826
rolling_min_28	0
rolling_max_28	0
ewm_3	0
ewm_7	0
ewm_14	0

```
days_from_avg      0
pct_of_avg          0
dtype: int64
```

```
In [91]: # 1. Fill lag features with 0 (indicates no historical data)
lag_cols = [f'lag_{i}' for i in [1, 2, 3, 7, 14, 21, 28]]
for col in lag_cols:
    booknow_visits[col].fillna(0, inplace=True)

# 2. Fill rolling std with 0 (no variation at start)
rolling_std_cols = [f'rolling_std_{i}' for i in [3, 7, 14, 21, 28]]
for col in rolling_std_cols:
    booknow_visits[col].fillna(0, inplace=True)
```

```
In [92]: booknow_visits.isna().sum()
```



```
days_from_avg      0
pct_of_avg          0
dtype: int64
```

```
In [93]: sample_submission.head()
```

```
Out[93]:
```

	ID	audience_count
0	book_00001_2024-03-01	0
1	book_00001_2024-03-02	0
2	book_00001_2024-03-03	0
3	book_00001_2024-03-04	0
4	book_00001_2024-03-06	0

3.4 Splitting date and id in sample_submission

```
In [94]: test = sample_submission.copy()

test['book_theater_id'] = test['ID'].apply(lambda x: "_".join(x.split('_')[:2])
test['show_date'] = test['ID'].apply(lambda x: x.split('_')[-1])

test['show_date'] = pd.to_datetime(test['show_date'])
test['year'] = test['show_date'].dt.year
test['month'] = test['show_date'].dt.month
test['day'] = test['show_date'].dt.day
test['day_of_week'] = test['show_date'].dt.day_of_week
test['weekofyear'] = test['show_date'].dt.isocalendar().week
test['is_weekend'] = test['day_of_week'].isin([5, 6]).astype(int)
test['is_month_start'] = test['show_date'].dt.is_month_start.astype(int)
test['is_month_end'] = test['show_date'].dt.is_month_end.astype(int)
```

```
In [95]: test.head()
```

```
Out[95]:
```

	ID	audience_count	book_theater_id	show_date	year	mo
0	book_00001_2024-03-01	0	book_00001	2024-03-01	2024	
1	book_00001_2024-03-02	0	book_00001	2024-03-02	2024	
2	book_00001_2024-03-03	0	book_00001	2024-03-03	2024	
3	book_00001_2024-03-04	0	book_00001	2024-03-04	2024	
4	book_00001_2024-03-06	0	book_00001	2024-03-06	2024	

```
In [96]: test_enhanced = test.merge(
    theater_performance,
    left_on='book_theater_id',
    right_index=True,
```

```
how='left'  
)
```

```
In [97]: test.shape
```

```
Out[97]: (38062, 12)
```

```
In [98]: test_enhanced.shape
```

```
Out[98]: (38062, 17)
```

```
In [99]: # Check for any missing theater statistics in test data  
missing_stats = test_enhanced[['avg_audience', 'min_audience', 'max_audience',  
print(missing_stats)
```

```
avg_audience    118  
min_audience    118  
max_audience    118  
std_audience    118  
dtype: int64
```

```
In [100... test_enhanced['avg_audience'].fillna(theater_performance['avg_audience'].media  
test_enhanced['min_audience'].fillna(theater_performance['min_audience'].media  
test_enhanced['max_audience'].fillna(theater_performance['max_audience'].media  
test_enhanced['std_audience'].fillna(theater_performance['std_audience'].media
```

```
In [101... test = test_enhanced.copy()
```

```
In [102... test.head()
```

```
Out[102...  
      ID  audience_count  book_theater_id  show_date  year  m  
0  book_00001_2024-03-01          0      book_00001  2024-03-01  2024  
1  book_00001_2024-03-02          0      book_00001  2024-03-02  2024  
2  book_00001_2024-03-03          0      book_00001  2024-03-03  2024  
3  book_00001_2024-03-04          0      book_00001  2024-03-04  2024  
4  book_00001_2024-03-06          0      book_00001  2024-03-06  2024
```

```
In [103... test.isna().sum()
```

```
Out[103... ID          0
audience_count  0
book_theater_id  0
show_date       0
year           0
month          0
day            0
day_of_week     0
weekofyear      0
is_weekend      0
is_month_start  0
is_month_end    0
avg_audience   0
min_audience   0
max_audience   0
std_audience   0
total_shows     118
dtype: int64
```

```
In [104... test.columns
```

```
Out[104... Index(['ID', 'audience_count', 'book_theater_id', 'show_date', 'year', 'month',
        'day', 'day_of_week', 'weekofyear', 'is_weekend', 'is_month_start',
        'is_month_end', 'avg_audience', 'min_audience', 'max_audience',
        'std_audience', 'total_shows'],
        dtype='object')
```

```
In [105... test = test[['book_theater_id', 'show_date', 'year', 'month',
        'day', 'day_of_week', 'weekofyear', 'is_weekend', 'is_month_start',
        'is_month_end', 'avg_audience', 'min_audience', 'max_audience',
        'std_audience']]
```

```
In [106... test.head()
```

```
Out[106...    book_theater_id  show_date  year  month  day  day_of_week  weekofyear  is_
0      book_00001  2024-03-01  2024     3     1           4           9
1      book_00001  2024-03-02  2024     3     2           5           9
2      book_00001  2024-03-03  2024     3     3           6           9
3      book_00001  2024-03-04  2024     3     4           0          10
4      book_00001  2024-03-06  2024     3     6           2          10
```

```
In [107... bookknow_visits.head()
```

```
Out[107...
```

	book_theater_id	show_date	audience_count	avg_audience	min_audience	m
0	book_00001	2023-01-13	50	47.06	2	
1	book_00001	2023-01-14	64	47.06	2	
2	book_00001	2023-01-15	58	47.06	2	
3	book_00001	2023-01-16	44	47.06	2	
4	book_00001	2023-01-18	12	47.06	2	

5 rows × 56 columns

3.5 Lebel Encoding

```
In [108... # Encode categorical features
le_theater = LabelEncoder()
bookknow_visits['book_theater_id_encoded'] = le_theater.fit_transform(bookknow_v

le_theater_month = LabelEncoder()
bookknow_visits['theater_x_month_encoded'] = le_theater_month.fit_transform(boc

le_theater_dow = LabelEncoder()
bookknow_visits['theater_x_dow_encoded'] = le_theater_dow.fit_transform(bookknow

le_month_dow = LabelEncoder()
bookknow_visits['month_x_dow_encoded'] = le_month_dow.fit_transform(bookknow_vis
```

```
In [109... bookknow_visits.columns
```

```
Out[109... Index(['book_theater_id', 'show_date', 'audience_count', 'avg_audience',
      'min_audience', 'max_audience', 'std_audience', 'total_shows', 'year',
      'month', 'day', 'day_of_week', 'weekofyear', 'day_of_year', 'quarter',
      'day_of_quarter', 'is_weekend', 'is_month_start', 'is_month_end',
      'is_quarter_start', 'is_quarter_end', 'theater_x_month',
      'theater_x_dow', 'month_x_dow', 'lag_1', 'lag_2', 'lag_3', 'lag_7',
      'lag_14', 'lag_21', 'lag_28', 'rolling_mean_3', 'rolling_std_3',
      'rolling_min_3', 'rolling_max_3', 'rolling_mean_7', 'rolling_std_7',
      'rolling_min_7', 'rolling_max_7', 'rolling_mean_14', 'rolling_std_14',
      'rolling_min_14', 'rolling_max_14', 'rolling_mean_21', 'rolling_std_2
1',
      'rolling_min_21', 'rolling_max_21', 'rolling_mean_28', 'rolling_std_2
8',
      'rolling_min_28', 'rolling_max_28', 'ewm_3', 'ewm_7', 'ewm_14',
      'days_from_avg', 'pct_of_avg', 'book_theater_id_encoded',
      'theater_x_month_encoded', 'theater_x_dow_encoded',
      'month_x_dow_encoded'],
      dtype='object')
```

```
In [110... bookknow_visits.head()
```

Out[110...

	book_theater_id	show_date	audience_count	avg_audience	min_audience	m
0	book_00001	2023-01-13	50	47.06	2	
1	book_00001	2023-01-14	64	47.06	2	
2	book_00001	2023-01-15	58	47.06	2	
3	book_00001	2023-01-16	44	47.06	2	
4	book_00001	2023-01-18	12	47.06	2	

5 rows × 60 columns

In [111...

```
bookknow_visits.isna().sum()
```

```
Out[111]: book_theater_id      0
          show_date            0
          audience_count       0
          avg_audience         0
          min_audience         0
          max_audience         0
          std_audience         0
          total_shows           0
          year                  0
          month                 0
          day                   0
          day_of_week           0
          weekofyear            0
          day_of_year           0
          quarter               0
          day_of_quarter        0
          is_weekend            0
          is_month_start        0
          is_month_end          0
          is_quarter_start      0
          is_quarter_end        0
          theater_x_month       0
          theater_x_dow         0
          month_x_dow           0
          lag_1                  0
          lag_2                  0
          lag_3                  0
          lag_7                  0
          lag_14                 0
          lag_21                 0
          lag_28                 0
          rolling_mean_3         0
          rolling_std_3          0
          rolling_min_3          0
          rolling_max_3          0
          rolling_mean_7         0
          rolling_std_7          0
          rolling_min_7          0
          rolling_max_7          0
          rolling_mean_14        0
          rolling_std_14         0
          rolling_min_14         0
          rolling_max_14         0
          rolling_mean_21        0
          rolling_std_21         0
          rolling_min_21         0
          rolling_max_21         0
          rolling_mean_28        0
          rolling_std_28         0
          rolling_min_28         0
          rolling_max_28         0
          ewm_3                  0
          ewm_7                  0
          ewm_14                 0
```

```
days_from_avg      0
pct_of_avg          0
book_theater_id_encoded  0
theater_x_month_encoded  0
theater_x_dow_encoded  0
month_x_dow_encoded  0
dtype: int64
```

4. Model Training and Evaluation

4.1 Train-test split

```
In [112... # Prepare train-test split (time-based with proper handling)
dates = booknow_visits['show_date'].copy()

# Exclude non-feature columns
feature_cols = [col for col in booknow_visits.columns if col not in
                 ['audience_count', 'show_date', 'book_theater_id',
                  'theater_x_month', 'theater_x_dow', 'month_x_dow', 'total_sho

X = booknow_visits[feature_cols].copy()
y = booknow_visits['audience_count'].copy()

# 80-20 time-based split
split_date = dates.quantile(0.8)
train_mask = dates < split_date
val_mask = dates >= split_date

X_train = X[train_mask].copy()
X_val = X[val_mask].copy()
y_train = y[train_mask].copy()
y_val = y[val_mask].copy()

# IMPROVEMENT 1: Fill missing values AFTER split to prevent leakage
# Fill lag features with 0 (no historical data)
lag_cols = [col for col in X_train.columns if col.startswith('lag_')]
for col in lag_cols:
    train_median = X_train[col].median()
    X_train[col].fillna(0, inplace=True)
    X_val[col].fillna(0, inplace=True) # Same strategy for validation

# Fill rolling std with 0 (no variation)
rolling_std_cols = [col for col in X_train.columns if 'rolling_std' in col]
for col in rolling_std_cols:
    X_train[col].fillna(0, inplace=True)
    X_val[col].fillna(0, inplace=True)

# IMPROVEMENT 2: Verify no data leakage
print(f"Split Date: {split_date}")
print(f"Training set: {X_train.shape} | Date range: {dates[train_mask].min()}")
```

```
print(f"Validation set: {X_val.shape} | Date range: {dates[val_mask].min()} to {dates[val_mask].max()}")
print(f"Number of features: {len(feature_cols)}")
print(f"\nMissing values after split:")
print(f"X_train: {X_train.isna().sum().sum()}")
print(f"X_val: {X_val.isna().sum().sum()}")
```

Split Date: 2023-12-26 00:00:00

Training set: (170848, 53) | Date range: 2023-01-01 00:00:00 to 2023-12-25 00:00:00

Validation set: (43198, 53) | Date range: 2023-12-26 00:00:00 to 2024-02-28 00:00:00

Number of features: 53

Missing values after split:

X_train: 0

X_val: 0

In [113... X.shape

Out[113... (214046, 53)

In [114... y.shape

Out[114... (214046,)

In [115... X.isna().sum()


```
Out[115... avg_audience      0
min_audience      0
max_audience      0
std_audience      0
year              0
month             0
day              0
day_of_week       0
weekofyear        0
day_of_year       0
quarter           0
day_of_quarter    0
is_weekend        0
is_month_start    0
is_month_end      0
is_quarter_start  0
is_quarter_end    0
lag_1             0
lag_2             0
lag_3             0
lag_7             0
lag_14            0
lag_21            0
lag_28            0
rolling_mean_3    0
rolling_std_3     0
rolling_min_3     0
rolling_max_3     0
rolling_mean_7    0
rolling_std_7     0
rolling_min_7     0
rolling_max_7     0
rolling_mean_14   0
rolling_std_14    0
rolling_min_14    0
rolling_max_14    0
rolling_mean_21   0
rolling_std_21    0
rolling_min_21    0
rolling_max_21    0
rolling_mean_28   0
rolling_std_28    0
rolling_min_28    0
rolling_max_28    0
ewm_3             0
ewm_7             0
ewm_14            0
days_from_avg    0
pct_of_avg        0
book_theater_id_encoded 0
theater_x_month_encoded 0
theater_x_dow_encoded 0
month_x_dow_encoded 0
dtype: int64
```

4.2 Training LightGBM

```
In [116... # Train LightGBM
lgb_model = lgb.LGBMRegressor(
    n_estimators= 2697,
    learning_rate= 0.008551710560716899,
    max_depth= 15,
    num_leaves= 128,
    min_child_samples= 5,
    subsample= 0.8477666519446061,
    colsample_bytree= 0.7683035452440005,
    reg_alpha= 4.92758821776188e-07,
    reg_lambda= 1.2372718914798502e-05,
    min_split_gain= 0.6160322533877769,
    min_child_weight= 0.00030183953515291486
)

lgb_model.fit(X_train, y_train)
y_pred_lgb = lgb_model.predict(X_val)
r2_lgb = r2_score(y_val, y_pred_lgb)
print(f"LightGBM R2 = {r2_lgb:.4f}")
```


[illegible]

4.4 Training CatBoost

```
In [118... # Train CatBoost
cat_model = CatBoostRegressor(
    iterations= 2982,
    learning_rate= 0.04004161059195288,
    depth= 6,
    l2_leaf_reg= 3.0660978237034504,
    border_count= 199,
    random_strength= 3.8370951834387395,
    bagging_temperature= 0.3226078854044478
)
cat_model.fit(X_train, y_train)
y_pred_cat = cat_model.predict(X_val)
r2_cat = r2_score(y_val, y_pred_cat)
print(f"CatBoost R2 = {r2_cat:.4f}")
has_catboost = True
```

0:	learn: 31.9871760	total: 88.5ms	remaining: 4m 23s
1:	learn: 30.9624309	total: 101ms	remaining: 2m 30s
2:	learn: 29.9555059	total: 115ms	remaining: 1m 54s
3:	learn: 28.9761824	total: 130ms	remaining: 1m 36s
4:	learn: 28.0745762	total: 142ms	remaining: 1m 24s
5:	learn: 27.1643316	total: 155ms	remaining: 1m 16s
6:	learn: 26.3214320	total: 168ms	remaining: 1m 11s
7:	learn: 25.5211396	total: 180ms	remaining: 1m 6s
8:	learn: 24.7060833	total: 193ms	remaining: 1m 3s
9:	learn: 23.9607191	total: 206ms	remaining: 1m 1s
10:	learn: 23.2374576	total: 220ms	remaining: 59.4s
11:	learn: 22.5304854	total: 235ms	remaining: 58.1s
12:	learn: 21.8489880	total: 249ms	remaining: 56.9s
13:	learn: 21.1964641	total: 263ms	remaining: 55.7s
14:	learn: 20.5772930	total: 276ms	remaining: 54.6s
15:	learn: 20.0029277	total: 290ms	remaining: 53.8s
16:	learn: 19.4159990	total: 304ms	remaining: 53s
17:	learn: 18.8671786	total: 317ms	remaining: 52.2s
18:	learn: 18.3125835	total: 331ms	remaining: 51.5s
19:	learn: 17.7751761	total: 344ms	remaining: 50.9s
20:	learn: 17.3196928	total: 358ms	remaining: 50.5s
21:	learn: 16.8266088	total: 372ms	remaining: 50.1s
22:	learn: 16.3520148	total: 385ms	remaining: 49.5s
23:	learn: 15.9154571	total: 398ms	remaining: 49.1s
24:	learn: 15.4720895	total: 412ms	remaining: 48.7s
25:	learn: 15.0556654	total: 424ms	remaining: 48.2s
26:	learn: 14.6529130	total: 437ms	remaining: 47.8s
27:	learn: 14.2613083	total: 450ms	remaining: 47.5s
28:	learn: 13.9075842	total: 463ms	remaining: 47.2s
29:	learn: 13.5880790	total: 476ms	remaining: 46.9s
30:	learn: 13.2703960	total: 490ms	remaining: 46.6s
31:	learn: 12.9452523	total: 503ms	remaining: 46.4s
32:	learn: 12.6057344	total: 515ms	remaining: 46s
33:	learn: 12.3018425	total: 528ms	remaining: 45.7s
34:	learn: 12.0196861	total: 542ms	remaining: 45.6s
35:	learn: 11.7227865	total: 555ms	remaining: 45.4s
36:	learn: 11.4209130	total: 568ms	remaining: 45.2s
37:	learn: 11.1416655	total: 580ms	remaining: 44.9s
38:	learn: 10.8864338	total: 592ms	remaining: 44.7s
39:	learn: 10.6405460	total: 604ms	remaining: 44.5s
40:	learn: 10.4084069	total: 617ms	remaining: 44.3s
41:	learn: 10.1590418	total: 630ms	remaining: 44.1s
42:	learn: 9.9465663	total: 642ms	remaining: 43.9s
43:	learn: 9.7262064	total: 654ms	remaining: 43.7s
44:	learn: 9.5198744	total: 665ms	remaining: 43.4s
45:	learn: 9.3202830	total: 677ms	remaining: 43.2s
46:	learn: 9.1139412	total: 689ms	remaining: 43s
47:	learn: 8.9241261	total: 702ms	remaining: 42.9s
48:	learn: 8.7435649	total: 714ms	remaining: 42.7s
49:	learn: 8.5807186	total: 727ms	remaining: 42.6s
50:	learn: 8.4273158	total: 739ms	remaining: 42.5s
51:	learn: 8.2648521	total: 752ms	remaining: 42.4s
52:	learn: 8.1130743	total: 764ms	remaining: 42.2s
53:	learn: 7.9640312	total: 776ms	remaining: 42.1s

54:	learn: 7.8230698	total: 788ms	remaining: 41.9s
55:	learn: 7.6863657	total: 800ms	remaining: 41.8s
56:	learn: 7.5512694	total: 812ms	remaining: 41.6s
57:	learn: 7.4279393	total: 824ms	remaining: 41.5s
58:	learn: 7.3009523	total: 836ms	remaining: 41.4s
59:	learn: 7.1859611	total: 849ms	remaining: 41.3s
60:	learn: 7.0737349	total: 861ms	remaining: 41.2s
61:	learn: 6.9616631	total: 872ms	remaining: 41.1s
62:	learn: 6.8586283	total: 885ms	remaining: 41s
63:	learn: 6.7437490	total: 898ms	remaining: 41s
64:	learn: 6.6443642	total: 910ms	remaining: 40.8s
65:	learn: 6.5443318	total: 922ms	remaining: 40.7s
66:	learn: 6.4555563	total: 934ms	remaining: 40.6s
67:	learn: 6.3552685	total: 947ms	remaining: 40.6s
68:	learn: 6.2595774	total: 959ms	remaining: 40.5s
69:	learn: 6.1752255	total: 971ms	remaining: 40.4s
70:	learn: 6.0970118	total: 983ms	remaining: 40.3s
71:	learn: 6.0166664	total: 995ms	remaining: 40.2s
72:	learn: 5.9429480	total: 1.01s	remaining: 40.1s
73:	learn: 5.8578874	total: 1.02s	remaining: 40s
74:	learn: 5.7881689	total: 1.03s	remaining: 39.9s
75:	learn: 5.7090659	total: 1.04s	remaining: 39.9s
76:	learn: 5.6280787	total: 1.06s	remaining: 39.9s
77:	learn: 5.5626462	total: 1.07s	remaining: 39.8s
78:	learn: 5.4963026	total: 1.08s	remaining: 39.7s
79:	learn: 5.4261131	total: 1.09s	remaining: 39.7s
80:	learn: 5.3585130	total: 1.1s	remaining: 39.6s
81:	learn: 5.2958725	total: 1.12s	remaining: 39.5s
82:	learn: 5.2415889	total: 1.13s	remaining: 39.5s
83:	learn: 5.1764909	total: 1.14s	remaining: 39.4s
84:	learn: 5.1248000	total: 1.15s	remaining: 39.4s
85:	learn: 5.0665709	total: 1.17s	remaining: 39.3s
86:	learn: 5.0120270	total: 1.18s	remaining: 39.2s
87:	learn: 4.9663698	total: 1.19s	remaining: 39.1s
88:	learn: 4.9152048	total: 1.2s	remaining: 39.1s
89:	learn: 4.8662886	total: 1.22s	remaining: 39.1s
90:	learn: 4.8188708	total: 1.23s	remaining: 39s
91:	learn: 4.7726668	total: 1.24s	remaining: 38.9s
92:	learn: 4.7297119	total: 1.25s	remaining: 38.9s
93:	learn: 4.6915539	total: 1.26s	remaining: 38.8s
94:	learn: 4.6511481	total: 1.28s	remaining: 38.8s
95:	learn: 4.6082138	total: 1.29s	remaining: 38.7s
96:	learn: 4.5624106	total: 1.3s	remaining: 38.7s
97:	learn: 4.5275075	total: 1.31s	remaining: 38.6s
98:	learn: 4.4946705	total: 1.32s	remaining: 38.6s
99:	learn: 4.4533849	total: 1.34s	remaining: 38.5s
100:	learn: 4.4144767	total: 1.35s	remaining: 38.5s
101:	learn: 4.3746356	total: 1.36s	remaining: 38.4s
102:	learn: 4.3308640	total: 1.37s	remaining: 38.3s
103:	learn: 4.2966685	total: 1.38s	remaining: 38.3s
104:	learn: 4.2584407	total: 1.4s	remaining: 38.3s
105:	learn: 4.2324481	total: 1.41s	remaining: 38.3s
106:	learn: 4.1910574	total: 1.42s	remaining: 38.2s
107:	learn: 4.1653046	total: 1.43s	remaining: 38.2s

108:	learn: 4.1386407	total: 1.45s	remaining: 38.1s
109:	learn: 4.1093160	total: 1.46s	remaining: 38.1s
110:	learn: 4.0736919	total: 1.47s	remaining: 38.1s
111:	learn: 4.0510926	total: 1.48s	remaining: 38s
112:	learn: 4.0213056	total: 1.5s	remaining: 38s
113:	learn: 3.9980608	total: 1.51s	remaining: 38s
114:	learn: 3.9650067	total: 1.52s	remaining: 38s
115:	learn: 3.9448670	total: 1.53s	remaining: 37.9s
116:	learn: 3.9129278	total: 1.55s	remaining: 37.9s
117:	learn: 3.8888651	total: 1.56s	remaining: 37.9s
118:	learn: 3.8590797	total: 1.58s	remaining: 37.9s
119:	learn: 3.8307622	total: 1.59s	remaining: 37.9s
120:	learn: 3.8033710	total: 1.6s	remaining: 37.9s
121:	learn: 3.7834831	total: 1.61s	remaining: 37.8s
122:	learn: 3.7614903	total: 1.63s	remaining: 37.8s
123:	learn: 3.7357955	total: 1.64s	remaining: 37.8s
124:	learn: 3.7134524	total: 1.65s	remaining: 37.8s
125:	learn: 3.6945700	total: 1.67s	remaining: 37.8s
126:	learn: 3.6694586	total: 1.68s	remaining: 37.7s
127:	learn: 3.6502600	total: 1.69s	remaining: 37.7s
128:	learn: 3.6268319	total: 1.71s	remaining: 37.7s
129:	learn: 3.6044650	total: 1.72s	remaining: 37.7s
130:	learn: 3.5845508	total: 1.73s	remaining: 37.7s
131:	learn: 3.5658111	total: 1.74s	remaining: 37.6s
132:	learn: 3.5476645	total: 1.75s	remaining: 37.6s
133:	learn: 3.5273192	total: 1.77s	remaining: 37.6s
134:	learn: 3.5102801	total: 1.78s	remaining: 37.5s
135:	learn: 3.4933311	total: 1.79s	remaining: 37.5s
136:	learn: 3.4774966	total: 1.81s	remaining: 37.6s
137:	learn: 3.4592164	total: 1.83s	remaining: 37.6s
138:	learn: 3.4338873	total: 1.84s	remaining: 37.7s
139:	learn: 3.4131543	total: 1.86s	remaining: 37.7s
140:	learn: 3.3896604	total: 1.88s	remaining: 37.8s
141:	learn: 3.3761394	total: 1.89s	remaining: 37.8s
142:	learn: 3.3585088	total: 1.91s	remaining: 37.8s
143:	learn: 3.3389538	total: 1.92s	remaining: 37.8s
144:	learn: 3.3266321	total: 1.94s	remaining: 37.9s
145:	learn: 3.3068939	total: 1.95s	remaining: 37.9s
146:	learn: 3.2879785	total: 1.96s	remaining: 37.9s
147:	learn: 3.2673210	total: 1.98s	remaining: 37.9s
148:	learn: 3.2551128	total: 1.99s	remaining: 37.9s
149:	learn: 3.2374236	total: 2s	remaining: 37.8s
150:	learn: 3.2237154	total: 2.02s	remaining: 37.8s
151:	learn: 3.2135360	total: 2.03s	remaining: 37.8s
152:	learn: 3.1972990	total: 2.04s	remaining: 37.8s
153:	learn: 3.1866549	total: 2.05s	remaining: 37.7s
154:	learn: 3.1728177	total: 2.07s	remaining: 37.7s
155:	learn: 3.1627634	total: 2.08s	remaining: 37.7s
156:	learn: 3.1441608	total: 2.1s	remaining: 37.7s
157:	learn: 3.1251105	total: 2.11s	remaining: 37.7s
158:	learn: 3.1074647	total: 2.12s	remaining: 37.6s
159:	learn: 3.0969961	total: 2.13s	remaining: 37.6s
160:	learn: 3.0859053	total: 2.15s	remaining: 37.6s
161:	learn: 3.0676297	total: 2.16s	remaining: 37.6s

162:	learn: 3.0503269	total: 2.17s	remaining: 37.6s
163:	learn: 3.0362073	total: 2.18s	remaining: 37.5s
164:	learn: 3.0192158	total: 2.2s	remaining: 37.5s
165:	learn: 3.0040634	total: 2.21s	remaining: 37.5s
166:	learn: 2.9907791	total: 2.22s	remaining: 37.5s
167:	learn: 2.9820337	total: 2.23s	remaining: 37.4s
168:	learn: 2.9672847	total: 2.25s	remaining: 37.4s
169:	learn: 2.9534541	total: 2.26s	remaining: 37.4s
170:	learn: 2.9398718	total: 2.27s	remaining: 37.3s
171:	learn: 2.9279143	total: 2.28s	remaining: 37.3s
172:	learn: 2.9169566	total: 2.29s	remaining: 37.3s
173:	learn: 2.9024085	total: 2.31s	remaining: 37.3s
174:	learn: 2.8896413	total: 2.32s	remaining: 37.2s
175:	learn: 2.8748948	total: 2.33s	remaining: 37.2s
176:	learn: 2.8599696	total: 2.35s	remaining: 37.2s
177:	learn: 2.8538556	total: 2.36s	remaining: 37.2s
178:	learn: 2.8441750	total: 2.37s	remaining: 37.2s
179:	learn: 2.8343029	total: 2.39s	remaining: 37.2s
180:	learn: 2.8250213	total: 2.4s	remaining: 37.2s
181:	learn: 2.8106468	total: 2.41s	remaining: 37.1s
182:	learn: 2.8058026	total: 2.43s	remaining: 37.1s
183:	learn: 2.7932726	total: 2.44s	remaining: 37.1s
184:	learn: 2.7811233	total: 2.45s	remaining: 37.1s
185:	learn: 2.7742124	total: 2.46s	remaining: 37.1s
186:	learn: 2.7625719	total: 2.48s	remaining: 37s
187:	learn: 2.7484501	total: 2.49s	remaining: 37.1s
188:	learn: 2.7356433	total: 2.5s	remaining: 37s
189:	learn: 2.7276488	total: 2.52s	remaining: 37s
190:	learn: 2.7175711	total: 2.53s	remaining: 37s
191:	learn: 2.7078044	total: 2.54s	remaining: 36.9s
192:	learn: 2.6978215	total: 2.55s	remaining: 36.9s
193:	learn: 2.6844563	total: 2.56s	remaining: 36.9s
194:	learn: 2.6716417	total: 2.58s	remaining: 36.8s
195:	learn: 2.6600081	total: 2.59s	remaining: 36.8s
196:	learn: 2.6469752	total: 2.6s	remaining: 36.8s
197:	learn: 2.6401615	total: 2.62s	remaining: 36.8s
198:	learn: 2.6283011	total: 2.63s	remaining: 36.8s
199:	learn: 2.6169590	total: 2.64s	remaining: 36.7s
200:	learn: 2.6090974	total: 2.65s	remaining: 36.7s
201:	learn: 2.6019816	total: 2.67s	remaining: 36.7s
202:	learn: 2.5932615	total: 2.68s	remaining: 36.7s
203:	learn: 2.5879223	total: 2.69s	remaining: 36.7s
204:	learn: 2.5780738	total: 2.71s	remaining: 36.6s
205:	learn: 2.5715629	total: 2.72s	remaining: 36.6s
206:	learn: 2.5614605	total: 2.73s	remaining: 36.6s
207:	learn: 2.5523169	total: 2.74s	remaining: 36.5s
208:	learn: 2.5415916	total: 2.75s	remaining: 36.5s
209:	learn: 2.5306997	total: 2.77s	remaining: 36.5s
210:	learn: 2.5300719	total: 2.78s	remaining: 36.5s
211:	learn: 2.5232040	total: 2.79s	remaining: 36.5s
212:	learn: 2.5153981	total: 2.81s	remaining: 36.5s
213:	learn: 2.5099560	total: 2.82s	remaining: 36.5s
214:	learn: 2.4990229	total: 2.83s	remaining: 36.4s
215:	learn: 2.4887280	total: 2.84s	remaining: 36.4s

216:	learn: 2.4794715	total: 2.85s	remaining: 36.4s
217:	learn: 2.4740241	total: 2.87s	remaining: 36.4s
218:	learn: 2.4646184	total: 2.88s	remaining: 36.3s
219:	learn: 2.4542786	total: 2.89s	remaining: 36.3s
220:	learn: 2.4491202	total: 2.9s	remaining: 36.3s
221:	learn: 2.4433655	total: 2.92s	remaining: 36.3s
222:	learn: 2.4362435	total: 2.93s	remaining: 36.3s
223:	learn: 2.4271774	total: 2.94s	remaining: 36.2s
224:	learn: 2.4194973	total: 2.96s	remaining: 36.2s
225:	learn: 2.4068776	total: 2.97s	remaining: 36.2s
226:	learn: 2.4025651	total: 2.98s	remaining: 36.2s
227:	learn: 2.3931122	total: 2.99s	remaining: 36.1s
228:	learn: 2.3827807	total: 3s	remaining: 36.1s
229:	learn: 2.3762856	total: 3.02s	remaining: 36.1s
230:	learn: 2.3668279	total: 3.03s	remaining: 36.1s
231:	learn: 2.3593125	total: 3.04s	remaining: 36.1s
232:	learn: 2.3531423	total: 3.06s	remaining: 36.1s
233:	learn: 2.3445978	total: 3.07s	remaining: 36s
234:	learn: 2.3359708	total: 3.08s	remaining: 36s
235:	learn: 2.3266461	total: 3.09s	remaining: 36s
236:	learn: 2.3182191	total: 3.11s	remaining: 36s
237:	learn: 2.3081928	total: 3.12s	remaining: 36s
238:	learn: 2.3010421	total: 3.13s	remaining: 36s
239:	learn: 2.2940403	total: 3.15s	remaining: 35.9s
240:	learn: 2.2849404	total: 3.16s	remaining: 35.9s
241:	learn: 2.2763101	total: 3.17s	remaining: 35.9s
242:	learn: 2.2687514	total: 3.19s	remaining: 35.9s
243:	learn: 2.2601148	total: 3.2s	remaining: 35.9s
244:	learn: 2.2519418	total: 3.21s	remaining: 35.9s
245:	learn: 2.2441795	total: 3.23s	remaining: 35.9s
246:	learn: 2.2372913	total: 3.24s	remaining: 35.9s
247:	learn: 2.2314665	total: 3.25s	remaining: 35.9s
248:	learn: 2.2241120	total: 3.27s	remaining: 35.8s
249:	learn: 2.2206559	total: 3.28s	remaining: 35.8s
250:	learn: 2.2149952	total: 3.29s	remaining: 35.8s
251:	learn: 2.2074949	total: 3.3s	remaining: 35.8s
252:	learn: 2.1967342	total: 3.31s	remaining: 35.8s
253:	learn: 2.1893158	total: 3.33s	remaining: 35.8s
254:	learn: 2.1842870	total: 3.35s	remaining: 35.8s
255:	learn: 2.1784009	total: 3.36s	remaining: 35.8s
256:	learn: 2.1717777	total: 3.37s	remaining: 35.8s
257:	learn: 2.1651056	total: 3.38s	remaining: 35.7s
258:	learn: 2.1600354	total: 3.4s	remaining: 35.7s
259:	learn: 2.1528683	total: 3.41s	remaining: 35.7s
260:	learn: 2.1459809	total: 3.42s	remaining: 35.7s
261:	learn: 2.1392615	total: 3.43s	remaining: 35.6s
262:	learn: 2.1340463	total: 3.44s	remaining: 35.6s
263:	learn: 2.1272287	total: 3.46s	remaining: 35.6s
264:	learn: 2.1234332	total: 3.47s	remaining: 35.6s
265:	learn: 2.1226171	total: 3.48s	remaining: 35.5s
266:	learn: 2.1157976	total: 3.49s	remaining: 35.5s
267:	learn: 2.1071794	total: 3.51s	remaining: 35.5s
268:	learn: 2.0990452	total: 3.52s	remaining: 35.5s
269:	learn: 2.0920430	total: 3.53s	remaining: 35.5s

270:	learn: 2.0912161	total: 3.54s	remaining: 35.5s
271:	learn: 2.0852629	total: 3.56s	remaining: 35.4s
272:	learn: 2.0785115	total: 3.57s	remaining: 35.4s
273:	learn: 2.0726606	total: 3.58s	remaining: 35.4s
274:	learn: 2.0690790	total: 3.59s	remaining: 35.3s
275:	learn: 2.0631893	total: 3.6s	remaining: 35.4s
276:	learn: 2.0574385	total: 3.62s	remaining: 35.3s
277:	learn: 2.0520017	total: 3.63s	remaining: 35.3s
278:	learn: 2.0468122	total: 3.64s	remaining: 35.3s
279:	learn: 2.0414599	total: 3.65s	remaining: 35.3s
280:	learn: 2.0358675	total: 3.67s	remaining: 35.3s
281:	learn: 2.0303032	total: 3.68s	remaining: 35.3s
282:	learn: 2.0257171	total: 3.69s	remaining: 35.2s
283:	learn: 2.0239737	total: 3.71s	remaining: 35.2s
284:	learn: 2.0178580	total: 3.72s	remaining: 35.2s
285:	learn: 2.0137824	total: 3.73s	remaining: 35.2s
286:	learn: 2.0071444	total: 3.74s	remaining: 35.1s
287:	learn: 2.0007516	total: 3.75s	remaining: 35.1s
288:	learn: 1.9948865	total: 3.77s	remaining: 35.1s
289:	learn: 1.9891076	total: 3.78s	remaining: 35.1s
290:	learn: 1.9826557	total: 3.79s	remaining: 35.1s
291:	learn: 1.9773150	total: 3.81s	remaining: 35.1s
292:	learn: 1.9727207	total: 3.82s	remaining: 35s
293:	learn: 1.9672679	total: 3.83s	remaining: 35s
294:	learn: 1.9623000	total: 3.85s	remaining: 35s
295:	learn: 1.9572775	total: 3.86s	remaining: 35s
296:	learn: 1.9522177	total: 3.87s	remaining: 35s
297:	learn: 1.9465892	total: 3.88s	remaining: 35s
298:	learn: 1.9428735	total: 3.89s	remaining: 34.9s
299:	learn: 1.9420645	total: 3.9s	remaining: 34.9s
300:	learn: 1.9376538	total: 3.92s	remaining: 34.9s
301:	learn: 1.9309702	total: 3.93s	remaining: 34.9s
302:	learn: 1.9268028	total: 3.94s	remaining: 34.9s
303:	learn: 1.9213061	total: 3.95s	remaining: 34.8s
304:	learn: 1.9173187	total: 3.96s	remaining: 34.8s
305:	learn: 1.9117648	total: 3.98s	remaining: 34.8s
306:	learn: 1.9065515	total: 3.99s	remaining: 34.8s
307:	learn: 1.9014247	total: 4s	remaining: 34.8s
308:	learn: 1.8960904	total: 4.02s	remaining: 34.8s
309:	learn: 1.8901311	total: 4.03s	remaining: 34.7s
310:	learn: 1.8855318	total: 4.04s	remaining: 34.7s
311:	learn: 1.8847037	total: 4.06s	remaining: 34.7s
312:	learn: 1.8789730	total: 4.07s	remaining: 34.7s
313:	learn: 1.8743833	total: 4.08s	remaining: 34.7s
314:	learn: 1.8685843	total: 4.1s	remaining: 34.7s
315:	learn: 1.8644154	total: 4.11s	remaining: 34.7s
316:	learn: 1.8602164	total: 4.12s	remaining: 34.6s
317:	learn: 1.8568927	total: 4.13s	remaining: 34.6s
318:	learn: 1.8525612	total: 4.14s	remaining: 34.6s
319:	learn: 1.8477709	total: 4.16s	remaining: 34.6s
320:	learn: 1.8428290	total: 4.17s	remaining: 34.6s
321:	learn: 1.8419799	total: 4.18s	remaining: 34.6s
322:	learn: 1.8374885	total: 4.2s	remaining: 34.5s
323:	learn: 1.8338568	total: 4.21s	remaining: 34.5s

324:	learn: 1.8302765	total: 4.22s	remaining: 34.5s
325:	learn: 1.8245925	total: 4.23s	remaining: 34.5s
326:	learn: 1.8203522	total: 4.25s	remaining: 34.5s
327:	learn: 1.8154819	total: 4.26s	remaining: 34.5s
328:	learn: 1.8103165	total: 4.28s	remaining: 34.5s
329:	learn: 1.8054175	total: 4.29s	remaining: 34.5s
330:	learn: 1.8011505	total: 4.3s	remaining: 34.5s
331:	learn: 1.7974478	total: 4.32s	remaining: 34.4s
332:	learn: 1.7924465	total: 4.33s	remaining: 34.4s
333:	learn: 1.7881538	total: 4.34s	remaining: 34.4s
334:	learn: 1.7845783	total: 4.35s	remaining: 34.4s
335:	learn: 1.7836532	total: 4.37s	remaining: 34.4s
336:	learn: 1.7793005	total: 4.38s	remaining: 34.4s
337:	learn: 1.7753555	total: 4.39s	remaining: 34.4s
338:	learn: 1.7710184	total: 4.4s	remaining: 34.3s
339:	learn: 1.7658340	total: 4.42s	remaining: 34.3s
340:	learn: 1.7608651	total: 4.43s	remaining: 34.3s
341:	learn: 1.7567374	total: 4.44s	remaining: 34.3s
342:	learn: 1.7523582	total: 4.45s	remaining: 34.3s
343:	learn: 1.7484035	total: 4.46s	remaining: 34.2s
344:	learn: 1.7454041	total: 4.48s	remaining: 34.2s
345:	learn: 1.7407208	total: 4.49s	remaining: 34.2s
346:	learn: 1.7375169	total: 4.5s	remaining: 34.2s
347:	learn: 1.7335554	total: 4.52s	remaining: 34.2s
348:	learn: 1.7288593	total: 4.53s	remaining: 34.2s
349:	learn: 1.7281975	total: 4.54s	remaining: 34.1s
350:	learn: 1.7241301	total: 4.55s	remaining: 34.1s
351:	learn: 1.7203907	total: 4.57s	remaining: 34.1s
352:	learn: 1.7169301	total: 4.58s	remaining: 34.1s
353:	learn: 1.7133907	total: 4.59s	remaining: 34.1s
354:	learn: 1.7092990	total: 4.61s	remaining: 34.1s
355:	learn: 1.7050138	total: 4.62s	remaining: 34.1s
356:	learn: 1.7024798	total: 4.63s	remaining: 34s
357:	learn: 1.6986764	total: 4.64s	remaining: 34s
358:	learn: 1.6948657	total: 4.65s	remaining: 34s
359:	learn: 1.6903224	total: 4.67s	remaining: 34s
360:	learn: 1.6867733	total: 4.68s	remaining: 34s
361:	learn: 1.6841709	total: 4.69s	remaining: 34s
362:	learn: 1.6800494	total: 4.71s	remaining: 34s
363:	learn: 1.6770896	total: 4.72s	remaining: 33.9s
364:	learn: 1.6734739	total: 4.73s	remaining: 33.9s
365:	learn: 1.6707397	total: 4.74s	remaining: 33.9s
366:	learn: 1.6670927	total: 4.75s	remaining: 33.9s
367:	learn: 1.6630099	total: 4.77s	remaining: 33.9s
368:	learn: 1.6621687	total: 4.78s	remaining: 33.8s
369:	learn: 1.6613448	total: 4.79s	remaining: 33.8s
370:	learn: 1.6574984	total: 4.81s	remaining: 33.8s
371:	learn: 1.6566134	total: 4.82s	remaining: 33.8s
372:	learn: 1.6559768	total: 4.83s	remaining: 33.8s
373:	learn: 1.6524916	total: 4.84s	remaining: 33.8s
374:	learn: 1.6494499	total: 4.86s	remaining: 33.8s
375:	learn: 1.6457361	total: 4.87s	remaining: 33.7s
376:	learn: 1.6421002	total: 4.88s	remaining: 33.7s
377:	learn: 1.6379646	total: 4.89s	remaining: 33.7s

378:	learn: 1.6342653	total: 4.91s	remaining: 33.7s
379:	learn: 1.6312548	total: 4.92s	remaining: 33.7s
380:	learn: 1.6284375	total: 4.93s	remaining: 33.7s
381:	learn: 1.6248597	total: 4.95s	remaining: 33.7s
382:	learn: 1.6211255	total: 4.96s	remaining: 33.6s
383:	learn: 1.6181962	total: 4.97s	remaining: 33.6s
384:	learn: 1.6147078	total: 4.98s	remaining: 33.6s
385:	learn: 1.6121451	total: 5s	remaining: 33.6s
386:	learn: 1.6090415	total: 5.01s	remaining: 33.6s
387:	learn: 1.6049800	total: 5.02s	remaining: 33.6s
388:	learn: 1.6021499	total: 5.03s	remaining: 33.5s
389:	learn: 1.6014075	total: 5.04s	remaining: 33.5s
390:	learn: 1.5984731	total: 5.06s	remaining: 33.5s
391:	learn: 1.5951072	total: 5.07s	remaining: 33.5s
392:	learn: 1.5920448	total: 5.09s	remaining: 33.5s
393:	learn: 1.5885408	total: 5.1s	remaining: 33.5s
394:	learn: 1.5878220	total: 5.11s	remaining: 33.5s
395:	learn: 1.5851589	total: 5.13s	remaining: 33.5s
396:	learn: 1.5823930	total: 5.14s	remaining: 33.4s
397:	learn: 1.5791667	total: 5.15s	remaining: 33.4s
398:	learn: 1.5758627	total: 5.16s	remaining: 33.4s
399:	learn: 1.5725280	total: 5.18s	remaining: 33.4s
400:	learn: 1.5691815	total: 5.19s	remaining: 33.4s
401:	learn: 1.5663964	total: 5.2s	remaining: 33.4s
402:	learn: 1.5626321	total: 5.21s	remaining: 33.4s
403:	learn: 1.5599348	total: 5.22s	remaining: 33.3s
404:	learn: 1.5595160	total: 5.24s	remaining: 33.3s
405:	learn: 1.5566273	total: 5.25s	remaining: 33.3s
406:	learn: 1.5538065	total: 5.26s	remaining: 33.3s
407:	learn: 1.5533658	total: 5.27s	remaining: 33.3s
408:	learn: 1.5499480	total: 5.29s	remaining: 33.3s
409:	learn: 1.5496181	total: 5.3s	remaining: 33.2s
410:	learn: 1.5470356	total: 5.31s	remaining: 33.2s
411:	learn: 1.5445835	total: 5.32s	remaining: 33.2s
412:	learn: 1.5420723	total: 5.33s	remaining: 33.2s
413:	learn: 1.5393540	total: 5.35s	remaining: 33.2s
414:	learn: 1.5366661	total: 5.36s	remaining: 33.2s
415:	learn: 1.5337125	total: 5.37s	remaining: 33.1s
416:	learn: 1.5305265	total: 5.39s	remaining: 33.1s
417:	learn: 1.5284613	total: 5.4s	remaining: 33.1s
418:	learn: 1.5256943	total: 5.41s	remaining: 33.1s
419:	learn: 1.5228916	total: 5.42s	remaining: 33.1s
420:	learn: 1.5201021	total: 5.44s	remaining: 33.1s
421:	learn: 1.5166840	total: 5.45s	remaining: 33.1s
422:	learn: 1.5146283	total: 5.46s	remaining: 33s
423:	learn: 1.5130860	total: 5.47s	remaining: 33s
424:	learn: 1.5097914	total: 5.49s	remaining: 33s
425:	learn: 1.5068021	total: 5.5s	remaining: 33s
426:	learn: 1.5040622	total: 5.51s	remaining: 33s
427:	learn: 1.5018827	total: 5.52s	remaining: 33s
428:	learn: 1.4989234	total: 5.54s	remaining: 32.9s
429:	learn: 1.4983140	total: 5.55s	remaining: 32.9s
430:	learn: 1.4961544	total: 5.56s	remaining: 32.9s
431:	learn: 1.4940940	total: 5.57s	remaining: 32.9s

432:	learn: 1.4916541	total: 5.58s	remaining: 32.9s
433:	learn: 1.4887100	total: 5.59s	remaining: 32.8s
434:	learn: 1.4881323	total: 5.61s	remaining: 32.8s
435:	learn: 1.4857746	total: 5.62s	remaining: 32.8s
436:	learn: 1.4852094	total: 5.63s	remaining: 32.8s
437:	learn: 1.4818893	total: 5.65s	remaining: 32.8s
438:	learn: 1.4813402	total: 5.66s	remaining: 32.8s
439:	learn: 1.4808038	total: 5.67s	remaining: 32.8s
440:	learn: 1.4802796	total: 5.69s	remaining: 32.8s
441:	learn: 1.4797673	total: 5.7s	remaining: 32.8s
442:	learn: 1.4792667	total: 5.71s	remaining: 32.7s
443:	learn: 1.4787775	total: 5.72s	remaining: 32.7s
444:	learn: 1.4782992	total: 5.74s	remaining: 32.7s
445:	learn: 1.4757748	total: 5.75s	remaining: 32.7s
446:	learn: 1.4753060	total: 5.76s	remaining: 32.7s
447:	learn: 1.4730362	total: 5.78s	remaining: 32.7s
448:	learn: 1.4702310	total: 5.79s	remaining: 32.7s
449:	learn: 1.4669768	total: 5.8s	remaining: 32.7s
450:	learn: 1.4665161	total: 5.82s	remaining: 32.6s
451:	learn: 1.4640908	total: 5.83s	remaining: 32.6s
452:	learn: 1.4610699	total: 5.84s	remaining: 32.6s
453:	learn: 1.4580241	total: 5.85s	remaining: 32.6s
454:	learn: 1.4558629	total: 5.87s	remaining: 32.6s
455:	learn: 1.4554723	total: 5.88s	remaining: 32.6s
456:	learn: 1.4531886	total: 5.89s	remaining: 32.6s
457:	learn: 1.4505343	total: 5.91s	remaining: 32.5s
458:	learn: 1.4480390	total: 5.92s	remaining: 32.5s
459:	learn: 1.4447582	total: 5.93s	remaining: 32.5s
460:	learn: 1.4427641	total: 5.94s	remaining: 32.5s
461:	learn: 1.4396677	total: 5.96s	remaining: 32.5s
462:	learn: 1.4375569	total: 5.97s	remaining: 32.5s
463:	learn: 1.4347481	total: 5.98s	remaining: 32.4s
464:	learn: 1.4314723	total: 5.99s	remaining: 32.4s
465:	learn: 1.4294882	total: 6s	remaining: 32.4s
466:	learn: 1.4275358	total: 6.01s	remaining: 32.4s
467:	learn: 1.4245140	total: 6.03s	remaining: 32.4s
468:	learn: 1.4241443	total: 6.04s	remaining: 32.4s
469:	learn: 1.4217620	total: 6.05s	remaining: 32.3s
470:	learn: 1.4189790	total: 6.06s	remaining: 32.3s
471:	learn: 1.4186414	total: 6.08s	remaining: 32.3s
472:	learn: 1.4183905	total: 6.09s	remaining: 32.3s
473:	learn: 1.4160018	total: 6.1s	remaining: 32.3s
474:	learn: 1.4157556	total: 6.11s	remaining: 32.3s
475:	learn: 1.4155146	total: 6.13s	remaining: 32.3s
476:	learn: 1.4128738	total: 6.14s	remaining: 32.2s
477:	learn: 1.4110033	total: 6.15s	remaining: 32.2s
478:	learn: 1.4087107	total: 6.17s	remaining: 32.2s
479:	learn: 1.4060072	total: 6.18s	remaining: 32.2s
480:	learn: 1.4028686	total: 6.19s	remaining: 32.2s
481:	learn: 1.4010160	total: 6.2s	remaining: 32.2s
482:	learn: 1.3986442	total: 6.21s	remaining: 32.1s
483:	learn: 1.3983283	total: 6.22s	remaining: 32.1s
484:	learn: 1.3965525	total: 6.24s	remaining: 32.1s
485:	learn: 1.3940849	total: 6.25s	remaining: 32.1s

486:	learn: 1.3915308	total: 6.26s	remaining: 32.1s
487:	learn: 1.3887771	total: 6.27s	remaining: 32s
488:	learn: 1.3865141	total: 6.28s	remaining: 32s
489:	learn: 1.3846793	total: 6.29s	remaining: 32s
490:	learn: 1.3822982	total: 6.31s	remaining: 32s
491:	learn: 1.3792294	total: 6.32s	remaining: 32s
492:	learn: 1.3771469	total: 6.33s	remaining: 31.9s
493:	learn: 1.3751726	total: 6.34s	remaining: 31.9s
494:	learn: 1.3729576	total: 6.36s	remaining: 31.9s
495:	learn: 1.3705144	total: 6.37s	remaining: 31.9s
496:	learn: 1.3680228	total: 6.38s	remaining: 31.9s
497:	learn: 1.3656276	total: 6.39s	remaining: 31.9s
498:	learn: 1.3635829	total: 6.4s	remaining: 31.9s
499:	learn: 1.3613657	total: 6.41s	remaining: 31.8s
500:	learn: 1.3592056	total: 6.43s	remaining: 31.8s
501:	learn: 1.3569301	total: 6.44s	remaining: 31.8s
502:	learn: 1.3542543	total: 6.45s	remaining: 31.8s
503:	learn: 1.3518021	total: 6.46s	remaining: 31.8s
504:	learn: 1.3492183	total: 6.47s	remaining: 31.7s
505:	learn: 1.3468050	total: 6.48s	remaining: 31.7s
506:	learn: 1.3448654	total: 6.5s	remaining: 31.7s
507:	learn: 1.3431122	total: 6.51s	remaining: 31.7s
508:	learn: 1.3408169	total: 6.52s	remaining: 31.7s
509:	learn: 1.3378452	total: 6.53s	remaining: 31.7s
510:	learn: 1.3357491	total: 6.54s	remaining: 31.6s
511:	learn: 1.3332898	total: 6.55s	remaining: 31.6s
512:	learn: 1.3314543	total: 6.57s	remaining: 31.6s
513:	learn: 1.3311680	total: 6.58s	remaining: 31.6s
514:	learn: 1.3308889	total: 6.59s	remaining: 31.6s
515:	learn: 1.3289561	total: 6.61s	remaining: 31.6s
516:	learn: 1.3255795	total: 6.62s	remaining: 31.6s
517:	learn: 1.3253077	total: 6.63s	remaining: 31.5s
518:	learn: 1.3234933	total: 6.64s	remaining: 31.5s
519:	learn: 1.3212632	total: 6.65s	remaining: 31.5s
520:	learn: 1.3184228	total: 6.67s	remaining: 31.5s
521:	learn: 1.3162067	total: 6.68s	remaining: 31.5s
522:	learn: 1.3145612	total: 6.69s	remaining: 31.5s
523:	learn: 1.3127564	total: 6.71s	remaining: 31.5s
524:	learn: 1.3113952	total: 6.72s	remaining: 31.4s
525:	learn: 1.3097734	total: 6.73s	remaining: 31.4s
526:	learn: 1.3074403	total: 6.74s	remaining: 31.4s
527:	learn: 1.3058524	total: 6.75s	remaining: 31.4s
528:	learn: 1.3044847	total: 6.77s	remaining: 31.4s
529:	learn: 1.3026654	total: 6.78s	remaining: 31.4s
530:	learn: 1.3009214	total: 6.79s	remaining: 31.4s
531:	learn: 1.2988859	total: 6.8s	remaining: 31.3s
532:	learn: 1.2970012	total: 6.82s	remaining: 31.3s
533:	learn: 1.2946053	total: 6.83s	remaining: 31.3s
534:	learn: 1.2928000	total: 6.84s	remaining: 31.3s
535:	learn: 1.2910183	total: 6.85s	remaining: 31.3s
536:	learn: 1.2889972	total: 6.87s	remaining: 31.3s
537:	learn: 1.2868353	total: 6.88s	remaining: 31.2s
538:	learn: 1.2853043	total: 6.89s	remaining: 31.2s
539:	learn: 1.2836737	total: 6.9s	remaining: 31.2s

540:	learn: 1.2822093	total: 6.91s	remaining: 31.2s
541:	learn: 1.2805699	total: 6.92s	remaining: 31.2s
542:	learn: 1.2790287	total: 6.94s	remaining: 31.2s
543:	learn: 1.2772451	total: 6.95s	remaining: 31.2s
544:	learn: 1.2755734	total: 6.96s	remaining: 31.1s
545:	learn: 1.2725457	total: 6.98s	remaining: 31.1s
546:	learn: 1.2705400	total: 6.99s	remaining: 31.1s
547:	learn: 1.2690024	total: 7s	remaining: 31.1s
548:	learn: 1.2673915	total: 7.01s	remaining: 31.1s
549:	learn: 1.2671332	total: 7.03s	remaining: 31.1s
550:	learn: 1.2650473	total: 7.04s	remaining: 31.1s
551:	learn: 1.2647957	total: 7.05s	remaining: 31.1s
552:	learn: 1.2632351	total: 7.07s	remaining: 31s
553:	learn: 1.2620128	total: 7.08s	remaining: 31s
554:	learn: 1.2605228	total: 7.09s	remaining: 31s
555:	learn: 1.2585258	total: 7.1s	remaining: 31s
556:	learn: 1.2582824	total: 7.12s	remaining: 31s
557:	learn: 1.2580448	total: 7.13s	remaining: 31s
558:	learn: 1.2560464	total: 7.14s	remaining: 30.9s
559:	learn: 1.2558145	total: 7.15s	remaining: 30.9s
560:	learn: 1.2537732	total: 7.17s	remaining: 30.9s
561:	learn: 1.2525627	total: 7.18s	remaining: 30.9s
562:	learn: 1.2511385	total: 7.19s	remaining: 30.9s
563:	learn: 1.2509132	total: 7.2s	remaining: 30.9s
564:	learn: 1.2495516	total: 7.21s	remaining: 30.9s
565:	learn: 1.2480588	total: 7.23s	remaining: 30.9s
566:	learn: 1.2460726	total: 7.24s	remaining: 30.8s
567:	learn: 1.2445876	total: 7.25s	remaining: 30.8s
568:	learn: 1.2443708	total: 7.26s	remaining: 30.8s
569:	learn: 1.2426989	total: 7.28s	remaining: 30.8s
570:	learn: 1.2409001	total: 7.29s	remaining: 30.8s
571:	learn: 1.2394872	total: 7.3s	remaining: 30.8s
572:	learn: 1.2393050	total: 7.31s	remaining: 30.7s
573:	learn: 1.2390961	total: 7.33s	remaining: 30.7s
574:	learn: 1.2379381	total: 7.34s	remaining: 30.7s
575:	learn: 1.2362808	total: 7.35s	remaining: 30.7s
576:	learn: 1.2347451	total: 7.37s	remaining: 30.7s
577:	learn: 1.2334892	total: 7.38s	remaining: 30.7s
578:	learn: 1.2318153	total: 7.39s	remaining: 30.7s
579:	learn: 1.2298631	total: 7.4s	remaining: 30.6s
580:	learn: 1.2278221	total: 7.41s	remaining: 30.6s
581:	learn: 1.2260050	total: 7.42s	remaining: 30.6s
582:	learn: 1.2246610	total: 7.43s	remaining: 30.6s
583:	learn: 1.2233269	total: 7.45s	remaining: 30.6s
584:	learn: 1.2231538	total: 7.46s	remaining: 30.6s
585:	learn: 1.2218484	total: 7.47s	remaining: 30.6s
586:	learn: 1.2206342	total: 7.49s	remaining: 30.5s
587:	learn: 1.2190861	total: 7.5s	remaining: 30.5s
588:	learn: 1.2175965	total: 7.51s	remaining: 30.5s
589:	learn: 1.2158493	total: 7.52s	remaining: 30.5s
590:	learn: 1.2140640	total: 7.53s	remaining: 30.5s
591:	learn: 1.2123436	total: 7.54s	remaining: 30.5s
592:	learn: 1.2107304	total: 7.56s	remaining: 30.4s
593:	learn: 1.2091592	total: 7.57s	remaining: 30.4s

594:	learn: 1.2076897	total: 7.58s	remaining: 30.4s
595:	learn: 1.2062821	total: 7.59s	remaining: 30.4s
596:	learn: 1.2052787	total: 7.61s	remaining: 30.4s
597:	learn: 1.2040037	total: 7.62s	remaining: 30.4s
598:	learn: 1.2025022	total: 7.63s	remaining: 30.4s
599:	learn: 1.2012086	total: 7.64s	remaining: 30.4s
600:	learn: 1.1986845	total: 7.66s	remaining: 30.3s
601:	learn: 1.1971350	total: 7.67s	remaining: 30.3s
602:	learn: 1.1969691	total: 7.68s	remaining: 30.3s
603:	learn: 1.1958703	total: 7.7s	remaining: 30.3s
604:	learn: 1.1957090	total: 7.71s	remaining: 30.3s
605:	learn: 1.1942220	total: 7.72s	remaining: 30.3s
606:	learn: 1.1928109	total: 7.74s	remaining: 30.3s
607:	learn: 1.1907875	total: 7.75s	remaining: 30.3s
608:	learn: 1.1890893	total: 7.76s	remaining: 30.2s
609:	learn: 1.1879005	total: 7.78s	remaining: 30.2s
610:	learn: 1.1864724	total: 7.79s	remaining: 30.2s
611:	learn: 1.1847445	total: 7.8s	remaining: 30.2s
612:	learn: 1.1836176	total: 7.81s	remaining: 30.2s
613:	learn: 1.1823635	total: 7.82s	remaining: 30.2s
614:	learn: 1.1809422	total: 7.84s	remaining: 30.2s
615:	learn: 1.1807880	total: 7.85s	remaining: 30.2s
616:	learn: 1.1795464	total: 7.86s	remaining: 30.1s
617:	learn: 1.1780715	total: 7.88s	remaining: 30.1s
618:	learn: 1.1766841	total: 7.89s	remaining: 30.1s
619:	learn: 1.1765332	total: 7.9s	remaining: 30.1s
620:	learn: 1.1750225	total: 7.91s	remaining: 30.1s
621:	learn: 1.1739965	total: 7.92s	remaining: 30.1s
622:	learn: 1.1725454	total: 7.94s	remaining: 30.1s
623:	learn: 1.1710998	total: 7.95s	remaining: 30s
624:	learn: 1.1698994	total: 7.96s	remaining: 30s
625:	learn: 1.1683130	total: 7.97s	remaining: 30s
626:	learn: 1.1674022	total: 7.98s	remaining: 30s
627:	learn: 1.1662764	total: 8s	remaining: 30s
628:	learn: 1.1640117	total: 8.01s	remaining: 29.9s
629:	learn: 1.1626937	total: 8.02s	remaining: 29.9s
630:	learn: 1.1613267	total: 8.03s	remaining: 29.9s
631:	learn: 1.1600575	total: 8.04s	remaining: 29.9s
632:	learn: 1.1585645	total: 8.05s	remaining: 29.9s
633:	learn: 1.1563257	total: 8.07s	remaining: 29.9s
634:	learn: 1.1550416	total: 8.08s	remaining: 29.9s
635:	learn: 1.1538966	total: 8.09s	remaining: 29.9s
636:	learn: 1.1527691	total: 8.11s	remaining: 29.8s
637:	learn: 1.1513197	total: 8.12s	remaining: 29.8s
638:	learn: 1.1498893	total: 8.13s	remaining: 29.8s
639:	learn: 1.1482690	total: 8.14s	remaining: 29.8s
640:	learn: 1.1467009	total: 8.15s	remaining: 29.8s
641:	learn: 1.1456159	total: 8.16s	remaining: 29.8s
642:	learn: 1.1440412	total: 8.18s	remaining: 29.8s
643:	learn: 1.1429010	total: 8.19s	remaining: 29.7s
644:	learn: 1.1415241	total: 8.21s	remaining: 29.7s
645:	learn: 1.1397939	total: 8.22s	remaining: 29.7s
646:	learn: 1.1384301	total: 8.23s	remaining: 29.7s
647:	learn: 1.1368173	total: 8.24s	remaining: 29.7s

648:	learn: 1.1355047	total: 8.26s	remaining: 29.7s
649:	learn: 1.1341972	total: 8.27s	remaining: 29.7s
650:	learn: 1.1332618	total: 8.28s	remaining: 29.6s
651:	learn: 1.1317067	total: 8.29s	remaining: 29.6s
652:	learn: 1.1303305	total: 8.3s	remaining: 29.6s
653:	learn: 1.1292489	total: 8.31s	remaining: 29.6s
654:	learn: 1.1284147	total: 8.33s	remaining: 29.6s
655:	learn: 1.1282580	total: 8.34s	remaining: 29.6s
656:	learn: 1.1273096	total: 8.35s	remaining: 29.6s
657:	learn: 1.1258637	total: 8.37s	remaining: 29.5s
658:	learn: 1.1248970	total: 8.38s	remaining: 29.5s
659:	learn: 1.1240296	total: 8.39s	remaining: 29.5s
660:	learn: 1.1224422	total: 8.4s	remaining: 29.5s
661:	learn: 1.1214332	total: 8.41s	remaining: 29.5s
662:	learn: 1.1199149	total: 8.42s	remaining: 29.5s
663:	learn: 1.1185386	total: 8.44s	remaining: 29.5s
664:	learn: 1.1175855	total: 8.45s	remaining: 29.4s
665:	learn: 1.1155002	total: 8.46s	remaining: 29.4s
666:	learn: 1.1142672	total: 8.47s	remaining: 29.4s
667:	learn: 1.1129847	total: 8.48s	remaining: 29.4s
668:	learn: 1.1118152	total: 8.5s	remaining: 29.4s
669:	learn: 1.1106430	total: 8.51s	remaining: 29.4s
670:	learn: 1.1096049	total: 8.52s	remaining: 29.4s
671:	learn: 1.1093970	total: 8.54s	remaining: 29.4s
672:	learn: 1.1083413	total: 8.56s	remaining: 29.4s
673:	learn: 1.1077494	total: 8.57s	remaining: 29.3s
674:	learn: 1.1075486	total: 8.58s	remaining: 29.3s
675:	learn: 1.1065250	total: 8.59s	remaining: 29.3s
676:	learn: 1.1055851	total: 8.61s	remaining: 29.3s
677:	learn: 1.1040332	total: 8.62s	remaining: 29.3s
678:	learn: 1.1027078	total: 8.63s	remaining: 29.3s
679:	learn: 1.1018248	total: 8.64s	remaining: 29.3s
680:	learn: 1.1004626	total: 8.65s	remaining: 29.2s
681:	learn: 1.0993644	total: 8.67s	remaining: 29.2s
682:	learn: 1.0983197	total: 8.68s	remaining: 29.2s
683:	learn: 1.0972944	total: 8.69s	remaining: 29.2s
684:	learn: 1.0959461	total: 8.7s	remaining: 29.2s
685:	learn: 1.0945353	total: 8.72s	remaining: 29.2s
686:	learn: 1.0938909	total: 8.73s	remaining: 29.2s
687:	learn: 1.0928324	total: 8.74s	remaining: 29.1s
688:	learn: 1.0920061	total: 8.75s	remaining: 29.1s
689:	learn: 1.0908826	total: 8.76s	remaining: 29.1s
690:	learn: 1.0898434	total: 8.78s	remaining: 29.1s
691:	learn: 1.0882964	total: 8.79s	remaining: 29.1s
692:	learn: 1.0873695	total: 8.8s	remaining: 29.1s
693:	learn: 1.0868360	total: 8.81s	remaining: 29.1s
694:	learn: 1.0860661	total: 8.82s	remaining: 29s
695:	learn: 1.0851887	total: 8.84s	remaining: 29s
696:	learn: 1.0841803	total: 8.85s	remaining: 29s
697:	learn: 1.0824802	total: 8.86s	remaining: 29s
698:	learn: 1.0812479	total: 8.88s	remaining: 29s
699:	learn: 1.0802401	total: 8.89s	remaining: 29s
700:	learn: 1.0787005	total: 8.9s	remaining: 29s
701:	learn: 1.0777473	total: 8.92s	remaining: 29s

702:	learn: 1.0767311	total: 8.93s	remaining: 28.9s
703:	learn: 1.0757339	total: 8.94s	remaining: 28.9s
704:	learn: 1.0747569	total: 8.96s	remaining: 28.9s
705:	learn: 1.0735551	total: 8.97s	remaining: 28.9s
706:	learn: 1.0728527	total: 8.98s	remaining: 28.9s
707:	learn: 1.0718186	total: 8.99s	remaining: 28.9s
708:	learn: 1.0703160	total: 9s	remaining: 28.9s
709:	learn: 1.0694680	total: 9.01s	remaining: 28.8s
710:	learn: 1.0689611	total: 9.03s	remaining: 28.8s
711:	learn: 1.0681198	total: 9.04s	remaining: 28.8s
712:	learn: 1.0673333	total: 9.05s	remaining: 28.8s
713:	learn: 1.0665601	total: 9.06s	remaining: 28.8s
714:	learn: 1.0654263	total: 9.07s	remaining: 28.8s
715:	learn: 1.0638547	total: 9.08s	remaining: 28.8s
716:	learn: 1.0627967	total: 9.1s	remaining: 28.7s
717:	learn: 1.0619034	total: 9.11s	remaining: 28.7s
718:	learn: 1.0612446	total: 9.12s	remaining: 28.7s
719:	learn: 1.0602852	total: 9.13s	remaining: 28.7s
720:	learn: 1.0593518	total: 9.15s	remaining: 28.7s
721:	learn: 1.0585015	total: 9.16s	remaining: 28.7s
722:	learn: 1.0575237	total: 9.17s	remaining: 28.7s
723:	learn: 1.0567665	total: 9.18s	remaining: 28.6s
724:	learn: 1.0558441	total: 9.2s	remaining: 28.6s
725:	learn: 1.0556724	total: 9.21s	remaining: 28.6s
726:	learn: 1.0549293	total: 9.22s	remaining: 28.6s
727:	learn: 1.0539223	total: 9.23s	remaining: 28.6s
728:	learn: 1.0534550	total: 9.24s	remaining: 28.6s
729:	learn: 1.0525939	total: 9.26s	remaining: 28.6s
730:	learn: 1.0518332	total: 9.27s	remaining: 28.6s
731:	learn: 1.0508378	total: 9.29s	remaining: 28.5s
732:	learn: 1.0498803	total: 9.3s	remaining: 28.5s
733:	learn: 1.0489287	total: 9.31s	remaining: 28.5s
734:	learn: 1.0485130	total: 9.32s	remaining: 28.5s
735:	learn: 1.0476349	total: 9.33s	remaining: 28.5s
736:	learn: 1.0468814	total: 9.35s	remaining: 28.5s
737:	learn: 1.0458303	total: 9.36s	remaining: 28.5s
738:	learn: 1.0451383	total: 9.37s	remaining: 28.5s
739:	learn: 1.0442608	total: 9.38s	remaining: 28.4s
740:	learn: 1.0430493	total: 9.4s	remaining: 28.4s
741:	learn: 1.0422092	total: 9.41s	remaining: 28.4s
742:	learn: 1.0416407	total: 9.42s	remaining: 28.4s
743:	learn: 1.0402666	total: 9.43s	remaining: 28.4s
744:	learn: 1.0394559	total: 9.44s	remaining: 28.4s
745:	learn: 1.0381106	total: 9.46s	remaining: 28.3s
746:	learn: 1.0373551	total: 9.47s	remaining: 28.3s
747:	learn: 1.0356811	total: 9.48s	remaining: 28.3s
748:	learn: 1.0343572	total: 9.49s	remaining: 28.3s
749:	learn: 1.0335207	total: 9.51s	remaining: 28.3s
750:	learn: 1.0324857	total: 9.52s	remaining: 28.3s
751:	learn: 1.0319556	total: 9.53s	remaining: 28.3s
752:	learn: 1.0309457	total: 9.54s	remaining: 28.3s
753:	learn: 1.0298833	total: 9.56s	remaining: 28.2s
754:	learn: 1.0289129	total: 9.57s	remaining: 28.2s
755:	learn: 1.0276861	total: 9.58s	remaining: 28.2s

756:	learn: 1.0265842	total: 9.59s	remaining: 28.2s
757:	learn: 1.0257741	total: 9.61s	remaining: 28.2s
758:	learn: 1.0250556	total: 9.62s	remaining: 28.2s
759:	learn: 1.0242446	total: 9.64s	remaining: 28.2s
760:	learn: 1.0236122	total: 9.65s	remaining: 28.2s
761:	learn: 1.0229822	total: 9.66s	remaining: 28.2s
762:	learn: 1.0223267	total: 9.67s	remaining: 28.1s
763:	learn: 1.0212275	total: 9.69s	remaining: 28.1s
764:	learn: 1.0203135	total: 9.7s	remaining: 28.1s
765:	learn: 1.0195815	total: 9.71s	remaining: 28.1s
766:	learn: 1.0184230	total: 9.72s	remaining: 28.1s
767:	learn: 1.0179627	total: 9.73s	remaining: 28.1s
768:	learn: 1.0172760	total: 9.75s	remaining: 28s
769:	learn: 1.0165304	total: 9.76s	remaining: 28s
770:	learn: 1.0154154	total: 9.77s	remaining: 28s
771:	learn: 1.0147634	total: 9.79s	remaining: 28s
772:	learn: 1.0139964	total: 9.8s	remaining: 28s
773:	learn: 1.0132812	total: 9.81s	remaining: 28s
774:	learn: 1.0126626	total: 9.82s	remaining: 28s
775:	learn: 1.0125200	total: 9.84s	remaining: 28s
776:	learn: 1.0122320	total: 9.85s	remaining: 27.9s
777:	learn: 1.0112465	total: 9.86s	remaining: 27.9s
778:	learn: 1.0104081	total: 9.87s	remaining: 27.9s
779:	learn: 1.0098075	total: 9.88s	remaining: 27.9s
780:	learn: 1.0087407	total: 9.89s	remaining: 27.9s
781:	learn: 1.0078745	total: 9.91s	remaining: 27.9s
782:	learn: 1.0068796	total: 9.92s	remaining: 27.9s
783:	learn: 1.0057352	total: 9.93s	remaining: 27.8s
784:	learn: 1.0047537	total: 9.94s	remaining: 27.8s
785:	learn: 1.0042246	total: 9.96s	remaining: 27.8s
786:	learn: 1.0030158	total: 9.96s	remaining: 27.8s
787:	learn: 1.0021747	total: 9.98s	remaining: 27.8s
788:	learn: 1.0013063	total: 9.99s	remaining: 27.8s
789:	learn: 1.0006333	total: 10s	remaining: 27.8s
790:	learn: 0.9999037	total: 10s	remaining: 27.7s
791:	learn: 0.9992825	total: 10s	remaining: 27.7s
792:	learn: 0.9983330	total: 10s	remaining: 27.7s
793:	learn: 0.9975893	total: 10s	remaining: 27.7s
794:	learn: 0.9969939	total: 10.1s	remaining: 27.7s
795:	learn: 0.9961034	total: 10.1s	remaining: 27.7s
796:	learn: 0.9953476	total: 10.1s	remaining: 27.7s
797:	learn: 0.9946945	total: 10.1s	remaining: 27.6s
798:	learn: 0.9938407	total: 10.1s	remaining: 27.6s
799:	learn: 0.9930836	total: 10.1s	remaining: 27.6s
800:	learn: 0.9923359	total: 10.1s	remaining: 27.6s
801:	learn: 0.9914957	total: 10.2s	remaining: 27.6s
802:	learn: 0.9907215	total: 10.2s	remaining: 27.6s
803:	learn: 0.9896521	total: 10.2s	remaining: 27.6s
804:	learn: 0.9890188	total: 10.2s	remaining: 27.6s
805:	learn: 0.9883515	total: 10.2s	remaining: 27.5s
806:	learn: 0.9877472	total: 10.2s	remaining: 27.5s
807:	learn: 0.9869743	total: 10.2s	remaining: 27.5s
808:	learn: 0.9858548	total: 10.2s	remaining: 27.5s
809:	learn: 0.9849632	total: 10.3s	remaining: 27.5s

810:	learn: 0.9843455	total: 10.3s	remaining: 27.5s
811:	learn: 0.9835339	total: 10.3s	remaining: 27.5s
812:	learn: 0.9830109	total: 10.3s	remaining: 27.5s
813:	learn: 0.9818279	total: 10.3s	remaining: 27.4s
814:	learn: 0.9816970	total: 10.3s	remaining: 27.4s
815:	learn: 0.9807753	total: 10.3s	remaining: 27.4s
816:	learn: 0.9803867	total: 10.3s	remaining: 27.4s
817:	learn: 0.9796517	total: 10.4s	remaining: 27.4s
818:	learn: 0.9786636	total: 10.4s	remaining: 27.4s
819:	learn: 0.9777436	total: 10.4s	remaining: 27.4s
820:	learn: 0.9767835	total: 10.4s	remaining: 27.3s
821:	learn: 0.9759027	total: 10.4s	remaining: 27.3s
822:	learn: 0.9751252	total: 10.4s	remaining: 27.3s
823:	learn: 0.9744601	total: 10.4s	remaining: 27.3s
824:	learn: 0.9743186	total: 10.4s	remaining: 27.3s
825:	learn: 0.9733797	total: 10.4s	remaining: 27.3s
826:	learn: 0.9730045	total: 10.5s	remaining: 27.3s
827:	learn: 0.9721894	total: 10.5s	remaining: 27.2s
828:	learn: 0.9715706	total: 10.5s	remaining: 27.2s
829:	learn: 0.9706361	total: 10.5s	remaining: 27.2s
830:	learn: 0.9700494	total: 10.5s	remaining: 27.2s
831:	learn: 0.9688633	total: 10.5s	remaining: 27.2s
832:	learn: 0.9682279	total: 10.5s	remaining: 27.2s
833:	learn: 0.9676078	total: 10.5s	remaining: 27.1s
834:	learn: 0.9669529	total: 10.6s	remaining: 27.1s
835:	learn: 0.9664450	total: 10.6s	remaining: 27.1s
836:	learn: 0.9659578	total: 10.6s	remaining: 27.1s
837:	learn: 0.9653349	total: 10.6s	remaining: 27.1s
838:	learn: 0.9647400	total: 10.6s	remaining: 27.1s
839:	learn: 0.9638911	total: 10.6s	remaining: 27.1s
840:	learn: 0.9631748	total: 10.6s	remaining: 27.1s
841:	learn: 0.9625471	total: 10.6s	remaining: 27s
842:	learn: 0.9619295	total: 10.7s	remaining: 27s
843:	learn: 0.9612118	total: 10.7s	remaining: 27s
844:	learn: 0.9604074	total: 10.7s	remaining: 27s
845:	learn: 0.9600492	total: 10.7s	remaining: 27s
846:	learn: 0.9592665	total: 10.7s	remaining: 27s
847:	learn: 0.9585521	total: 10.7s	remaining: 27s
848:	learn: 0.9581004	total: 10.7s	remaining: 26.9s
849:	learn: 0.9576086	total: 10.7s	remaining: 26.9s
850:	learn: 0.9568378	total: 10.7s	remaining: 26.9s
851:	learn: 0.9560558	total: 10.8s	remaining: 26.9s
852:	learn: 0.9549016	total: 10.8s	remaining: 26.9s
853:	learn: 0.9542839	total: 10.8s	remaining: 26.9s
854:	learn: 0.9536456	total: 10.8s	remaining: 26.9s
855:	learn: 0.9531876	total: 10.8s	remaining: 26.8s
856:	learn: 0.9522195	total: 10.8s	remaining: 26.8s
857:	learn: 0.9514745	total: 10.8s	remaining: 26.8s
858:	learn: 0.9509058	total: 10.8s	remaining: 26.8s
859:	learn: 0.9503445	total: 10.9s	remaining: 26.8s
860:	learn: 0.9500090	total: 10.9s	remaining: 26.8s
861:	learn: 0.9494146	total: 10.9s	remaining: 26.8s
862:	learn: 0.9483950	total: 10.9s	remaining: 26.8s
863:	learn: 0.9482724	total: 10.9s	remaining: 26.7s

864:	learn: 0.9474082	total: 10.9s	remaining: 26.7s
865:	learn: 0.9466797	total: 10.9s	remaining: 26.7s
866:	learn: 0.9461713	total: 10.9s	remaining: 26.7s
867:	learn: 0.9451213	total: 11s	remaining: 26.7s
868:	learn: 0.9442570	total: 11s	remaining: 26.7s
869:	learn: 0.9434169	total: 11s	remaining: 26.6s
870:	learn: 0.9425801	total: 11s	remaining: 26.6s
871:	learn: 0.9424597	total: 11s	remaining: 26.6s
872:	learn: 0.9416390	total: 11s	remaining: 26.6s
873:	learn: 0.9415226	total: 11s	remaining: 26.6s
874:	learn: 0.9408646	total: 11s	remaining: 26.6s
875:	learn: 0.9407504	total: 11.1s	remaining: 26.6s
876:	learn: 0.9406385	total: 11.1s	remaining: 26.6s
877:	learn: 0.9399896	total: 11.1s	remaining: 26.6s
878:	learn: 0.9393591	total: 11.1s	remaining: 26.5s
879:	learn: 0.9392497	total: 11.1s	remaining: 26.5s
880:	learn: 0.9385403	total: 11.1s	remaining: 26.5s
881:	learn: 0.9380830	total: 11.1s	remaining: 26.5s
882:	learn: 0.9375427	total: 11.1s	remaining: 26.5s
883:	learn: 0.9366606	total: 11.2s	remaining: 26.5s
884:	learn: 0.9360995	total: 11.2s	remaining: 26.5s
885:	learn: 0.9355897	total: 11.2s	remaining: 26.4s
886:	learn: 0.9354842	total: 11.2s	remaining: 26.4s
887:	learn: 0.9347668	total: 11.2s	remaining: 26.4s
888:	learn: 0.9342117	total: 11.2s	remaining: 26.4s
889:	learn: 0.9336591	total: 11.2s	remaining: 26.4s
890:	learn: 0.9329930	total: 11.2s	remaining: 26.4s
891:	learn: 0.9325011	total: 11.3s	remaining: 26.4s
892:	learn: 0.9319115	total: 11.3s	remaining: 26.4s
893:	learn: 0.9313822	total: 11.3s	remaining: 26.3s
894:	learn: 0.9305235	total: 11.3s	remaining: 26.3s
895:	learn: 0.9296311	total: 11.3s	remaining: 26.3s
896:	learn: 0.9289811	total: 11.3s	remaining: 26.3s
897:	learn: 0.9284060	total: 11.3s	remaining: 26.3s
898:	learn: 0.9278866	total: 11.3s	remaining: 26.3s
899:	learn: 0.9273250	total: 11.4s	remaining: 26.3s
900:	learn: 0.9264326	total: 11.4s	remaining: 26.2s
901:	learn: 0.9257934	total: 11.4s	remaining: 26.2s
902:	learn: 0.9252368	total: 11.4s	remaining: 26.2s
903:	learn: 0.9246607	total: 11.4s	remaining: 26.2s
904:	learn: 0.9245610	total: 11.4s	remaining: 26.2s
905:	learn: 0.9240733	total: 11.4s	remaining: 26.2s
906:	learn: 0.9233578	total: 11.4s	remaining: 26.2s
907:	learn: 0.9228426	total: 11.5s	remaining: 26.2s
908:	learn: 0.9220783	total: 11.5s	remaining: 26.1s
909:	learn: 0.9214433	total: 11.5s	remaining: 26.1s
910:	learn: 0.9213458	total: 11.5s	remaining: 26.1s
911:	learn: 0.9212503	total: 11.5s	remaining: 26.1s
912:	learn: 0.9205380	total: 11.5s	remaining: 26.1s
913:	learn: 0.9197120	total: 11.5s	remaining: 26.1s
914:	learn: 0.9189994	total: 11.5s	remaining: 26.1s
915:	learn: 0.9185824	total: 11.5s	remaining: 26s
916:	learn: 0.9179842	total: 11.6s	remaining: 26s
917:	learn: 0.9172776	total: 11.6s	remaining: 26s

918:	learn: 0.9166618	total: 11.6s	remaining: 26s
919:	learn: 0.9161798	total: 11.6s	remaining: 26s
920:	learn: 0.9154963	total: 11.6s	remaining: 26s
921:	learn: 0.9147098	total: 11.6s	remaining: 26s
922:	learn: 0.9141721	total: 11.6s	remaining: 25.9s
923:	learn: 0.9134078	total: 11.6s	remaining: 25.9s
924:	learn: 0.9128931	total: 11.7s	remaining: 25.9s
925:	learn: 0.9122971	total: 11.7s	remaining: 25.9s
926:	learn: 0.9114498	total: 11.7s	remaining: 25.9s
927:	learn: 0.9109904	total: 11.7s	remaining: 25.9s
928:	learn: 0.9106255	total: 11.7s	remaining: 25.9s
929:	learn: 0.9097130	total: 11.7s	remaining: 25.8s
930:	learn: 0.9090622	total: 11.7s	remaining: 25.8s
931:	learn: 0.9084373	total: 11.7s	remaining: 25.8s
932:	learn: 0.9076661	total: 11.7s	remaining: 25.8s
933:	learn: 0.9071514	total: 11.8s	remaining: 25.8s
934:	learn: 0.9066340	total: 11.8s	remaining: 25.8s
935:	learn: 0.9060275	total: 11.8s	remaining: 25.8s
936:	learn: 0.9054270	total: 11.8s	remaining: 25.8s
937:	learn: 0.9048644	total: 11.8s	remaining: 25.7s
938:	learn: 0.9039725	total: 11.8s	remaining: 25.7s
939:	learn: 0.9035077	total: 11.8s	remaining: 25.7s
940:	learn: 0.9030155	total: 11.8s	remaining: 25.7s
941:	learn: 0.9021306	total: 11.9s	remaining: 25.7s
942:	learn: 0.9017827	total: 11.9s	remaining: 25.7s
943:	learn: 0.9011012	total: 11.9s	remaining: 25.7s
944:	learn: 0.9005065	total: 11.9s	remaining: 25.6s
945:	learn: 0.8999255	total: 11.9s	remaining: 25.6s
946:	learn: 0.8995407	total: 11.9s	remaining: 25.6s
947:	learn: 0.8989626	total: 11.9s	remaining: 25.6s
948:	learn: 0.8983771	total: 11.9s	remaining: 25.6s
949:	learn: 0.8979158	total: 12s	remaining: 25.6s
950:	learn: 0.8972578	total: 12s	remaining: 25.6s
951:	learn: 0.8965696	total: 12s	remaining: 25.5s
952:	learn: 0.8960136	total: 12s	remaining: 25.5s
953:	learn: 0.8953748	total: 12s	remaining: 25.5s
954:	learn: 0.8949197	total: 12s	remaining: 25.5s
955:	learn: 0.8942914	total: 12s	remaining: 25.5s
956:	learn: 0.8934549	total: 12s	remaining: 25.5s
957:	learn: 0.8926439	total: 12s	remaining: 25.5s
958:	learn: 0.8920485	total: 12.1s	remaining: 25.4s
959:	learn: 0.8915770	total: 12.1s	remaining: 25.4s
960:	learn: 0.8910667	total: 12.1s	remaining: 25.4s
961:	learn: 0.8903141	total: 12.1s	remaining: 25.4s
962:	learn: 0.8897603	total: 12.1s	remaining: 25.4s
963:	learn: 0.8894558	total: 12.1s	remaining: 25.4s
964:	learn: 0.8888579	total: 12.1s	remaining: 25.4s
965:	learn: 0.8883869	total: 12.1s	remaining: 25.3s
966:	learn: 0.8878019	total: 12.2s	remaining: 25.3s
967:	learn: 0.8873199	total: 12.2s	remaining: 25.3s
968:	learn: 0.8866482	total: 12.2s	remaining: 25.3s
969:	learn: 0.8859952	total: 12.2s	remaining: 25.3s
970:	learn: 0.8857216	total: 12.2s	remaining: 25.3s
971:	learn: 0.8851533	total: 12.2s	remaining: 25.3s

972:	learn: 0.8847212	total: 12.2s	remaining: 25.3s
973:	learn: 0.8840006	total: 12.2s	remaining: 25.2s
974:	learn: 0.8833544	total: 12.3s	remaining: 25.2s
975:	learn: 0.8828304	total: 12.3s	remaining: 25.2s
976:	learn: 0.8822845	total: 12.3s	remaining: 25.2s
977:	learn: 0.8817773	total: 12.3s	remaining: 25.2s
978:	learn: 0.8812166	total: 12.3s	remaining: 25.2s
979:	learn: 0.8806958	total: 12.3s	remaining: 25.2s
980:	learn: 0.8802631	total: 12.3s	remaining: 25.2s
981:	learn: 0.8797025	total: 12.3s	remaining: 25.1s
982:	learn: 0.8790796	total: 12.4s	remaining: 25.1s
983:	learn: 0.8783518	total: 12.4s	remaining: 25.1s
984:	learn: 0.8777683	total: 12.4s	remaining: 25.1s
985:	learn: 0.8771471	total: 12.4s	remaining: 25.1s
986:	learn: 0.8764027	total: 12.4s	remaining: 25.1s
987:	learn: 0.8758510	total: 12.4s	remaining: 25.1s
988:	learn: 0.8754021	total: 12.4s	remaining: 25s
989:	learn: 0.8748432	total: 12.4s	remaining: 25s
990:	learn: 0.8744304	total: 12.5s	remaining: 25s
991:	learn: 0.8737976	total: 12.5s	remaining: 25s
992:	learn: 0.8734013	total: 12.5s	remaining: 25s
993:	learn: 0.8730332	total: 12.5s	remaining: 25s
994:	learn: 0.8724731	total: 12.5s	remaining: 25s
995:	learn: 0.8721204	total: 12.5s	remaining: 24.9s
996:	learn: 0.8715057	total: 12.5s	remaining: 24.9s
997:	learn: 0.8709735	total: 12.5s	remaining: 24.9s
998:	learn: 0.8705442	total: 12.5s	remaining: 24.9s
999:	learn: 0.8700070	total: 12.6s	remaining: 24.9s
1000:	learn: 0.8692658	total: 12.6s	remaining: 24.9s
1001:	learn: 0.8688402	total: 12.6s	remaining: 24.9s
1002:	learn: 0.8682278	total: 12.6s	remaining: 24.8s
1003:	learn: 0.8677097	total: 12.6s	remaining: 24.8s
1004:	learn: 0.8671433	total: 12.6s	remaining: 24.8s
1005:	learn: 0.8665942	total: 12.6s	remaining: 24.8s
1006:	learn: 0.8662995	total: 12.6s	remaining: 24.8s
1007:	learn: 0.8660388	total: 12.7s	remaining: 24.8s
1008:	learn: 0.8656838	total: 12.7s	remaining: 24.8s
1009:	learn: 0.8651174	total: 12.7s	remaining: 24.8s
1010:	learn: 0.8645926	total: 12.7s	remaining: 24.7s
1011:	learn: 0.8638991	total: 12.7s	remaining: 24.7s
1012:	learn: 0.8630028	total: 12.7s	remaining: 24.7s
1013:	learn: 0.8624995	total: 12.7s	remaining: 24.7s
1014:	learn: 0.8620169	total: 12.7s	remaining: 24.7s
1015:	learn: 0.8616745	total: 12.8s	remaining: 24.7s
1016:	learn: 0.8611622	total: 12.8s	remaining: 24.7s
1017:	learn: 0.8606494	total: 12.8s	remaining: 24.7s
1018:	learn: 0.8602660	total: 12.8s	remaining: 24.6s
1019:	learn: 0.8598070	total: 12.8s	remaining: 24.6s
1020:	learn: 0.8593244	total: 12.8s	remaining: 24.6s
1021:	learn: 0.8589008	total: 12.8s	remaining: 24.6s
1022:	learn: 0.8584206	total: 12.8s	remaining: 24.6s
1023:	learn: 0.8579886	total: 12.8s	remaining: 24.6s
1024:	learn: 0.8574143	total: 12.9s	remaining: 24.6s
1025:	learn: 0.8566569	total: 12.9s	remaining: 24.5s

1026:	learn: 0.8561578	total: 12.9s	remaining: 24.5s
1027:	learn: 0.8559400	total: 12.9s	remaining: 24.5s
1028:	learn: 0.8554832	total: 12.9s	remaining: 24.5s
1029:	learn: 0.8549848	total: 12.9s	remaining: 24.5s
1030:	learn: 0.8545606	total: 12.9s	remaining: 24.5s
1031:	learn: 0.8540605	total: 12.9s	remaining: 24.5s
1032:	learn: 0.8533826	total: 13s	remaining: 24.5s
1033:	learn: 0.8528822	total: 13s	remaining: 24.4s
1034:	learn: 0.8525456	total: 13s	remaining: 24.4s
1035:	learn: 0.8519158	total: 13s	remaining: 24.4s
1036:	learn: 0.8512995	total: 13s	remaining: 24.4s
1037:	learn: 0.8508544	total: 13s	remaining: 24.4s
1038:	learn: 0.8504340	total: 13s	remaining: 24.4s
1039:	learn: 0.8499974	total: 13s	remaining: 24.4s
1040:	learn: 0.8493851	total: 13.1s	remaining: 24.3s
1041:	learn: 0.8486239	total: 13.1s	remaining: 24.3s
1042:	learn: 0.8481285	total: 13.1s	remaining: 24.3s
1043:	learn: 0.8475173	total: 13.1s	remaining: 24.3s
1044:	learn: 0.8472011	total: 13.1s	remaining: 24.3s
1045:	learn: 0.8467949	total: 13.1s	remaining: 24.3s
1046:	learn: 0.8461342	total: 13.1s	remaining: 24.3s
1047:	learn: 0.8456441	total: 13.1s	remaining: 24.2s
1048:	learn: 0.8450384	total: 13.2s	remaining: 24.2s
1049:	learn: 0.8445215	total: 13.2s	remaining: 24.2s
1050:	learn: 0.8441747	total: 13.2s	remaining: 24.2s
1051:	learn: 0.8437362	total: 13.2s	remaining: 24.2s
1052:	learn: 0.8430908	total: 13.2s	remaining: 24.2s
1053:	learn: 0.8428553	total: 13.2s	remaining: 24.2s
1054:	learn: 0.8424443	total: 13.2s	remaining: 24.2s
1055:	learn: 0.8421148	total: 13.2s	remaining: 24.1s
1056:	learn: 0.8417163	total: 13.2s	remaining: 24.1s
1057:	learn: 0.8412143	total: 13.3s	remaining: 24.1s
1058:	learn: 0.8407192	total: 13.3s	remaining: 24.1s
1059:	learn: 0.8403341	total: 13.3s	remaining: 24.1s
1060:	learn: 0.8399522	total: 13.3s	remaining: 24.1s
1061:	learn: 0.8395643	total: 13.3s	remaining: 24.1s
1062:	learn: 0.8390975	total: 13.3s	remaining: 24.1s
1063:	learn: 0.8388759	total: 13.3s	remaining: 24.1s
1064:	learn: 0.8384826	total: 13.4s	remaining: 24.1s
1065:	learn: 0.8378354	total: 13.4s	remaining: 24s
1066:	learn: 0.8375204	total: 13.4s	remaining: 24s
1067:	learn: 0.8369429	total: 13.4s	remaining: 24s
1068:	learn: 0.8365307	total: 13.4s	remaining: 24s
1069:	learn: 0.8359430	total: 13.4s	remaining: 24s
1070:	learn: 0.8355106	total: 13.4s	remaining: 24s
1071:	learn: 0.8350016	total: 13.5s	remaining: 24s
1072:	learn: 0.8345563	total: 13.5s	remaining: 23.9s
1073:	learn: 0.8342401	total: 13.5s	remaining: 23.9s
1074:	learn: 0.8336267	total: 13.5s	remaining: 23.9s
1075:	learn: 0.8329407	total: 13.5s	remaining: 23.9s
1076:	learn: 0.8325277	total: 13.5s	remaining: 23.9s
1077:	learn: 0.8323276	total: 13.5s	remaining: 23.9s
1078:	learn: 0.8318695	total: 13.5s	remaining: 23.9s
1079:	learn: 0.8314734	total: 13.5s	remaining: 23.9s

1080:	learn: 0.8310703	total: 13.6s	remaining: 23.8s
1081:	learn: 0.8307006	total: 13.6s	remaining: 23.8s
1082:	learn: 0.8301927	total: 13.6s	remaining: 23.8s
1083:	learn: 0.8296776	total: 13.6s	remaining: 23.8s
1084:	learn: 0.8291636	total: 13.6s	remaining: 23.8s
1085:	learn: 0.8286326	total: 13.6s	remaining: 23.8s
1086:	learn: 0.8282084	total: 13.6s	remaining: 23.8s
1087:	learn: 0.8275832	total: 13.6s	remaining: 23.8s
1088:	learn: 0.8271816	total: 13.7s	remaining: 23.7s
1089:	learn: 0.8267329	total: 13.7s	remaining: 23.7s
1090:	learn: 0.8263266	total: 13.7s	remaining: 23.7s
1091:	learn: 0.8259233	total: 13.7s	remaining: 23.7s
1092:	learn: 0.8253716	total: 13.7s	remaining: 23.7s
1093:	learn: 0.8249989	total: 13.7s	remaining: 23.7s
1094:	learn: 0.8243179	total: 13.7s	remaining: 23.7s
1095:	learn: 0.8239842	total: 13.7s	remaining: 23.6s
1096:	learn: 0.8234270	total: 13.8s	remaining: 23.6s
1097:	learn: 0.8230778	total: 13.8s	remaining: 23.6s
1098:	learn: 0.8228428	total: 13.8s	remaining: 23.6s
1099:	learn: 0.8224812	total: 13.8s	remaining: 23.6s
1100:	learn: 0.8219434	total: 13.8s	remaining: 23.6s
1101:	learn: 0.8215997	total: 13.8s	remaining: 23.6s
1102:	learn: 0.8210171	total: 13.8s	remaining: 23.6s
1103:	learn: 0.8205812	total: 13.8s	remaining: 23.5s
1104:	learn: 0.8201568	total: 13.9s	remaining: 23.5s
1105:	learn: 0.8196694	total: 13.9s	remaining: 23.5s
1106:	learn: 0.8192045	total: 13.9s	remaining: 23.5s
1107:	learn: 0.8186690	total: 13.9s	remaining: 23.5s
1108:	learn: 0.8182149	total: 13.9s	remaining: 23.5s
1109:	learn: 0.8178490	total: 13.9s	remaining: 23.5s
1110:	learn: 0.8174056	total: 13.9s	remaining: 23.4s
1111:	learn: 0.8169459	total: 13.9s	remaining: 23.4s
1112:	learn: 0.8165104	total: 13.9s	remaining: 23.4s
1113:	learn: 0.8159994	total: 14s	remaining: 23.4s
1114:	learn: 0.8156212	total: 14s	remaining: 23.4s
1115:	learn: 0.8153982	total: 14s	remaining: 23.4s
1116:	learn: 0.8149057	total: 14s	remaining: 23.4s
1117:	learn: 0.8143992	total: 14s	remaining: 23.4s
1118:	learn: 0.8142212	total: 14s	remaining: 23.3s
1119:	learn: 0.8138720	total: 14s	remaining: 23.3s
1120:	learn: 0.8135674	total: 14s	remaining: 23.3s
1121:	learn: 0.8131135	total: 14.1s	remaining: 23.3s
1122:	learn: 0.8126530	total: 14.1s	remaining: 23.3s
1123:	learn: 0.8121692	total: 14.1s	remaining: 23.3s
1124:	learn: 0.8118691	total: 14.1s	remaining: 23.3s
1125:	learn: 0.8114588	total: 14.1s	remaining: 23.3s
1126:	learn: 0.8110554	total: 14.1s	remaining: 23.2s
1127:	learn: 0.8106961	total: 14.1s	remaining: 23.2s
1128:	learn: 0.8103343	total: 14.1s	remaining: 23.2s
1129:	learn: 0.8101533	total: 14.2s	remaining: 23.2s
1130:	learn: 0.8097874	total: 14.2s	remaining: 23.2s
1131:	learn: 0.8093700	total: 14.2s	remaining: 23.2s
1132:	learn: 0.8089162	total: 14.2s	remaining: 23.2s
1133:	learn: 0.8084991	total: 14.2s	remaining: 23.2s

1134:	learn: 0.8083367	total: 14.2s	remaining: 23.1s
1135:	learn: 0.8078430	total: 14.2s	remaining: 23.1s
1136:	learn: 0.8072325	total: 14.2s	remaining: 23.1s
1137:	learn: 0.8068269	total: 14.3s	remaining: 23.1s
1138:	learn: 0.8062812	total: 14.3s	remaining: 23.1s
1139:	learn: 0.8059478	total: 14.3s	remaining: 23.1s
1140:	learn: 0.8055920	total: 14.3s	remaining: 23.1s
1141:	learn: 0.8052224	total: 14.3s	remaining: 23s
1142:	learn: 0.8049030	total: 14.3s	remaining: 23s
1143:	learn: 0.8042915	total: 14.3s	remaining: 23s
1144:	learn: 0.8038116	total: 14.3s	remaining: 23s
1145:	learn: 0.8034097	total: 14.4s	remaining: 23s
1146:	learn: 0.8030421	total: 14.4s	remaining: 23s
1147:	learn: 0.8025751	total: 14.4s	remaining: 23s
1148:	learn: 0.8022304	total: 14.4s	remaining: 23s
1149:	learn: 0.8018607	total: 14.4s	remaining: 22.9s
1150:	learn: 0.8015290	total: 14.4s	remaining: 22.9s
1151:	learn: 0.8012033	total: 14.4s	remaining: 22.9s
1152:	learn: 0.8007957	total: 14.4s	remaining: 22.9s
1153:	learn: 0.8003394	total: 14.5s	remaining: 22.9s
1154:	learn: 0.7999007	total: 14.5s	remaining: 22.9s
1155:	learn: 0.7995956	total: 14.5s	remaining: 22.9s
1156:	learn: 0.7990945	total: 14.5s	remaining: 22.9s
1157:	learn: 0.7987798	total: 14.5s	remaining: 22.8s
1158:	learn: 0.7984607	total: 14.5s	remaining: 22.8s
1159:	learn: 0.7981075	total: 14.5s	remaining: 22.8s
1160:	learn: 0.7977619	total: 14.5s	remaining: 22.8s
1161:	learn: 0.7973231	total: 14.5s	remaining: 22.8s
1162:	learn: 0.7971724	total: 14.6s	remaining: 22.8s
1163:	learn: 0.7967823	total: 14.6s	remaining: 22.8s
1164:	learn: 0.7963120	total: 14.6s	remaining: 22.8s
1165:	learn: 0.7957708	total: 14.6s	remaining: 22.7s
1166:	learn: 0.7953346	total: 14.6s	remaining: 22.7s
1167:	learn: 0.7949157	total: 14.6s	remaining: 22.7s
1168:	learn: 0.7945708	total: 14.6s	remaining: 22.7s
1169:	learn: 0.7942083	total: 14.7s	remaining: 22.7s
1170:	learn: 0.7938574	total: 14.7s	remaining: 22.7s
1171:	learn: 0.7933618	total: 14.7s	remaining: 22.7s
1172:	learn: 0.7929329	total: 14.7s	remaining: 22.6s
1173:	learn: 0.7923732	total: 14.7s	remaining: 22.6s
1174:	learn: 0.7917993	total: 14.7s	remaining: 22.6s
1175:	learn: 0.7909457	total: 14.7s	remaining: 22.6s
1176:	learn: 0.7904650	total: 14.7s	remaining: 22.6s
1177:	learn: 0.7900635	total: 14.7s	remaining: 22.6s
1178:	learn: 0.7895827	total: 14.8s	remaining: 22.6s
1179:	learn: 0.7891219	total: 14.8s	remaining: 22.6s
1180:	learn: 0.7886625	total: 14.8s	remaining: 22.5s
1181:	learn: 0.7881281	total: 14.8s	remaining: 22.5s
1182:	learn: 0.7879309	total: 14.8s	remaining: 22.5s
1183:	learn: 0.7878098	total: 14.8s	remaining: 22.5s
1184:	learn: 0.7874765	total: 14.8s	remaining: 22.5s
1185:	learn: 0.7870597	total: 14.8s	remaining: 22.5s
1186:	learn: 0.7865713	total: 14.9s	remaining: 22.5s
1187:	learn: 0.7863038	total: 14.9s	remaining: 22.4s

1188:	learn: 0.7860305	total: 14.9s	remaining: 22.4s
1189:	learn: 0.7856747	total: 14.9s	remaining: 22.4s
1190:	learn: 0.7854473	total: 14.9s	remaining: 22.4s
1191:	learn: 0.7850980	total: 14.9s	remaining: 22.4s
1192:	learn: 0.7847507	total: 14.9s	remaining: 22.4s
1193:	learn: 0.7843024	total: 14.9s	remaining: 22.4s
1194:	learn: 0.7839125	total: 14.9s	remaining: 22.4s
1195:	learn: 0.7835368	total: 15s	remaining: 22.3s
1196:	learn: 0.7832375	total: 15s	remaining: 22.3s
1197:	learn: 0.7826542	total: 15s	remaining: 22.3s
1198:	learn: 0.7823157	total: 15s	remaining: 22.3s
1199:	learn: 0.7819473	total: 15s	remaining: 22.3s
1200:	learn: 0.7817473	total: 15s	remaining: 22.3s
1201:	learn: 0.7813493	total: 15s	remaining: 22.3s
1202:	learn: 0.7809511	total: 15s	remaining: 22.2s
1203:	learn: 0.7805562	total: 15.1s	remaining: 22.2s
1204:	learn: 0.7801419	total: 15.1s	remaining: 22.2s
1205:	learn: 0.7798304	total: 15.1s	remaining: 22.2s
1206:	learn: 0.7794893	total: 15.1s	remaining: 22.2s
1207:	learn: 0.7789933	total: 15.1s	remaining: 22.2s
1208:	learn: 0.7786725	total: 15.1s	remaining: 22.2s
1209:	learn: 0.7779818	total: 15.1s	remaining: 22.2s
1210:	learn: 0.7775965	total: 15.1s	remaining: 22.1s
1211:	learn: 0.7771845	total: 15.2s	remaining: 22.1s
1212:	learn: 0.7768135	total: 15.2s	remaining: 22.1s
1213:	learn: 0.7764520	total: 15.2s	remaining: 22.1s
1214:	learn: 0.7760611	total: 15.2s	remaining: 22.1s
1215:	learn: 0.7753359	total: 15.2s	remaining: 22.1s
1216:	learn: 0.7746093	total: 15.2s	remaining: 22.1s
1217:	learn: 0.7743764	total: 15.2s	remaining: 22.1s
1218:	learn: 0.7739389	total: 15.2s	remaining: 22s
1219:	learn: 0.7735115	total: 15.3s	remaining: 22s
1220:	learn: 0.7730074	total: 15.3s	remaining: 22s
1221:	learn: 0.7724733	total: 15.3s	remaining: 22s
1222:	learn: 0.7721285	total: 15.3s	remaining: 22s
1223:	learn: 0.7718957	total: 15.3s	remaining: 22s
1224:	learn: 0.7711890	total: 15.3s	remaining: 22s
1225:	learn: 0.7706873	total: 15.3s	remaining: 22s
1226:	learn: 0.7702840	total: 15.3s	remaining: 21.9s
1227:	learn: 0.7699176	total: 15.4s	remaining: 21.9s
1228:	learn: 0.7694674	total: 15.4s	remaining: 21.9s
1229:	learn: 0.7690422	total: 15.4s	remaining: 21.9s
1230:	learn: 0.7687016	total: 15.4s	remaining: 21.9s
1231:	learn: 0.7683318	total: 15.4s	remaining: 21.9s
1232:	learn: 0.7678921	total: 15.4s	remaining: 21.9s
1233:	learn: 0.7675511	total: 15.4s	remaining: 21.9s
1234:	learn: 0.7671635	total: 15.4s	remaining: 21.8s
1235:	learn: 0.7667787	total: 15.5s	remaining: 21.8s
1236:	learn: 0.7665603	total: 15.5s	remaining: 21.8s
1237:	learn: 0.7660969	total: 15.5s	remaining: 21.8s
1238:	learn: 0.7658130	total: 15.5s	remaining: 21.8s
1239:	learn: 0.7654785	total: 15.5s	remaining: 21.8s
1240:	learn: 0.7650092	total: 15.5s	remaining: 21.8s
1241:	learn: 0.7646209	total: 15.5s	remaining: 21.7s

1242:	learn: 0.7643932	total: 15.5s	remaining: 21.7s
1243:	learn: 0.7640489	total: 15.5s	remaining: 21.7s
1244:	learn: 0.7637957	total: 15.6s	remaining: 21.7s
1245:	learn: 0.7633690	total: 15.6s	remaining: 21.7s
1246:	learn: 0.7630226	total: 15.6s	remaining: 21.7s
1247:	learn: 0.7628080	total: 15.6s	remaining: 21.7s
1248:	learn: 0.7623940	total: 15.6s	remaining: 21.7s
1249:	learn: 0.7620669	total: 15.6s	remaining: 21.6s
1250:	learn: 0.7617328	total: 15.6s	remaining: 21.6s
1251:	learn: 0.7613489	total: 15.7s	remaining: 21.6s
1252:	learn: 0.7609769	total: 15.7s	remaining: 21.6s
1253:	learn: 0.7606308	total: 15.7s	remaining: 21.6s
1254:	learn: 0.7603490	total: 15.7s	remaining: 21.6s
1255:	learn: 0.7600780	total: 15.7s	remaining: 21.6s
1256:	learn: 0.7597088	total: 15.7s	remaining: 21.6s
1257:	learn: 0.7592745	total: 15.7s	remaining: 21.6s
1258:	learn: 0.7589457	total: 15.7s	remaining: 21.5s
1259:	learn: 0.7586759	total: 15.8s	remaining: 21.5s
1260:	learn: 0.7584339	total: 15.8s	remaining: 21.5s
1261:	learn: 0.7580489	total: 15.8s	remaining: 21.5s
1262:	learn: 0.7577552	total: 15.8s	remaining: 21.5s
1263:	learn: 0.7571566	total: 15.8s	remaining: 21.5s
1264:	learn: 0.7568259	total: 15.8s	remaining: 21.5s
1265:	learn: 0.7565275	total: 15.8s	remaining: 21.5s
1266:	learn: 0.7561419	total: 15.8s	remaining: 21.4s
1267:	learn: 0.7559211	total: 15.9s	remaining: 21.4s
1268:	learn: 0.7555366	total: 15.9s	remaining: 21.4s
1269:	learn: 0.7551839	total: 15.9s	remaining: 21.4s
1270:	learn: 0.7548652	total: 15.9s	remaining: 21.4s
1271:	learn: 0.7544894	total: 15.9s	remaining: 21.4s
1272:	learn: 0.7541140	total: 15.9s	remaining: 21.4s
1273:	learn: 0.7537685	total: 15.9s	remaining: 21.3s
1274:	learn: 0.7534133	total: 15.9s	remaining: 21.3s
1275:	learn: 0.7529195	total: 15.9s	remaining: 21.3s
1276:	learn: 0.7524585	total: 16s	remaining: 21.3s
1277:	learn: 0.7521931	total: 16s	remaining: 21.3s
1278:	learn: 0.7518435	total: 16s	remaining: 21.3s
1279:	learn: 0.7516710	total: 16s	remaining: 21.3s
1280:	learn: 0.7513838	total: 16s	remaining: 21.3s
1281:	learn: 0.7510991	total: 16s	remaining: 21.2s
1282:	learn: 0.7508002	total: 16s	remaining: 21.2s
1283:	learn: 0.7504322	total: 16s	remaining: 21.2s
1284:	learn: 0.7500855	total: 16.1s	remaining: 21.2s
1285:	learn: 0.7498549	total: 16.1s	remaining: 21.2s
1286:	learn: 0.7495826	total: 16.1s	remaining: 21.2s
1287:	learn: 0.7493076	total: 16.1s	remaining: 21.2s
1288:	learn: 0.7489212	total: 16.1s	remaining: 21.2s
1289:	learn: 0.7487746	total: 16.1s	remaining: 21.1s
1290:	learn: 0.7484170	total: 16.1s	remaining: 21.1s
1291:	learn: 0.7481426	total: 16.1s	remaining: 21.1s
1292:	learn: 0.7478002	total: 16.2s	remaining: 21.1s
1293:	learn: 0.7475558	total: 16.2s	remaining: 21.1s
1294:	learn: 0.7469330	total: 16.2s	remaining: 21.1s
1295:	learn: 0.7464369	total: 16.2s	remaining: 21.1s

1296:	learn: 0.7460291	total: 16.2s	remaining: 21s
1297:	learn: 0.7456104	total: 16.2s	remaining: 21s
1298:	learn: 0.7452013	total: 16.2s	remaining: 21s
1299:	learn: 0.7447855	total: 16.2s	remaining: 21s
1300:	learn: 0.7444270	total: 16.2s	remaining: 21s
1301:	learn: 0.7440346	total: 16.3s	remaining: 21s
1302:	learn: 0.7435887	total: 16.3s	remaining: 21s
1303:	learn: 0.7433710	total: 16.3s	remaining: 21s
1304:	learn: 0.7428729	total: 16.3s	remaining: 20.9s
1305:	learn: 0.7427086	total: 16.3s	remaining: 20.9s
1306:	learn: 0.7424375	total: 16.3s	remaining: 20.9s
1307:	learn: 0.7422395	total: 16.3s	remaining: 20.9s
1308:	learn: 0.7418718	total: 16.3s	remaining: 20.9s
1309:	learn: 0.7414107	total: 16.4s	remaining: 20.9s
1310:	learn: 0.7411769	total: 16.4s	remaining: 20.9s
1311:	learn: 0.7409132	total: 16.4s	remaining: 20.8s
1312:	learn: 0.7402881	total: 16.4s	remaining: 20.8s
1313:	learn: 0.7399321	total: 16.4s	remaining: 20.8s
1314:	learn: 0.7395258	total: 16.4s	remaining: 20.8s
1315:	learn: 0.7391550	total: 16.4s	remaining: 20.8s
1316:	learn: 0.7388991	total: 16.4s	remaining: 20.8s
1317:	learn: 0.7386370	total: 16.5s	remaining: 20.8s
1318:	learn: 0.7383113	total: 16.5s	remaining: 20.8s
1319:	learn: 0.7379561	total: 16.5s	remaining: 20.7s
1320:	learn: 0.7375587	total: 16.5s	remaining: 20.7s
1321:	learn: 0.7371219	total: 16.5s	remaining: 20.7s
1322:	learn: 0.7369154	total: 16.5s	remaining: 20.7s
1323:	learn: 0.7364860	total: 16.5s	remaining: 20.7s
1324:	learn: 0.7362464	total: 16.5s	remaining: 20.7s
1325:	learn: 0.7359031	total: 16.5s	remaining: 20.7s
1326:	learn: 0.7357082	total: 16.6s	remaining: 20.7s
1327:	learn: 0.7354061	total: 16.6s	remaining: 20.6s
1328:	learn: 0.7350352	total: 16.6s	remaining: 20.6s
1329:	learn: 0.7348508	total: 16.6s	remaining: 20.6s
1330:	learn: 0.7345537	total: 16.6s	remaining: 20.6s
1331:	learn: 0.7341033	total: 16.6s	remaining: 20.6s
1332:	learn: 0.7336021	total: 16.6s	remaining: 20.6s
1333:	learn: 0.7333185	total: 16.6s	remaining: 20.6s
1334:	learn: 0.7331406	total: 16.7s	remaining: 20.5s
1335:	learn: 0.7328415	total: 16.7s	remaining: 20.5s
1336:	learn: 0.7324019	total: 16.7s	remaining: 20.5s
1337:	learn: 0.7319972	total: 16.7s	remaining: 20.5s
1338:	learn: 0.7317347	total: 16.7s	remaining: 20.5s
1339:	learn: 0.7314324	total: 16.7s	remaining: 20.5s
1340:	learn: 0.7311415	total: 16.7s	remaining: 20.5s
1341:	learn: 0.7308602	total: 16.7s	remaining: 20.5s
1342:	learn: 0.7305088	total: 16.8s	remaining: 20.4s
1343:	learn: 0.7301656	total: 16.8s	remaining: 20.4s
1344:	learn: 0.7299235	total: 16.8s	remaining: 20.4s
1345:	learn: 0.7296328	total: 16.8s	remaining: 20.4s
1346:	learn: 0.7293556	total: 16.8s	remaining: 20.4s
1347:	learn: 0.7290209	total: 16.8s	remaining: 20.4s
1348:	learn: 0.7287564	total: 16.8s	remaining: 20.4s
1349:	learn: 0.7284749	total: 16.8s	remaining: 20.4s

1350:	learn: 0.7280937	total: 16.9s	remaining: 20.4s
1351:	learn: 0.7278827	total: 16.9s	remaining: 20.3s
1352:	learn: 0.7275116	total: 16.9s	remaining: 20.3s
1353:	learn: 0.7272163	total: 16.9s	remaining: 20.3s
1354:	learn: 0.7269785	total: 16.9s	remaining: 20.3s
1355:	learn: 0.7267669	total: 16.9s	remaining: 20.3s
1356:	learn: 0.7264042	total: 16.9s	remaining: 20.3s
1357:	learn: 0.7259419	total: 17s	remaining: 20.3s
1358:	learn: 0.7255531	total: 17s	remaining: 20.3s
1359:	learn: 0.7252419	total: 17s	remaining: 20.2s
1360:	learn: 0.7249462	total: 17s	remaining: 20.2s
1361:	learn: 0.7246045	total: 17s	remaining: 20.2s
1362:	learn: 0.7242734	total: 17s	remaining: 20.2s
1363:	learn: 0.7240403	total: 17s	remaining: 20.2s
1364:	learn: 0.7237541	total: 17s	remaining: 20.2s
1365:	learn: 0.7234331	total: 17.1s	remaining: 20.2s
1366:	learn: 0.7231458	total: 17.1s	remaining: 20.2s
1367:	learn: 0.7227962	total: 17.1s	remaining: 20.2s
1368:	learn: 0.7226139	total: 17.1s	remaining: 20.1s
1369:	learn: 0.7222233	total: 17.1s	remaining: 20.1s
1370:	learn: 0.7219873	total: 17.1s	remaining: 20.1s
1371:	learn: 0.7216442	total: 17.1s	remaining: 20.1s
1372:	learn: 0.7213338	total: 17.1s	remaining: 20.1s
1373:	learn: 0.7210385	total: 17.2s	remaining: 20.1s
1374:	learn: 0.7208295	total: 17.2s	remaining: 20.1s
1375:	learn: 0.7205558	total: 17.2s	remaining: 20.1s
1376:	learn: 0.7203332	total: 17.2s	remaining: 20s
1377:	learn: 0.7201021	total: 17.2s	remaining: 20s
1378:	learn: 0.7199359	total: 17.2s	remaining: 20s
1379:	learn: 0.7196423	total: 17.2s	remaining: 20s
1380:	learn: 0.7192090	total: 17.2s	remaining: 20s
1381:	learn: 0.7189239	total: 17.3s	remaining: 20s
1382:	learn: 0.7186193	total: 17.3s	remaining: 20s
1383:	learn: 0.7182375	total: 17.3s	remaining: 20s
1384:	learn: 0.7180655	total: 17.3s	remaining: 19.9s
1385:	learn: 0.7177872	total: 17.3s	remaining: 19.9s
1386:	learn: 0.7175170	total: 17.3s	remaining: 19.9s
1387:	learn: 0.7172115	total: 17.3s	remaining: 19.9s
1388:	learn: 0.7169704	total: 17.3s	remaining: 19.9s
1389:	learn: 0.7167694	total: 17.4s	remaining: 19.9s
1390:	learn: 0.7164608	total: 17.4s	remaining: 19.9s
1391:	learn: 0.7161974	total: 17.4s	remaining: 19.9s
1392:	learn: 0.7160074	total: 17.4s	remaining: 19.8s
1393:	learn: 0.7156607	total: 17.4s	remaining: 19.8s
1394:	learn: 0.7154178	total: 17.4s	remaining: 19.8s
1395:	learn: 0.7151595	total: 17.4s	remaining: 19.8s
1396:	learn: 0.7148814	total: 17.4s	remaining: 19.8s
1397:	learn: 0.7146661	total: 17.5s	remaining: 19.8s
1398:	learn: 0.7144408	total: 17.5s	remaining: 19.8s
1399:	learn: 0.7141946	total: 17.5s	remaining: 19.8s
1400:	learn: 0.7137983	total: 17.5s	remaining: 19.7s
1401:	learn: 0.7136666	total: 17.5s	remaining: 19.7s
1402:	learn: 0.7134146	total: 17.5s	remaining: 19.7s
1403:	learn: 0.7131588	total: 17.5s	remaining: 19.7s

1404:	learn: 0.7129487	total: 17.5s	remaining: 19.7s
1405:	learn: 0.7126854	total: 17.6s	remaining: 19.7s
1406:	learn: 0.7123624	total: 17.6s	remaining: 19.7s
1407:	learn: 0.7121854	total: 17.6s	remaining: 19.7s
1408:	learn: 0.7119465	total: 17.6s	remaining: 19.6s
1409:	learn: 0.7116750	total: 17.6s	remaining: 19.6s
1410:	learn: 0.7114383	total: 17.6s	remaining: 19.6s
1411:	learn: 0.7110986	total: 17.6s	remaining: 19.6s
1412:	learn: 0.7108756	total: 17.6s	remaining: 19.6s
1413:	learn: 0.7106451	total: 17.7s	remaining: 19.6s
1414:	learn: 0.7104465	total: 17.7s	remaining: 19.6s
1415:	learn: 0.7098932	total: 17.7s	remaining: 19.5s
1416:	learn: 0.7095809	total: 17.7s	remaining: 19.5s
1417:	learn: 0.7091339	total: 17.7s	remaining: 19.5s
1418:	learn: 0.7086402	total: 17.7s	remaining: 19.5s
1419:	learn: 0.7084059	total: 17.7s	remaining: 19.5s
1420:	learn: 0.7081452	total: 17.7s	remaining: 19.5s
1421:	learn: 0.7079025	total: 17.7s	remaining: 19.5s
1422:	learn: 0.7077091	total: 17.8s	remaining: 19.5s
1423:	learn: 0.7074461	total: 17.8s	remaining: 19.4s
1424:	learn: 0.7071848	total: 17.8s	remaining: 19.4s
1425:	learn: 0.7069449	total: 17.8s	remaining: 19.4s
1426:	learn: 0.7063479	total: 17.8s	remaining: 19.4s
1427:	learn: 0.7060011	total: 17.8s	remaining: 19.4s
1428:	learn: 0.7055450	total: 17.8s	remaining: 19.4s
1429:	learn: 0.7051369	total: 17.8s	remaining: 19.4s
1430:	learn: 0.7048348	total: 17.9s	remaining: 19.4s
1431:	learn: 0.7045668	total: 17.9s	remaining: 19.3s
1432:	learn: 0.7043010	total: 17.9s	remaining: 19.3s
1433:	learn: 0.7040768	total: 17.9s	remaining: 19.3s
1434:	learn: 0.7037950	total: 17.9s	remaining: 19.3s
1435:	learn: 0.7035048	total: 17.9s	remaining: 19.3s
1436:	learn: 0.7031789	total: 17.9s	remaining: 19.3s
1437:	learn: 0.7029391	total: 17.9s	remaining: 19.3s
1438:	learn: 0.7027346	total: 18s	remaining: 19.3s
1439:	learn: 0.7024165	total: 18s	remaining: 19.2s
1440:	learn: 0.7021134	total: 18s	remaining: 19.2s
1441:	learn: 0.7018307	total: 18s	remaining: 19.2s
1442:	learn: 0.7015434	total: 18s	remaining: 19.2s
1443:	learn: 0.7011345	total: 18s	remaining: 19.2s
1444:	learn: 0.7008573	total: 18s	remaining: 19.2s
1445:	learn: 0.7006189	total: 18s	remaining: 19.2s
1446:	learn: 0.7004001	total: 18.1s	remaining: 19.1s
1447:	learn: 0.7000694	total: 18.1s	remaining: 19.1s
1448:	learn: 0.6998646	total: 18.1s	remaining: 19.1s
1449:	learn: 0.6997081	total: 18.1s	remaining: 19.1s
1450:	learn: 0.6994255	total: 18.1s	remaining: 19.1s
1451:	learn: 0.6992742	total: 18.1s	remaining: 19.1s
1452:	learn: 0.6990055	total: 18.1s	remaining: 19.1s
1453:	learn: 0.6987831	total: 18.1s	remaining: 19.1s
1454:	learn: 0.6983620	total: 18.2s	remaining: 19.1s
1455:	learn: 0.6980961	total: 18.2s	remaining: 19s
1456:	learn: 0.6978435	total: 18.2s	remaining: 19s
1457:	learn: 0.6975427	total: 18.2s	remaining: 19s

1458:	learn: 0.6973029	total: 18.2s	remaining: 19s
1459:	learn: 0.6969660	total: 18.2s	remaining: 19s
1460:	learn: 0.6966064	total: 18.2s	remaining: 19s
1461:	learn: 0.6962781	total: 18.2s	remaining: 19s
1462:	learn: 0.6960609	total: 18.3s	remaining: 19s
1463:	learn: 0.6957941	total: 18.3s	remaining: 18.9s
1464:	learn: 0.6955656	total: 18.3s	remaining: 18.9s
1465:	learn: 0.6951128	total: 18.3s	remaining: 18.9s
1466:	learn: 0.6948535	total: 18.3s	remaining: 18.9s
1467:	learn: 0.6945701	total: 18.3s	remaining: 18.9s
1468:	learn: 0.6943043	total: 18.3s	remaining: 18.9s
1469:	learn: 0.6940397	total: 18.3s	remaining: 18.9s
1470:	learn: 0.6935618	total: 18.3s	remaining: 18.8s
1471:	learn: 0.6933503	total: 18.4s	remaining: 18.8s
1472:	learn: 0.6928933	total: 18.4s	remaining: 18.8s
1473:	learn: 0.6926546	total: 18.4s	remaining: 18.8s
1474:	learn: 0.6924740	total: 18.4s	remaining: 18.8s
1475:	learn: 0.6922173	total: 18.4s	remaining: 18.8s
1476:	learn: 0.6919006	total: 18.4s	remaining: 18.8s
1477:	learn: 0.6916885	total: 18.4s	remaining: 18.8s
1478:	learn: 0.6912428	total: 18.4s	remaining: 18.7s
1479:	learn: 0.6910033	total: 18.5s	remaining: 18.7s
1480:	learn: 0.6907185	total: 18.5s	remaining: 18.7s
1481:	learn: 0.6905013	total: 18.5s	remaining: 18.7s
1482:	learn: 0.6902357	total: 18.5s	remaining: 18.7s
1483:	learn: 0.6899672	total: 18.5s	remaining: 18.7s
1484:	learn: 0.6896657	total: 18.5s	remaining: 18.7s
1485:	learn: 0.6893219	total: 18.5s	remaining: 18.6s
1486:	learn: 0.6890479	total: 18.5s	remaining: 18.6s
1487:	learn: 0.6888413	total: 18.6s	remaining: 18.6s
1488:	learn: 0.6886017	total: 18.6s	remaining: 18.6s
1489:	learn: 0.6883316	total: 18.6s	remaining: 18.6s
1490:	learn: 0.6880519	total: 18.6s	remaining: 18.6s
1491:	learn: 0.6876955	total: 18.6s	remaining: 18.6s
1492:	learn: 0.6874712	total: 18.6s	remaining: 18.6s
1493:	learn: 0.6872420	total: 18.6s	remaining: 18.5s
1494:	learn: 0.6871043	total: 18.6s	remaining: 18.5s
1495:	learn: 0.6867552	total: 18.6s	remaining: 18.5s
1496:	learn: 0.6865268	total: 18.7s	remaining: 18.5s
1497:	learn: 0.6861376	total: 18.7s	remaining: 18.5s
1498:	learn: 0.6859345	total: 18.7s	remaining: 18.5s
1499:	learn: 0.6855774	total: 18.7s	remaining: 18.5s
1500:	learn: 0.6852760	total: 18.7s	remaining: 18.5s
1501:	learn: 0.6848878	total: 18.7s	remaining: 18.4s
1502:	learn: 0.6846291	total: 18.7s	remaining: 18.4s
1503:	learn: 0.6843176	total: 18.7s	remaining: 18.4s
1504:	learn: 0.6840033	total: 18.7s	remaining: 18.4s
1505:	learn: 0.6838405	total: 18.8s	remaining: 18.4s
1506:	learn: 0.6836733	total: 18.8s	remaining: 18.4s
1507:	learn: 0.6835149	total: 18.8s	remaining: 18.4s
1508:	learn: 0.6833705	total: 18.8s	remaining: 18.3s
1509:	learn: 0.6831535	total: 18.8s	remaining: 18.3s
1510:	learn: 0.6829369	total: 18.8s	remaining: 18.3s
1511:	learn: 0.6826478	total: 18.8s	remaining: 18.3s

1512:	learn: 0.6824751	total: 18.8s	remaining: 18.3s
1513:	learn: 0.6822520	total: 18.9s	remaining: 18.3s
1514:	learn: 0.6820698	total: 18.9s	remaining: 18.3s
1515:	learn: 0.6817604	total: 18.9s	remaining: 18.3s
1516:	learn: 0.6815339	total: 18.9s	remaining: 18.2s
1517:	learn: 0.6812925	total: 18.9s	remaining: 18.2s
1518:	learn: 0.6810879	total: 18.9s	remaining: 18.2s
1519:	learn: 0.6808831	total: 18.9s	remaining: 18.2s
1520:	learn: 0.6806710	total: 18.9s	remaining: 18.2s
1521:	learn: 0.6803733	total: 18.9s	remaining: 18.2s
1522:	learn: 0.6801861	total: 19s	remaining: 18.2s
1523:	learn: 0.6799082	total: 19s	remaining: 18.2s
1524:	learn: 0.6796431	total: 19s	remaining: 18.1s
1525:	learn: 0.6793730	total: 19s	remaining: 18.1s
1526:	learn: 0.6791382	total: 19s	remaining: 18.1s
1527:	learn: 0.6787629	total: 19s	remaining: 18.1s
1528:	learn: 0.6785286	total: 19s	remaining: 18.1s
1529:	learn: 0.6782786	total: 19.1s	remaining: 18.1s
1530:	learn: 0.6779892	total: 19.1s	remaining: 18.1s
1531:	learn: 0.6777866	total: 19.1s	remaining: 18.1s
1532:	learn: 0.6775573	total: 19.1s	remaining: 18s
1533:	learn: 0.6773268	total: 19.1s	remaining: 18s
1534:	learn: 0.6771156	total: 19.1s	remaining: 18s
1535:	learn: 0.6769228	total: 19.1s	remaining: 18s
1536:	learn: 0.6767128	total: 19.1s	remaining: 18s
1537:	learn: 0.6765116	total: 19.2s	remaining: 18s
1538:	learn: 0.6763331	total: 19.2s	remaining: 18s
1539:	learn: 0.6761389	total: 19.2s	remaining: 18s
1540:	learn: 0.6757446	total: 19.2s	remaining: 17.9s
1541:	learn: 0.6754979	total: 19.2s	remaining: 17.9s
1542:	learn: 0.6753407	total: 19.2s	remaining: 17.9s
1543:	learn: 0.6751363	total: 19.2s	remaining: 17.9s
1544:	learn: 0.6748680	total: 19.2s	remaining: 17.9s
1545:	learn: 0.6746404	total: 19.3s	remaining: 17.9s
1546:	learn: 0.6742791	total: 19.3s	remaining: 17.9s
1547:	learn: 0.6741359	total: 19.3s	remaining: 17.9s
1548:	learn: 0.6739706	total: 19.3s	remaining: 17.8s
1549:	learn: 0.6736199	total: 19.3s	remaining: 17.8s
1550:	learn: 0.6734052	total: 19.3s	remaining: 17.8s
1551:	learn: 0.6731899	total: 19.3s	remaining: 17.8s
1552:	learn: 0.6729635	total: 19.3s	remaining: 17.8s
1553:	learn: 0.6726125	total: 19.4s	remaining: 17.8s
1554:	learn: 0.6723966	total: 19.4s	remaining: 17.8s
1555:	learn: 0.6721235	total: 19.4s	remaining: 17.8s
1556:	learn: 0.6718409	total: 19.4s	remaining: 17.7s
1557:	learn: 0.6717012	total: 19.4s	remaining: 17.7s
1558:	learn: 0.6713764	total: 19.4s	remaining: 17.7s
1559:	learn: 0.6710821	total: 19.4s	remaining: 17.7s
1560:	learn: 0.6708641	total: 19.4s	remaining: 17.7s
1561:	learn: 0.6706303	total: 19.5s	remaining: 17.7s
1562:	learn: 0.6703999	total: 19.5s	remaining: 17.7s
1563:	learn: 0.6699980	total: 19.5s	remaining: 17.7s
1564:	learn: 0.6698089	total: 19.5s	remaining: 17.6s
1565:	learn: 0.6696454	total: 19.5s	remaining: 17.6s

1566:	learn: 0.6693758	total: 19.5s	remaining: 17.6s
1567:	learn: 0.6691535	total: 19.5s	remaining: 17.6s
1568:	learn: 0.6689877	total: 19.5s	remaining: 17.6s
1569:	learn: 0.6687823	total: 19.5s	remaining: 17.6s
1570:	learn: 0.6686230	total: 19.6s	remaining: 17.6s
1571:	learn: 0.6684394	total: 19.6s	remaining: 17.6s
1572:	learn: 0.6681404	total: 19.6s	remaining: 17.5s
1573:	learn: 0.6679785	total: 19.6s	remaining: 17.5s
1574:	learn: 0.6677974	total: 19.6s	remaining: 17.5s
1575:	learn: 0.6673075	total: 19.6s	remaining: 17.5s
1576:	learn: 0.6671289	total: 19.6s	remaining: 17.5s
1577:	learn: 0.6669060	total: 19.6s	remaining: 17.5s
1578:	learn: 0.6666280	total: 19.7s	remaining: 17.5s
1579:	learn: 0.6664172	total: 19.7s	remaining: 17.5s
1580:	learn: 0.6662143	total: 19.7s	remaining: 17.4s
1581:	learn: 0.6660370	total: 19.7s	remaining: 17.4s
1582:	learn: 0.6658609	total: 19.7s	remaining: 17.4s
1583:	learn: 0.6656706	total: 19.7s	remaining: 17.4s
1584:	learn: 0.6654656	total: 19.7s	remaining: 17.4s
1585:	learn: 0.6652424	total: 19.8s	remaining: 17.4s
1586:	learn: 0.6650932	total: 19.8s	remaining: 17.4s
1587:	learn: 0.6648555	total: 19.8s	remaining: 17.4s
1588:	learn: 0.6646107	total: 19.8s	remaining: 17.3s
1589:	learn: 0.6643971	total: 19.8s	remaining: 17.3s
1590:	learn: 0.6641882	total: 19.8s	remaining: 17.3s
1591:	learn: 0.6639296	total: 19.8s	remaining: 17.3s
1592:	learn: 0.6637885	total: 19.8s	remaining: 17.3s
1593:	learn: 0.6636242	total: 19.8s	remaining: 17.3s
1594:	learn: 0.6633975	total: 19.9s	remaining: 17.3s
1595:	learn: 0.6631098	total: 19.9s	remaining: 17.3s
1596:	learn: 0.6628183	total: 19.9s	remaining: 17.2s
1597:	learn: 0.6625973	total: 19.9s	remaining: 17.2s
1598:	learn: 0.6624023	total: 19.9s	remaining: 17.2s
1599:	learn: 0.6621764	total: 19.9s	remaining: 17.2s
1600:	learn: 0.6619164	total: 19.9s	remaining: 17.2s
1601:	learn: 0.6617496	total: 19.9s	remaining: 17.2s
1602:	learn: 0.6616197	total: 20s	remaining: 17.2s
1603:	learn: 0.6613901	total: 20s	remaining: 17.2s
1604:	learn: 0.6612407	total: 20s	remaining: 17.1s
1605:	learn: 0.6610273	total: 20s	remaining: 17.1s
1606:	learn: 0.6607222	total: 20s	remaining: 17.1s
1607:	learn: 0.6605154	total: 20s	remaining: 17.1s
1608:	learn: 0.6603212	total: 20s	remaining: 17.1s
1609:	learn: 0.6600802	total: 20s	remaining: 17.1s
1610:	learn: 0.6599070	total: 20.1s	remaining: 17.1s
1611:	learn: 0.6597475	total: 20.1s	remaining: 17.1s
1612:	learn: 0.6595509	total: 20.1s	remaining: 17s
1613:	learn: 0.6594128	total: 20.1s	remaining: 17s
1614:	learn: 0.6591837	total: 20.1s	remaining: 17s
1615:	learn: 0.6589222	total: 20.1s	remaining: 17s
1616:	learn: 0.6588123	total: 20.1s	remaining: 17s
1617:	learn: 0.6586406	total: 20.1s	remaining: 17s
1618:	learn: 0.6583887	total: 20.2s	remaining: 17s
1619:	learn: 0.6581885	total: 20.2s	remaining: 17s

1620:	learn: 0.6579830	total: 20.2s	remaining: 16.9s
1621:	learn: 0.6577724	total: 20.2s	remaining: 16.9s
1622:	learn: 0.6575608	total: 20.2s	remaining: 16.9s
1623:	learn: 0.6573595	total: 20.2s	remaining: 16.9s
1624:	learn: 0.6571495	total: 20.2s	remaining: 16.9s
1625:	learn: 0.6569342	total: 20.2s	remaining: 16.9s
1626:	learn: 0.6567371	total: 20.2s	remaining: 16.9s
1627:	learn: 0.6564995	total: 20.3s	remaining: 16.9s
1628:	learn: 0.6563145	total: 20.3s	remaining: 16.8s
1629:	learn: 0.6561192	total: 20.3s	remaining: 16.8s
1630:	learn: 0.6559882	total: 20.3s	remaining: 16.8s
1631:	learn: 0.6558355	total: 20.3s	remaining: 16.8s
1632:	learn: 0.6555737	total: 20.3s	remaining: 16.8s
1633:	learn: 0.6551811	total: 20.3s	remaining: 16.8s
1634:	learn: 0.6550613	total: 20.3s	remaining: 16.8s
1635:	learn: 0.6548647	total: 20.4s	remaining: 16.8s
1636:	learn: 0.6546743	total: 20.4s	remaining: 16.7s
1637:	learn: 0.6544318	total: 20.4s	remaining: 16.7s
1638:	learn: 0.6542331	total: 20.4s	remaining: 16.7s
1639:	learn: 0.6540360	total: 20.4s	remaining: 16.7s
1640:	learn: 0.6537637	total: 20.4s	remaining: 16.7s
1641:	learn: 0.6536386	total: 20.4s	remaining: 16.7s
1642:	learn: 0.6534956	total: 20.4s	remaining: 16.7s
1643:	learn: 0.6531840	total: 20.5s	remaining: 16.6s
1644:	learn: 0.6529334	total: 20.5s	remaining: 16.6s
1645:	learn: 0.6526234	total: 20.5s	remaining: 16.6s
1646:	learn: 0.6523345	total: 20.5s	remaining: 16.6s
1647:	learn: 0.6520842	total: 20.5s	remaining: 16.6s
1648:	learn: 0.6518470	total: 20.5s	remaining: 16.6s
1649:	learn: 0.6515546	total: 20.5s	remaining: 16.6s
1650:	learn: 0.6513465	total: 20.5s	remaining: 16.6s
1651:	learn: 0.6511452	total: 20.5s	remaining: 16.5s
1652:	learn: 0.6508921	total: 20.6s	remaining: 16.5s
1653:	learn: 0.6506604	total: 20.6s	remaining: 16.5s
1654:	learn: 0.6504429	total: 20.6s	remaining: 16.5s
1655:	learn: 0.6502514	total: 20.6s	remaining: 16.5s
1656:	learn: 0.6501134	total: 20.6s	remaining: 16.5s
1657:	learn: 0.6498858	total: 20.6s	remaining: 16.5s
1658:	learn: 0.6497181	total: 20.6s	remaining: 16.5s
1659:	learn: 0.6495609	total: 20.6s	remaining: 16.4s
1660:	learn: 0.6493571	total: 20.7s	remaining: 16.4s
1661:	learn: 0.6490923	total: 20.7s	remaining: 16.4s
1662:	learn: 0.6488532	total: 20.7s	remaining: 16.4s
1663:	learn: 0.6485904	total: 20.7s	remaining: 16.4s
1664:	learn: 0.6483286	total: 20.7s	remaining: 16.4s
1665:	learn: 0.6481713	total: 20.7s	remaining: 16.4s
1666:	learn: 0.6479693	total: 20.7s	remaining: 16.4s
1667:	learn: 0.6477502	total: 20.7s	remaining: 16.3s
1668:	learn: 0.6475398	total: 20.8s	remaining: 16.3s
1669:	learn: 0.6473681	total: 20.8s	remaining: 16.3s
1670:	learn: 0.6471200	total: 20.8s	remaining: 16.3s
1671:	learn: 0.6468267	total: 20.8s	remaining: 16.3s
1672:	learn: 0.6466423	total: 20.8s	remaining: 16.3s
1673:	learn: 0.6464160	total: 20.8s	remaining: 16.3s

1674:	learn: 0.6462262	total: 20.8s	remaining: 16.2s
1675:	learn: 0.6460598	total: 20.8s	remaining: 16.2s
1676:	learn: 0.6458690	total: 20.9s	remaining: 16.2s
1677:	learn: 0.6456770	total: 20.9s	remaining: 16.2s
1678:	learn: 0.6454254	total: 20.9s	remaining: 16.2s
1679:	learn: 0.6452279	total: 20.9s	remaining: 16.2s
1680:	learn: 0.6451020	total: 20.9s	remaining: 16.2s
1681:	learn: 0.6449308	total: 20.9s	remaining: 16.2s
1682:	learn: 0.6447555	total: 20.9s	remaining: 16.2s
1683:	learn: 0.6445188	total: 20.9s	remaining: 16.1s
1684:	learn: 0.6442695	total: 21s	remaining: 16.1s
1685:	learn: 0.6440679	total: 21s	remaining: 16.1s
1686:	learn: 0.6438679	total: 21s	remaining: 16.1s
1687:	learn: 0.6436289	total: 21s	remaining: 16.1s
1688:	learn: 0.6434100	total: 21s	remaining: 16.1s
1689:	learn: 0.6431492	total: 21s	remaining: 16.1s
1690:	learn: 0.6429196	total: 21s	remaining: 16.1s
1691:	learn: 0.6427146	total: 21s	remaining: 16s
1692:	learn: 0.6426012	total: 21.1s	remaining: 16s
1693:	learn: 0.6424183	total: 21.1s	remaining: 16s
1694:	learn: 0.6422166	total: 21.1s	remaining: 16s
1695:	learn: 0.6420762	total: 21.1s	remaining: 16s
1696:	learn: 0.6417994	total: 21.1s	remaining: 16s
1697:	learn: 0.6416658	total: 21.1s	remaining: 16s
1698:	learn: 0.6414695	total: 21.1s	remaining: 16s
1699:	learn: 0.6412917	total: 21.1s	remaining: 15.9s
1700:	learn: 0.6411405	total: 21.2s	remaining: 15.9s
1701:	learn: 0.6408084	total: 21.2s	remaining: 15.9s
1702:	learn: 0.6405166	total: 21.2s	remaining: 15.9s
1703:	learn: 0.6403734	total: 21.2s	remaining: 15.9s
1704:	learn: 0.6401726	total: 21.2s	remaining: 15.9s
1705:	learn: 0.6399513	total: 21.2s	remaining: 15.9s
1706:	learn: 0.6396537	total: 21.2s	remaining: 15.9s
1707:	learn: 0.6393977	total: 21.2s	remaining: 15.8s
1708:	learn: 0.6392640	total: 21.3s	remaining: 15.8s
1709:	learn: 0.6390520	total: 21.3s	remaining: 15.8s
1710:	learn: 0.6389046	total: 21.3s	remaining: 15.8s
1711:	learn: 0.6386963	total: 21.3s	remaining: 15.8s
1712:	learn: 0.6385534	total: 21.3s	remaining: 15.8s
1713:	learn: 0.6383869	total: 21.3s	remaining: 15.8s
1714:	learn: 0.6382119	total: 21.3s	remaining: 15.8s
1715:	learn: 0.6380020	total: 21.3s	remaining: 15.7s
1716:	learn: 0.6378624	total: 21.4s	remaining: 15.7s
1717:	learn: 0.6376703	total: 21.4s	remaining: 15.7s
1718:	learn: 0.6373478	total: 21.4s	remaining: 15.7s
1719:	learn: 0.6371160	total: 21.4s	remaining: 15.7s
1720:	learn: 0.6369281	total: 21.4s	remaining: 15.7s
1721:	learn: 0.6367382	total: 21.4s	remaining: 15.7s
1722:	learn: 0.6365332	total: 21.4s	remaining: 15.7s
1723:	learn: 0.6363165	total: 21.4s	remaining: 15.6s
1724:	learn: 0.6361707	total: 21.5s	remaining: 15.6s
1725:	learn: 0.6359654	total: 21.5s	remaining: 15.6s
1726:	learn: 0.6358139	total: 21.5s	remaining: 15.6s
1727:	learn: 0.6355534	total: 21.5s	remaining: 15.6s

1728:	learn: 0.6351828	total: 21.5s	remaining: 15.6s
1729:	learn: 0.6349853	total: 21.5s	remaining: 15.6s
1730:	learn: 0.6347756	total: 21.5s	remaining: 15.6s
1731:	learn: 0.6346473	total: 21.5s	remaining: 15.5s
1732:	learn: 0.6343854	total: 21.6s	remaining: 15.5s
1733:	learn: 0.6342031	total: 21.6s	remaining: 15.5s
1734:	learn: 0.6340414	total: 21.6s	remaining: 15.5s
1735:	learn: 0.6338589	total: 21.6s	remaining: 15.5s
1736:	learn: 0.6337000	total: 21.6s	remaining: 15.5s
1737:	learn: 0.6334134	total: 21.6s	remaining: 15.5s
1738:	learn: 0.6331948	total: 21.6s	remaining: 15.5s
1739:	learn: 0.6330531	total: 21.6s	remaining: 15.5s
1740:	learn: 0.6328242	total: 21.7s	remaining: 15.4s
1741:	learn: 0.6326583	total: 21.7s	remaining: 15.4s
1742:	learn: 0.6324381	total: 21.7s	remaining: 15.4s
1743:	learn: 0.6323055	total: 21.7s	remaining: 15.4s
1744:	learn: 0.6321084	total: 21.7s	remaining: 15.4s
1745:	learn: 0.6319968	total: 21.7s	remaining: 15.4s
1746:	learn: 0.6318025	total: 21.7s	remaining: 15.4s
1747:	learn: 0.6316294	total: 21.7s	remaining: 15.3s
1748:	learn: 0.6314570	total: 21.8s	remaining: 15.3s
1749:	learn: 0.6312658	total: 21.8s	remaining: 15.3s
1750:	learn: 0.6310955	total: 21.8s	remaining: 15.3s
1751:	learn: 0.6308652	total: 21.8s	remaining: 15.3s
1752:	learn: 0.6305675	total: 21.8s	remaining: 15.3s
1753:	learn: 0.6303956	total: 21.8s	remaining: 15.3s
1754:	learn: 0.6301817	total: 21.8s	remaining: 15.3s
1755:	learn: 0.6299690	total: 21.8s	remaining: 15.2s
1756:	learn: 0.6297506	total: 21.9s	remaining: 15.2s
1757:	learn: 0.6294722	total: 21.9s	remaining: 15.2s
1758:	learn: 0.6292576	total: 21.9s	remaining: 15.2s
1759:	learn: 0.6290883	total: 21.9s	remaining: 15.2s
1760:	learn: 0.6289749	total: 21.9s	remaining: 15.2s
1761:	learn: 0.6286576	total: 21.9s	remaining: 15.2s
1762:	learn: 0.6284975	total: 21.9s	remaining: 15.2s
1763:	learn: 0.6282863	total: 21.9s	remaining: 15.1s
1764:	learn: 0.6279982	total: 22s	remaining: 15.1s
1765:	learn: 0.6278508	total: 22s	remaining: 15.1s
1766:	learn: 0.6277231	total: 22s	remaining: 15.1s
1767:	learn: 0.6275149	total: 22s	remaining: 15.1s
1768:	learn: 0.6272463	total: 22s	remaining: 15.1s
1769:	learn: 0.6271233	total: 22s	remaining: 15.1s
1770:	learn: 0.6269712	total: 22s	remaining: 15.1s
1771:	learn: 0.6267841	total: 22s	remaining: 15.1s
1772:	learn: 0.6265974	total: 22.1s	remaining: 15s
1773:	learn: 0.6263350	total: 22.1s	remaining: 15s
1774:	learn: 0.6259394	total: 22.1s	remaining: 15s
1775:	learn: 0.6257714	total: 22.1s	remaining: 15s
1776:	learn: 0.6255865	total: 22.1s	remaining: 15s
1777:	learn: 0.6253784	total: 22.1s	remaining: 15s
1778:	learn: 0.6252115	total: 22.1s	remaining: 15s
1779:	learn: 0.6250664	total: 22.1s	remaining: 15s
1780:	learn: 0.6248948	total: 22.2s	remaining: 14.9s
1781:	learn: 0.6247385	total: 22.2s	remaining: 14.9s

1782:	learn: 0.6245375	total: 22.2s	remaining: 14.9s
1783:	learn: 0.6243747	total: 22.2s	remaining: 14.9s
1784:	learn: 0.6241260	total: 22.2s	remaining: 14.9s
1785:	learn: 0.6239158	total: 22.2s	remaining: 14.9s
1786:	learn: 0.6236836	total: 22.2s	remaining: 14.9s
1787:	learn: 0.6234763	total: 22.2s	remaining: 14.9s
1788:	learn: 0.6232640	total: 22.3s	remaining: 14.8s
1789:	learn: 0.6230837	total: 22.3s	remaining: 14.8s
1790:	learn: 0.6228450	total: 22.3s	remaining: 14.8s
1791:	learn: 0.6226588	total: 22.3s	remaining: 14.8s
1792:	learn: 0.6224412	total: 22.3s	remaining: 14.8s
1793:	learn: 0.6222198	total: 22.3s	remaining: 14.8s
1794:	learn: 0.6220239	total: 22.3s	remaining: 14.8s
1795:	learn: 0.6218235	total: 22.3s	remaining: 14.8s
1796:	learn: 0.6217131	total: 22.4s	remaining: 14.7s
1797:	learn: 0.6215576	total: 22.4s	remaining: 14.7s
1798:	learn: 0.6213386	total: 22.4s	remaining: 14.7s
1799:	learn: 0.6211032	total: 22.4s	remaining: 14.7s
1800:	learn: 0.6209491	total: 22.4s	remaining: 14.7s
1801:	learn: 0.6206924	total: 22.4s	remaining: 14.7s
1802:	learn: 0.6204835	total: 22.4s	remaining: 14.7s
1803:	learn: 0.6203171	total: 22.4s	remaining: 14.7s
1804:	learn: 0.6201786	total: 22.4s	remaining: 14.6s
1805:	learn: 0.6200569	total: 22.5s	remaining: 14.6s
1806:	learn: 0.6198647	total: 22.5s	remaining: 14.6s
1807:	learn: 0.6196729	total: 22.5s	remaining: 14.6s
1808:	learn: 0.6193086	total: 22.5s	remaining: 14.6s
1809:	learn: 0.6191033	total: 22.5s	remaining: 14.6s
1810:	learn: 0.6188763	total: 22.5s	remaining: 14.6s
1811:	learn: 0.6187078	total: 22.5s	remaining: 14.5s
1812:	learn: 0.6185474	total: 22.5s	remaining: 14.5s
1813:	learn: 0.6183929	total: 22.6s	remaining: 14.5s
1814:	learn: 0.6182337	total: 22.6s	remaining: 14.5s
1815:	learn: 0.6180741	total: 22.6s	remaining: 14.5s
1816:	learn: 0.6179741	total: 22.6s	remaining: 14.5s
1817:	learn: 0.6176645	total: 22.6s	remaining: 14.5s
1818:	learn: 0.6175197	total: 22.6s	remaining: 14.5s
1819:	learn: 0.6173570	total: 22.6s	remaining: 14.4s
1820:	learn: 0.6171938	total: 22.6s	remaining: 14.4s
1821:	learn: 0.6170370	total: 22.6s	remaining: 14.4s
1822:	learn: 0.6168462	total: 22.7s	remaining: 14.4s
1823:	learn: 0.6167238	total: 22.7s	remaining: 14.4s
1824:	learn: 0.6165787	total: 22.7s	remaining: 14.4s
1825:	learn: 0.6163688	total: 22.7s	remaining: 14.4s
1826:	learn: 0.6161958	total: 22.7s	remaining: 14.4s
1827:	learn: 0.6159739	total: 22.7s	remaining: 14.3s
1828:	learn: 0.6157806	total: 22.7s	remaining: 14.3s
1829:	learn: 0.6156165	total: 22.7s	remaining: 14.3s
1830:	learn: 0.6154457	total: 22.8s	remaining: 14.3s
1831:	learn: 0.6153394	total: 22.8s	remaining: 14.3s
1832:	learn: 0.6151876	total: 22.8s	remaining: 14.3s
1833:	learn: 0.6149952	total: 22.8s	remaining: 14.3s
1834:	learn: 0.6148677	total: 22.8s	remaining: 14.3s
1835:	learn: 0.6146900	total: 22.8s	remaining: 14.2s

1836:	learn: 0.6145165	total: 22.8s	remaining: 14.2s
1837:	learn: 0.6144138	total: 22.8s	remaining: 14.2s
1838:	learn: 0.6141175	total: 22.9s	remaining: 14.2s
1839:	learn: 0.6139682	total: 22.9s	remaining: 14.2s
1840:	learn: 0.6137557	total: 22.9s	remaining: 14.2s
1841:	learn: 0.6135831	total: 22.9s	remaining: 14.2s
1842:	learn: 0.6133988	total: 22.9s	remaining: 14.2s
1843:	learn: 0.6132563	total: 22.9s	remaining: 14.1s
1844:	learn: 0.6130765	total: 22.9s	remaining: 14.1s
1845:	learn: 0.6128966	total: 22.9s	remaining: 14.1s
1846:	learn: 0.6126890	total: 23s	remaining: 14.1s
1847:	learn: 0.6125983	total: 23s	remaining: 14.1s
1848:	learn: 0.6124698	total: 23s	remaining: 14.1s
1849:	learn: 0.6122433	total: 23s	remaining: 14.1s
1850:	learn: 0.6120719	total: 23s	remaining: 14.1s
1851:	learn: 0.6118709	total: 23s	remaining: 14s
1852:	learn: 0.6117065	total: 23s	remaining: 14s
1853:	learn: 0.6114937	total: 23s	remaining: 14s
1854:	learn: 0.6113599	total: 23.1s	remaining: 14s
1855:	learn: 0.6110880	total: 23.1s	remaining: 14s
1856:	learn: 0.6107581	total: 23.1s	remaining: 14s
1857:	learn: 0.6106165	total: 23.1s	remaining: 14s
1858:	learn: 0.6104328	total: 23.1s	remaining: 14s
1859:	learn: 0.6102157	total: 23.1s	remaining: 13.9s
1860:	learn: 0.6100347	total: 23.1s	remaining: 13.9s
1861:	learn: 0.6099337	total: 23.1s	remaining: 13.9s
1862:	learn: 0.6098060	total: 23.2s	remaining: 13.9s
1863:	learn: 0.6096032	total: 23.2s	remaining: 13.9s
1864:	learn: 0.6094012	total: 23.2s	remaining: 13.9s
1865:	learn: 0.6092479	total: 23.2s	remaining: 13.9s
1866:	learn: 0.6089872	total: 23.2s	remaining: 13.9s
1867:	learn: 0.6088372	total: 23.2s	remaining: 13.8s
1868:	learn: 0.6086942	total: 23.2s	remaining: 13.8s
1869:	learn: 0.6085638	total: 23.2s	remaining: 13.8s
1870:	learn: 0.6083564	total: 23.3s	remaining: 13.8s
1871:	learn: 0.6081993	total: 23.3s	remaining: 13.8s
1872:	learn: 0.6080836	total: 23.3s	remaining: 13.8s
1873:	learn: 0.6079038	total: 23.3s	remaining: 13.8s
1874:	learn: 0.6077487	total: 23.3s	remaining: 13.8s
1875:	learn: 0.6075992	total: 23.3s	remaining: 13.7s
1876:	learn: 0.6074466	total: 23.3s	remaining: 13.7s
1877:	learn: 0.6071967	total: 23.3s	remaining: 13.7s
1878:	learn: 0.6069898	total: 23.4s	remaining: 13.7s
1879:	learn: 0.6068151	total: 23.4s	remaining: 13.7s
1880:	learn: 0.6066148	total: 23.4s	remaining: 13.7s
1881:	learn: 0.6064628	total: 23.4s	remaining: 13.7s
1882:	learn: 0.6062510	total: 23.4s	remaining: 13.7s
1883:	learn: 0.6060356	total: 23.4s	remaining: 13.6s
1884:	learn: 0.6058718	total: 23.4s	remaining: 13.6s
1885:	learn: 0.6057416	total: 23.4s	remaining: 13.6s
1886:	learn: 0.6055734	total: 23.4s	remaining: 13.6s
1887:	learn: 0.6054434	total: 23.5s	remaining: 13.6s
1888:	learn: 0.6052837	total: 23.5s	remaining: 13.6s
1889:	learn: 0.6051156	total: 23.5s	remaining: 13.6s

1890:	learn: 0.6049872	total: 23.5s	remaining: 13.6s
1891:	learn: 0.6048240	total: 23.5s	remaining: 13.5s
1892:	learn: 0.6046525	total: 23.5s	remaining: 13.5s
1893:	learn: 0.6044530	total: 23.5s	remaining: 13.5s
1894:	learn: 0.6042615	total: 23.5s	remaining: 13.5s
1895:	learn: 0.6040815	total: 23.6s	remaining: 13.5s
1896:	learn: 0.6039152	total: 23.6s	remaining: 13.5s
1897:	learn: 0.6038177	total: 23.6s	remaining: 13.5s
1898:	learn: 0.6036474	total: 23.6s	remaining: 13.5s
1899:	learn: 0.6034959	total: 23.6s	remaining: 13.4s
1900:	learn: 0.6033520	total: 23.6s	remaining: 13.4s
1901:	learn: 0.6032235	total: 23.6s	remaining: 13.4s
1902:	learn: 0.6030513	total: 23.6s	remaining: 13.4s
1903:	learn: 0.6028704	total: 23.7s	remaining: 13.4s
1904:	learn: 0.6026804	total: 23.7s	remaining: 13.4s
1905:	learn: 0.6024970	total: 23.7s	remaining: 13.4s
1906:	learn: 0.6023546	total: 23.7s	remaining: 13.4s
1907:	learn: 0.6022197	total: 23.7s	remaining: 13.3s
1908:	learn: 0.6020419	total: 23.7s	remaining: 13.3s
1909:	learn: 0.6018663	total: 23.7s	remaining: 13.3s
1910:	learn: 0.6017469	total: 23.8s	remaining: 13.3s
1911:	learn: 0.6015445	total: 23.8s	remaining: 13.3s
1912:	learn: 0.6012968	total: 23.8s	remaining: 13.3s
1913:	learn: 0.6011260	total: 23.8s	remaining: 13.3s
1914:	learn: 0.6009976	total: 23.8s	remaining: 13.3s
1915:	learn: 0.6008184	total: 23.8s	remaining: 13.2s
1916:	learn: 0.6006719	total: 23.8s	remaining: 13.2s
1917:	learn: 0.6004909	total: 23.8s	remaining: 13.2s
1918:	learn: 0.6003214	total: 23.9s	remaining: 13.2s
1919:	learn: 0.6001566	total: 23.9s	remaining: 13.2s
1920:	learn: 0.5999030	total: 23.9s	remaining: 13.2s
1921:	learn: 0.5997340	total: 23.9s	remaining: 13.2s
1922:	learn: 0.5995157	total: 23.9s	remaining: 13.2s
1923:	learn: 0.5993143	total: 23.9s	remaining: 13.2s
1924:	learn: 0.5991363	total: 23.9s	remaining: 13.1s
1925:	learn: 0.5989796	total: 23.9s	remaining: 13.1s
1926:	learn: 0.5988556	total: 24s	remaining: 13.1s
1927:	learn: 0.5986444	total: 24s	remaining: 13.1s
1928:	learn: 0.5984676	total: 24s	remaining: 13.1s
1929:	learn: 0.5983280	total: 24s	remaining: 13.1s
1930:	learn: 0.5981364	total: 24s	remaining: 13.1s
1931:	learn: 0.5979693	total: 24s	remaining: 13.1s
1932:	learn: 0.5978162	total: 24s	remaining: 13s
1933:	learn: 0.5976446	total: 24.1s	remaining: 13s
1934:	learn: 0.5974523	total: 24.1s	remaining: 13s
1935:	learn: 0.5972721	total: 24.1s	remaining: 13s
1936:	learn: 0.5971172	total: 24.1s	remaining: 13s
1937:	learn: 0.5969469	total: 24.1s	remaining: 13s
1938:	learn: 0.5966900	total: 24.1s	remaining: 13s
1939:	learn: 0.5965563	total: 24.1s	remaining: 13s
1940:	learn: 0.5964223	total: 24.2s	remaining: 13s
1941:	learn: 0.5962621	total: 24.2s	remaining: 12.9s
1942:	learn: 0.5961086	total: 24.2s	remaining: 12.9s
1943:	learn: 0.5959861	total: 24.2s	remaining: 12.9s

1944:	learn: 0.5958099	total: 24.2s	remaining: 12.9s
1945:	learn: 0.5956713	total: 24.2s	remaining: 12.9s
1946:	learn: 0.5954835	total: 24.2s	remaining: 12.9s
1947:	learn: 0.5952859	total: 24.2s	remaining: 12.9s
1948:	learn: 0.5951007	total: 24.3s	remaining: 12.9s
1949:	learn: 0.5949316	total: 24.3s	remaining: 12.8s
1950:	learn: 0.5947474	total: 24.3s	remaining: 12.8s
1951:	learn: 0.5945458	total: 24.3s	remaining: 12.8s
1952:	learn: 0.5943871	total: 24.3s	remaining: 12.8s
1953:	learn: 0.5942543	total: 24.3s	remaining: 12.8s
1954:	learn: 0.5941316	total: 24.3s	remaining: 12.8s
1955:	learn: 0.5939787	total: 24.3s	remaining: 12.8s
1956:	learn: 0.5937617	total: 24.4s	remaining: 12.8s
1957:	learn: 0.5935588	total: 24.4s	remaining: 12.7s
1958:	learn: 0.5933521	total: 24.4s	remaining: 12.7s
1959:	learn: 0.5931730	total: 24.4s	remaining: 12.7s
1960:	learn: 0.5930145	total: 24.4s	remaining: 12.7s
1961:	learn: 0.5927623	total: 24.4s	remaining: 12.7s
1962:	learn: 0.5925859	total: 24.4s	remaining: 12.7s
1963:	learn: 0.5924115	total: 24.4s	remaining: 12.7s
1964:	learn: 0.5923204	total: 24.4s	remaining: 12.7s
1965:	learn: 0.5921953	total: 24.5s	remaining: 12.6s
1966:	learn: 0.5920964	total: 24.5s	remaining: 12.6s
1967:	learn: 0.5919057	total: 24.5s	remaining: 12.6s
1968:	learn: 0.5917239	total: 24.5s	remaining: 12.6s
1969:	learn: 0.5916078	total: 24.5s	remaining: 12.6s
1970:	learn: 0.5914808	total: 24.5s	remaining: 12.6s
1971:	learn: 0.5913398	total: 24.5s	remaining: 12.6s
1972:	learn: 0.5912325	total: 24.6s	remaining: 12.6s
1973:	learn: 0.5911023	total: 24.6s	remaining: 12.5s
1974:	learn: 0.5909665	total: 24.6s	remaining: 12.5s
1975:	learn: 0.5907846	total: 24.6s	remaining: 12.5s
1976:	learn: 0.5906842	total: 24.6s	remaining: 12.5s
1977:	learn: 0.5905662	total: 24.6s	remaining: 12.5s
1978:	learn: 0.5904562	total: 24.6s	remaining: 12.5s
1979:	learn: 0.5903780	total: 24.6s	remaining: 12.5s
1980:	learn: 0.5902211	total: 24.7s	remaining: 12.5s
1981:	learn: 0.5900595	total: 24.7s	remaining: 12.4s
1982:	learn: 0.5899443	total: 24.7s	remaining: 12.4s
1983:	learn: 0.5897938	total: 24.7s	remaining: 12.4s
1984:	learn: 0.5896137	total: 24.7s	remaining: 12.4s
1985:	learn: 0.5894902	total: 24.7s	remaining: 12.4s
1986:	learn: 0.5893323	total: 24.7s	remaining: 12.4s
1987:	learn: 0.5891988	total: 24.7s	remaining: 12.4s
1988:	learn: 0.5890403	total: 24.8s	remaining: 12.4s
1989:	learn: 0.5889244	total: 24.8s	remaining: 12.3s
1990:	learn: 0.5887765	total: 24.8s	remaining: 12.3s
1991:	learn: 0.5886447	total: 24.8s	remaining: 12.3s
1992:	learn: 0.5884696	total: 24.8s	remaining: 12.3s
1993:	learn: 0.5883293	total: 24.8s	remaining: 12.3s
1994:	learn: 0.5882391	total: 24.8s	remaining: 12.3s
1995:	learn: 0.5880891	total: 24.8s	remaining: 12.3s
1996:	learn: 0.5878728	total: 24.9s	remaining: 12.3s
1997:	learn: 0.5877458	total: 24.9s	remaining: 12.2s

1998:	learn: 0.5875619	total: 24.9s	remaining: 12.2s
1999:	learn: 0.5873931	total: 24.9s	remaining: 12.2s
2000:	learn: 0.5872742	total: 24.9s	remaining: 12.2s
2001:	learn: 0.5871229	total: 24.9s	remaining: 12.2s
2002:	learn: 0.5869620	total: 24.9s	remaining: 12.2s
2003:	learn: 0.5868077	total: 24.9s	remaining: 12.2s
2004:	learn: 0.5866524	total: 24.9s	remaining: 12.2s
2005:	learn: 0.5865058	total: 25s	remaining: 12.1s
2006:	learn: 0.5863708	total: 25s	remaining: 12.1s
2007:	learn: 0.5861263	total: 25s	remaining: 12.1s
2008:	learn: 0.5859551	total: 25s	remaining: 12.1s
2009:	learn: 0.5858503	total: 25s	remaining: 12.1s
2010:	learn: 0.5856551	total: 25s	remaining: 12.1s
2011:	learn: 0.5855228	total: 25s	remaining: 12.1s
2012:	learn: 0.5853612	total: 25s	remaining: 12.1s
2013:	learn: 0.5852116	total: 25.1s	remaining: 12s
2014:	learn: 0.5850678	total: 25.1s	remaining: 12s
2015:	learn: 0.5849191	total: 25.1s	remaining: 12s
2016:	learn: 0.5847742	total: 25.1s	remaining: 12s
2017:	learn: 0.5846130	total: 25.1s	remaining: 12s
2018:	learn: 0.5844640	total: 25.1s	remaining: 12s
2019:	learn: 0.5843502	total: 25.1s	remaining: 12s
2020:	learn: 0.5842454	total: 25.1s	remaining: 12s
2021:	learn: 0.5841104	total: 25.2s	remaining: 11.9s
2022:	learn: 0.5838847	total: 25.2s	remaining: 11.9s
2023:	learn: 0.5837166	total: 25.2s	remaining: 11.9s
2024:	learn: 0.5835601	total: 25.2s	remaining: 11.9s
2025:	learn: 0.5834006	total: 25.2s	remaining: 11.9s
2026:	learn: 0.5831914	total: 25.2s	remaining: 11.9s
2027:	learn: 0.5829669	total: 25.2s	remaining: 11.9s
2028:	learn: 0.5828528	total: 25.2s	remaining: 11.9s
2029:	learn: 0.5827290	total: 25.3s	remaining: 11.8s
2030:	learn: 0.5825531	total: 25.3s	remaining: 11.8s
2031:	learn: 0.5823942	total: 25.3s	remaining: 11.8s
2032:	learn: 0.5822950	total: 25.3s	remaining: 11.8s
2033:	learn: 0.5821502	total: 25.3s	remaining: 11.8s
2034:	learn: 0.5819635	total: 25.3s	remaining: 11.8s
2035:	learn: 0.5817627	total: 25.3s	remaining: 11.8s
2036:	learn: 0.5815325	total: 25.3s	remaining: 11.8s
2037:	learn: 0.5814455	total: 25.4s	remaining: 11.7s
2038:	learn: 0.5812252	total: 25.4s	remaining: 11.7s
2039:	learn: 0.5810421	total: 25.4s	remaining: 11.7s
2040:	learn: 0.5809073	total: 25.4s	remaining: 11.7s
2041:	learn: 0.5807688	total: 25.4s	remaining: 11.7s
2042:	learn: 0.5806111	total: 25.4s	remaining: 11.7s
2043:	learn: 0.5804731	total: 25.4s	remaining: 11.7s
2044:	learn: 0.5803222	total: 25.4s	remaining: 11.7s
2045:	learn: 0.5801136	total: 25.5s	remaining: 11.6s
2046:	learn: 0.5799841	total: 25.5s	remaining: 11.6s
2047:	learn: 0.5798655	total: 25.5s	remaining: 11.6s
2048:	learn: 0.5797665	total: 25.5s	remaining: 11.6s
2049:	learn: 0.5796235	total: 25.5s	remaining: 11.6s
2050:	learn: 0.5795168	total: 25.5s	remaining: 11.6s
2051:	learn: 0.5793805	total: 25.5s	remaining: 11.6s

2052:	learn: 0.5792502	total: 25.5s	remaining: 11.6s
2053:	learn: 0.5791165	total: 25.6s	remaining: 11.5s
2054:	learn: 0.5789850	total: 25.6s	remaining: 11.5s
2055:	learn: 0.5788241	total: 25.6s	remaining: 11.5s
2056:	learn: 0.5786822	total: 25.6s	remaining: 11.5s
2057:	learn: 0.5785407	total: 25.6s	remaining: 11.5s
2058:	learn: 0.5783852	total: 25.6s	remaining: 11.5s
2059:	learn: 0.5782735	total: 25.6s	remaining: 11.5s
2060:	learn: 0.5781500	total: 25.6s	remaining: 11.5s
2061:	learn: 0.5779389	total: 25.7s	remaining: 11.4s
2062:	learn: 0.5777496	total: 25.7s	remaining: 11.4s
2063:	learn: 0.5776002	total: 25.7s	remaining: 11.4s
2064:	learn: 0.5773802	total: 25.7s	remaining: 11.4s
2065:	learn: 0.5772256	total: 25.7s	remaining: 11.4s
2066:	learn: 0.5770742	total: 25.7s	remaining: 11.4s
2067:	learn: 0.5769214	total: 25.7s	remaining: 11.4s
2068:	learn: 0.5767815	total: 25.8s	remaining: 11.4s
2069:	learn: 0.5766384	total: 25.8s	remaining: 11.4s
2070:	learn: 0.5764793	total: 25.8s	remaining: 11.3s
2071:	learn: 0.5763311	total: 25.8s	remaining: 11.3s
2072:	learn: 0.5762049	total: 25.8s	remaining: 11.3s
2073:	learn: 0.5760181	total: 25.8s	remaining: 11.3s
2074:	learn: 0.5759034	total: 25.8s	remaining: 11.3s
2075:	learn: 0.5758032	total: 25.9s	remaining: 11.3s
2076:	learn: 0.5756723	total: 25.9s	remaining: 11.3s
2077:	learn: 0.5755325	total: 25.9s	remaining: 11.3s
2078:	learn: 0.5753923	total: 25.9s	remaining: 11.2s
2079:	learn: 0.5752102	total: 25.9s	remaining: 11.2s
2080:	learn: 0.5750817	total: 25.9s	remaining: 11.2s
2081:	learn: 0.5749394	total: 25.9s	remaining: 11.2s
2082:	learn: 0.5747145	total: 25.9s	remaining: 11.2s
2083:	learn: 0.5745819	total: 26s	remaining: 11.2s
2084:	learn: 0.5744377	total: 26s	remaining: 11.2s
2085:	learn: 0.5743230	total: 26s	remaining: 11.2s
2086:	learn: 0.5741899	total: 26s	remaining: 11.1s
2087:	learn: 0.5739953	total: 26s	remaining: 11.1s
2088:	learn: 0.5739141	total: 26s	remaining: 11.1s
2089:	learn: 0.5737169	total: 26s	remaining: 11.1s
2090:	learn: 0.5735594	total: 26s	remaining: 11.1s
2091:	learn: 0.5734884	total: 26.1s	remaining: 11.1s
2092:	learn: 0.5733226	total: 26.1s	remaining: 11.1s
2093:	learn: 0.5731336	total: 26.1s	remaining: 11.1s
2094:	learn: 0.5730104	total: 26.1s	remaining: 11s
2095:	learn: 0.5728877	total: 26.1s	remaining: 11s
2096:	learn: 0.5727592	total: 26.1s	remaining: 11s
2097:	learn: 0.5727068	total: 26.1s	remaining: 11s
2098:	learn: 0.5726004	total: 26.1s	remaining: 11s
2099:	learn: 0.5724529	total: 26.2s	remaining: 11s
2100:	learn: 0.5723311	total: 26.2s	remaining: 11s
2101:	learn: 0.5721944	total: 26.2s	remaining: 11s
2102:	learn: 0.5720662	total: 26.2s	remaining: 10.9s
2103:	learn: 0.5719479	total: 26.2s	remaining: 10.9s
2104:	learn: 0.5718103	total: 26.2s	remaining: 10.9s
2105:	learn: 0.5716875	total: 26.2s	remaining: 10.9s

2106:	learn: 0.5715536	total: 26.2s	remaining: 10.9s
2107:	learn: 0.5714089	total: 26.3s	remaining: 10.9s
2108:	learn: 0.5712851	total: 26.3s	remaining: 10.9s
2109:	learn: 0.5711388	total: 26.3s	remaining: 10.9s
2110:	learn: 0.5710078	total: 26.3s	remaining: 10.9s
2111:	learn: 0.5708916	total: 26.3s	remaining: 10.8s
2112:	learn: 0.5707859	total: 26.3s	remaining: 10.8s
2113:	learn: 0.5706511	total: 26.3s	remaining: 10.8s
2114:	learn: 0.5705225	total: 26.4s	remaining: 10.8s
2115:	learn: 0.5704101	total: 26.4s	remaining: 10.8s
2116:	learn: 0.5702540	total: 26.4s	remaining: 10.8s
2117:	learn: 0.5700865	total: 26.4s	remaining: 10.8s
2118:	learn: 0.5699409	total: 26.4s	remaining: 10.8s
2119:	learn: 0.5697988	total: 26.4s	remaining: 10.7s
2120:	learn: 0.5696552	total: 26.4s	remaining: 10.7s
2121:	learn: 0.5695241	total: 26.4s	remaining: 10.7s
2122:	learn: 0.5694805	total: 26.5s	remaining: 10.7s
2123:	learn: 0.5692788	total: 26.5s	remaining: 10.7s
2124:	learn: 0.5691641	total: 26.5s	remaining: 10.7s
2125:	learn: 0.5690028	total: 26.5s	remaining: 10.7s
2126:	learn: 0.5688397	total: 26.5s	remaining: 10.7s
2127:	learn: 0.5687546	total: 26.5s	remaining: 10.6s
2128:	learn: 0.5686227	total: 26.5s	remaining: 10.6s
2129:	learn: 0.5685309	total: 26.5s	remaining: 10.6s
2130:	learn: 0.5683550	total: 26.6s	remaining: 10.6s
2131:	learn: 0.5682456	total: 26.6s	remaining: 10.6s
2132:	learn: 0.5681273	total: 26.6s	remaining: 10.6s
2133:	learn: 0.5679623	total: 26.6s	remaining: 10.6s
2134:	learn: 0.5678547	total: 26.6s	remaining: 10.6s
2135:	learn: 0.5677668	total: 26.6s	remaining: 10.5s
2136:	learn: 0.5675786	total: 26.6s	remaining: 10.5s
2137:	learn: 0.5674526	total: 26.6s	remaining: 10.5s
2138:	learn: 0.5672498	total: 26.7s	remaining: 10.5s
2139:	learn: 0.5671318	total: 26.7s	remaining: 10.5s
2140:	learn: 0.5668796	total: 26.7s	remaining: 10.5s
2141:	learn: 0.5667452	total: 26.7s	remaining: 10.5s
2142:	learn: 0.5665969	total: 26.7s	remaining: 10.5s
2143:	learn: 0.5664824	total: 26.7s	remaining: 10.4s
2144:	learn: 0.5663552	total: 26.7s	remaining: 10.4s
2145:	learn: 0.5662205	total: 26.7s	remaining: 10.4s
2146:	learn: 0.5661132	total: 26.8s	remaining: 10.4s
2147:	learn: 0.5659380	total: 26.8s	remaining: 10.4s
2148:	learn: 0.5657157	total: 26.8s	remaining: 10.4s
2149:	learn: 0.5655756	total: 26.8s	remaining: 10.4s
2150:	learn: 0.5654278	total: 26.8s	remaining: 10.4s
2151:	learn: 0.5653505	total: 26.8s	remaining: 10.3s
2152:	learn: 0.5651500	total: 26.8s	remaining: 10.3s
2153:	learn: 0.5650415	total: 26.8s	remaining: 10.3s
2154:	learn: 0.5648613	total: 26.8s	remaining: 10.3s
2155:	learn: 0.5647559	total: 26.9s	remaining: 10.3s
2156:	learn: 0.5646411	total: 26.9s	remaining: 10.3s
2157:	learn: 0.5645568	total: 26.9s	remaining: 10.3s
2158:	learn: 0.5644519	total: 26.9s	remaining: 10.3s
2159:	learn: 0.5643167	total: 26.9s	remaining: 10.2s

2160:	learn: 0.5642330	total: 26.9s	remaining: 10.2s
2161:	learn: 0.5641074	total: 26.9s	remaining: 10.2s
2162:	learn: 0.5640109	total: 27s	remaining: 10.2s
2163:	learn: 0.5638659	total: 27s	remaining: 10.2s
2164:	learn: 0.5636893	total: 27s	remaining: 10.2s
2165:	learn: 0.5635785	total: 27s	remaining: 10.2s
2166:	learn: 0.5633839	total: 27s	remaining: 10.2s
2167:	learn: 0.5632553	total: 27s	remaining: 10.1s
2168:	learn: 0.5631741	total: 27s	remaining: 10.1s
2169:	learn: 0.5630824	total: 27s	remaining: 10.1s
2170:	learn: 0.5629493	total: 27.1s	remaining: 10.1s
2171:	learn: 0.5627884	total: 27.1s	remaining: 10.1s
2172:	learn: 0.5626249	total: 27.1s	remaining: 10.1s
2173:	learn: 0.5625210	total: 27.1s	remaining: 10.1s
2174:	learn: 0.5624083	total: 27.1s	remaining: 10.1s
2175:	learn: 0.5622796	total: 27.1s	remaining: 10s
2176:	learn: 0.5621055	total: 27.1s	remaining: 10s
2177:	learn: 0.5620242	total: 27.1s	remaining: 10s
2178:	learn: 0.5619291	total: 27.2s	remaining: 10s
2179:	learn: 0.5617851	total: 27.2s	remaining: 9.99s
2180:	learn: 0.5616339	total: 27.2s	remaining: 9.98s
2181:	learn: 0.5615297	total: 27.2s	remaining: 9.97s
2182:	learn: 0.5613429	total: 27.2s	remaining: 9.96s
2183:	learn: 0.5612049	total: 27.2s	remaining: 9.94s
2184:	learn: 0.5610547	total: 27.2s	remaining: 9.93s
2185:	learn: 0.5609054	total: 27.2s	remaining: 9.92s
2186:	learn: 0.5607573	total: 27.3s	remaining: 9.91s
2187:	learn: 0.5606549	total: 27.3s	remaining: 9.89s
2188:	learn: 0.5605394	total: 27.3s	remaining: 9.88s
2189:	learn: 0.5604447	total: 27.3s	remaining: 9.87s
2190:	learn: 0.5602840	total: 27.3s	remaining: 9.86s
2191:	learn: 0.5601467	total: 27.3s	remaining: 9.84s
2192:	learn: 0.5600060	total: 27.3s	remaining: 9.83s
2193:	learn: 0.5598650	total: 27.3s	remaining: 9.82s
2194:	learn: 0.5596201	total: 27.4s	remaining: 9.81s
2195:	learn: 0.5595021	total: 27.4s	remaining: 9.79s
2196:	learn: 0.5593208	total: 27.4s	remaining: 9.78s
2197:	learn: 0.5591740	total: 27.4s	remaining: 9.77s
2198:	learn: 0.5590351	total: 27.4s	remaining: 9.76s
2199:	learn: 0.5589253	total: 27.4s	remaining: 9.74s
2200:	learn: 0.5588150	total: 27.4s	remaining: 9.73s
2201:	learn: 0.5586857	total: 27.4s	remaining: 9.72s
2202:	learn: 0.5586148	total: 27.5s	remaining: 9.71s
2203:	learn: 0.5585044	total: 27.5s	remaining: 9.7s
2204:	learn: 0.5583222	total: 27.5s	remaining: 9.68s
2205:	learn: 0.5581923	total: 27.5s	remaining: 9.67s
2206:	learn: 0.5580590	total: 27.5s	remaining: 9.66s
2207:	learn: 0.5579648	total: 27.5s	remaining: 9.65s
2208:	learn: 0.5578301	total: 27.5s	remaining: 9.63s
2209:	learn: 0.5577289	total: 27.5s	remaining: 9.62s
2210:	learn: 0.5575828	total: 27.6s	remaining: 9.61s
2211:	learn: 0.5575235	total: 27.6s	remaining: 9.6s
2212:	learn: 0.5573830	total: 27.6s	remaining: 9.58s
2213:	learn: 0.5572098	total: 27.6s	remaining: 9.57s

2214:	learn: 0.5570895	total: 27.6s	remaining: 9.56s
2215:	learn: 0.5569931	total: 27.6s	remaining: 9.54s
2216:	learn: 0.5568117	total: 27.6s	remaining: 9.53s
2217:	learn: 0.5566501	total: 27.6s	remaining: 9.52s
2218:	learn: 0.5564975	total: 27.7s	remaining: 9.51s
2219:	learn: 0.5563362	total: 27.7s	remaining: 9.49s
2220:	learn: 0.5561947	total: 27.7s	remaining: 9.48s
2221:	learn: 0.5560640	total: 27.7s	remaining: 9.47s
2222:	learn: 0.5559094	total: 27.7s	remaining: 9.46s
2223:	learn: 0.5557592	total: 27.7s	remaining: 9.45s
2224:	learn: 0.5555931	total: 27.7s	remaining: 9.43s
2225:	learn: 0.5554913	total: 27.7s	remaining: 9.42s
2226:	learn: 0.5553655	total: 27.8s	remaining: 9.41s
2227:	learn: 0.5552456	total: 27.8s	remaining: 9.4s
2228:	learn: 0.5551130	total: 27.8s	remaining: 9.39s
2229:	learn: 0.5549698	total: 27.8s	remaining: 9.37s
2230:	learn: 0.5548171	total: 27.8s	remaining: 9.36s
2231:	learn: 0.5547151	total: 27.8s	remaining: 9.35s
2232:	learn: 0.5546141	total: 27.8s	remaining: 9.34s
2233:	learn: 0.5544879	total: 27.8s	remaining: 9.32s
2234:	learn: 0.5543977	total: 27.9s	remaining: 9.31s
2235:	learn: 0.5542324	total: 27.9s	remaining: 9.3s
2236:	learn: 0.5540892	total: 27.9s	remaining: 9.29s
2237:	learn: 0.5538777	total: 27.9s	remaining: 9.27s
2238:	learn: 0.5537470	total: 27.9s	remaining: 9.26s
2239:	learn: 0.5536451	total: 27.9s	remaining: 9.25s
2240:	learn: 0.5535606	total: 27.9s	remaining: 9.24s
2241:	learn: 0.5534455	total: 27.9s	remaining: 9.22s
2242:	learn: 0.5533047	total: 28s	remaining: 9.21s
2243:	learn: 0.5531619	total: 28s	remaining: 9.2s
2244:	learn: 0.5530535	total: 28s	remaining: 9.19s
2245:	learn: 0.5529603	total: 28s	remaining: 9.18s
2246:	learn: 0.5528456	total: 28s	remaining: 9.16s
2247:	learn: 0.5526258	total: 28s	remaining: 9.15s
2248:	learn: 0.5525198	total: 28s	remaining: 9.14s
2249:	learn: 0.5522751	total: 28s	remaining: 9.12s
2250:	learn: 0.5521879	total: 28.1s	remaining: 9.11s
2251:	learn: 0.5520807	total: 28.1s	remaining: 9.1s
2252:	learn: 0.5519587	total: 28.1s	remaining: 9.09s
2253:	learn: 0.5518493	total: 28.1s	remaining: 9.07s
2254:	learn: 0.5516862	total: 28.1s	remaining: 9.06s
2255:	learn: 0.5515857	total: 28.1s	remaining: 9.05s
2256:	learn: 0.5514919	total: 28.1s	remaining: 9.04s
2257:	learn: 0.5513662	total: 28.1s	remaining: 9.02s
2258:	learn: 0.5511604	total: 28.1s	remaining: 9.01s
2259:	learn: 0.5510865	total: 28.2s	remaining: 9s
2260:	learn: 0.5509799	total: 28.2s	remaining: 8.98s
2261:	learn: 0.5508511	total: 28.2s	remaining: 8.97s
2262:	learn: 0.5507294	total: 28.2s	remaining: 8.96s
2263:	learn: 0.5506086	total: 28.2s	remaining: 8.95s
2264:	learn: 0.5504828	total: 28.2s	remaining: 8.94s
2265:	learn: 0.5503027	total: 28.2s	remaining: 8.92s
2266:	learn: 0.5502156	total: 28.3s	remaining: 8.91s
2267:	learn: 0.5501117	total: 28.3s	remaining: 8.9s

2268:	learn: 0.5500071	total: 28.3s	remaining: 8.88s
2269:	learn: 0.5498937	total: 28.3s	remaining: 8.87s
2270:	learn: 0.5497520	total: 28.3s	remaining: 8.86s
2271:	learn: 0.5496627	total: 28.3s	remaining: 8.85s
2272:	learn: 0.5495361	total: 28.3s	remaining: 8.84s
2273:	learn: 0.5494535	total: 28.3s	remaining: 8.82s
2274:	learn: 0.5492977	total: 28.3s	remaining: 8.81s
2275:	learn: 0.5491682	total: 28.4s	remaining: 8.8s
2276:	learn: 0.5490585	total: 28.4s	remaining: 8.79s
2277:	learn: 0.5489527	total: 28.4s	remaining: 8.77s
2278:	learn: 0.5488436	total: 28.4s	remaining: 8.76s
2279:	learn: 0.5486809	total: 28.4s	remaining: 8.75s
2280:	learn: 0.5485760	total: 28.4s	remaining: 8.74s
2281:	learn: 0.5484703	total: 28.4s	remaining: 8.72s
2282:	learn: 0.5483321	total: 28.5s	remaining: 8.71s
2283:	learn: 0.5481977	total: 28.5s	remaining: 8.7s
2284:	learn: 0.5480819	total: 28.5s	remaining: 8.69s
2285:	learn: 0.5479632	total: 28.5s	remaining: 8.68s
2286:	learn: 0.5478496	total: 28.5s	remaining: 8.66s
2287:	learn: 0.5477265	total: 28.5s	remaining: 8.65s
2288:	learn: 0.5475866	total: 28.5s	remaining: 8.64s
2289:	learn: 0.5474586	total: 28.5s	remaining: 8.63s
2290:	learn: 0.5473074	total: 28.6s	remaining: 8.61s
2291:	learn: 0.5472033	total: 28.6s	remaining: 8.6s
2292:	learn: 0.5470517	total: 28.6s	remaining: 8.59s
2293:	learn: 0.5468792	total: 28.6s	remaining: 8.57s
2294:	learn: 0.5467435	total: 28.6s	remaining: 8.56s
2295:	learn: 0.5466176	total: 28.6s	remaining: 8.55s
2296:	learn: 0.5465415	total: 28.6s	remaining: 8.54s
2297:	learn: 0.5464486	total: 28.6s	remaining: 8.53s
2298:	learn: 0.5463432	total: 28.7s	remaining: 8.51s
2299:	learn: 0.5462294	total: 28.7s	remaining: 8.5s
2300:	learn: 0.5460842	total: 28.7s	remaining: 8.49s
2301:	learn: 0.5459526	total: 28.7s	remaining: 8.48s
2302:	learn: 0.5458005	total: 28.7s	remaining: 8.46s
2303:	learn: 0.5456556	total: 28.7s	remaining: 8.45s
2304:	learn: 0.5455679	total: 28.7s	remaining: 8.44s
2305:	learn: 0.5454762	total: 28.7s	remaining: 8.43s
2306:	learn: 0.5452712	total: 28.8s	remaining: 8.41s
2307:	learn: 0.5451581	total: 28.8s	remaining: 8.4s
2308:	learn: 0.5449745	total: 28.8s	remaining: 8.39s
2309:	learn: 0.5448623	total: 28.8s	remaining: 8.37s
2310:	learn: 0.5447393	total: 28.8s	remaining: 8.36s
2311:	learn: 0.5445800	total: 28.8s	remaining: 8.35s
2312:	learn: 0.5444473	total: 28.8s	remaining: 8.34s
2313:	learn: 0.5442858	total: 28.8s	remaining: 8.32s
2314:	learn: 0.5441592	total: 28.9s	remaining: 8.31s
2315:	learn: 0.5440665	total: 28.9s	remaining: 8.3s
2316:	learn: 0.5439397	total: 28.9s	remaining: 8.29s
2317:	learn: 0.5437995	total: 28.9s	remaining: 8.28s
2318:	learn: 0.5436872	total: 28.9s	remaining: 8.26s
2319:	learn: 0.5435465	total: 28.9s	remaining: 8.25s
2320:	learn: 0.5434423	total: 28.9s	remaining: 8.24s
2321:	learn: 0.5433313	total: 28.9s	remaining: 8.23s

2322:	learn: 0.5432444	total: 29s	remaining: 8.21s
2323:	learn: 0.5431123	total: 29s	remaining: 8.2s
2324:	learn: 0.5430464	total: 29s	remaining: 8.19s
2325:	learn: 0.5429577	total: 29s	remaining: 8.18s
2326:	learn: 0.5428233	total: 29s	remaining: 8.16s
2327:	learn: 0.5426743	total: 29s	remaining: 8.15s
2328:	learn: 0.5425566	total: 29s	remaining: 8.14s
2329:	learn: 0.5423987	total: 29s	remaining: 8.13s
2330:	learn: 0.5422736	total: 29.1s	remaining: 8.12s
2331:	learn: 0.5421896	total: 29.1s	remaining: 8.1s
2332:	learn: 0.5420571	total: 29.1s	remaining: 8.09s
2333:	learn: 0.5418967	total: 29.1s	remaining: 8.08s
2334:	learn: 0.5417552	total: 29.1s	remaining: 8.07s
2335:	learn: 0.5416220	total: 29.1s	remaining: 8.05s
2336:	learn: 0.5415162	total: 29.1s	remaining: 8.04s
2337:	learn: 0.5413794	total: 29.2s	remaining: 8.03s
2338:	learn: 0.5412338	total: 29.2s	remaining: 8.02s
2339:	learn: 0.5410633	total: 29.2s	remaining: 8s
2340:	learn: 0.5409753	total: 29.2s	remaining: 7.99s
2341:	learn: 0.5408668	total: 29.2s	remaining: 7.98s
2342:	learn: 0.5407111	total: 29.2s	remaining: 7.97s
2343:	learn: 0.5405967	total: 29.2s	remaining: 7.96s
2344:	learn: 0.5405306	total: 29.2s	remaining: 7.94s
2345:	learn: 0.5403957	total: 29.3s	remaining: 7.93s
2346:	learn: 0.5402734	total: 29.3s	remaining: 7.92s
2347:	learn: 0.5401750	total: 29.3s	remaining: 7.91s
2348:	learn: 0.5400969	total: 29.3s	remaining: 7.89s
2349:	learn: 0.5399916	total: 29.3s	remaining: 7.88s
2350:	learn: 0.5398402	total: 29.3s	remaining: 7.87s
2351:	learn: 0.5397692	total: 29.3s	remaining: 7.85s
2352:	learn: 0.5396465	total: 29.3s	remaining: 7.84s
2353:	learn: 0.5395520	total: 29.4s	remaining: 7.83s
2354:	learn: 0.5394591	total: 29.4s	remaining: 7.82s
2355:	learn: 0.5393239	total: 29.4s	remaining: 7.8s
2356:	learn: 0.5392353	total: 29.4s	remaining: 7.79s
2357:	learn: 0.5391297	total: 29.4s	remaining: 7.78s
2358:	learn: 0.5389393	total: 29.4s	remaining: 7.77s
2359:	learn: 0.5387967	total: 29.4s	remaining: 7.76s
2360:	learn: 0.5386326	total: 29.4s	remaining: 7.74s
2361:	learn: 0.5385183	total: 29.5s	remaining: 7.73s
2362:	learn: 0.5383597	total: 29.5s	remaining: 7.72s
2363:	learn: 0.5382612	total: 29.5s	remaining: 7.71s
2364:	learn: 0.5381282	total: 29.5s	remaining: 7.69s
2365:	learn: 0.5380122	total: 29.5s	remaining: 7.68s
2366:	learn: 0.5379240	total: 29.5s	remaining: 7.67s
2367:	learn: 0.5377765	total: 29.5s	remaining: 7.66s
2368:	learn: 0.5376680	total: 29.5s	remaining: 7.64s
2369:	learn: 0.5375357	total: 29.6s	remaining: 7.63s
2370:	learn: 0.5374464	total: 29.6s	remaining: 7.62s
2371:	learn: 0.5373172	total: 29.6s	remaining: 7.61s
2372:	learn: 0.5372226	total: 29.6s	remaining: 7.59s
2373:	learn: 0.5371443	total: 29.6s	remaining: 7.58s
2374:	learn: 0.5370380	total: 29.6s	remaining: 7.57s
2375:	learn: 0.5369530	total: 29.6s	remaining: 7.56s

2376:	learn: 0.5368310	total: 29.6s	remaining: 7.54s
2377:	learn: 0.5367393	total: 29.7s	remaining: 7.53s
2378:	learn: 0.5366069	total: 29.7s	remaining: 7.52s
2379:	learn: 0.5364998	total: 29.7s	remaining: 7.51s
2380:	learn: 0.5363384	total: 29.7s	remaining: 7.5s
2381:	learn: 0.5362220	total: 29.7s	remaining: 7.48s
2382:	learn: 0.5360702	total: 29.7s	remaining: 7.47s
2383:	learn: 0.5359813	total: 29.7s	remaining: 7.46s
2384:	learn: 0.5358769	total: 29.7s	remaining: 7.45s
2385:	learn: 0.5357747	total: 29.8s	remaining: 7.43s
2386:	learn: 0.5357092	total: 29.8s	remaining: 7.42s
2387:	learn: 0.5356209	total: 29.8s	remaining: 7.41s
2388:	learn: 0.5355136	total: 29.8s	remaining: 7.39s
2389:	learn: 0.5353857	total: 29.8s	remaining: 7.38s
2390:	learn: 0.5352629	total: 29.8s	remaining: 7.37s
2391:	learn: 0.5351593	total: 29.8s	remaining: 7.36s
2392:	learn: 0.5350293	total: 29.8s	remaining: 7.34s
2393:	learn: 0.5348894	total: 29.9s	remaining: 7.33s
2394:	learn: 0.5347832	total: 29.9s	remaining: 7.32s
2395:	learn: 0.5346336	total: 29.9s	remaining: 7.31s
2396:	learn: 0.5345625	total: 29.9s	remaining: 7.29s
2397:	learn: 0.5344381	total: 29.9s	remaining: 7.28s
2398:	learn: 0.5342799	total: 29.9s	remaining: 7.27s
2399:	learn: 0.5341997	total: 29.9s	remaining: 7.26s
2400:	learn: 0.5340577	total: 29.9s	remaining: 7.24s
2401:	learn: 0.5339052	total: 30s	remaining: 7.23s
2402:	learn: 0.5337572	total: 30s	remaining: 7.22s
2403:	learn: 0.5336214	total: 30s	remaining: 7.21s
2404:	learn: 0.5334739	total: 30s	remaining: 7.2s
2405:	learn: 0.5333346	total: 30s	remaining: 7.18s
2406:	learn: 0.5332286	total: 30s	remaining: 7.17s
2407:	learn: 0.5330757	total: 30s	remaining: 7.16s
2408:	learn: 0.5329447	total: 30s	remaining: 7.14s
2409:	learn: 0.5327955	total: 30s	remaining: 7.13s
2410:	learn: 0.5327013	total: 30.1s	remaining: 7.12s
2411:	learn: 0.5325960	total: 30.1s	remaining: 7.11s
2412:	learn: 0.5324936	total: 30.1s	remaining: 7.09s
2413:	learn: 0.5324158	total: 30.1s	remaining: 7.08s
2414:	learn: 0.5323108	total: 30.1s	remaining: 7.07s
2415:	learn: 0.5321911	total: 30.1s	remaining: 7.06s
2416:	learn: 0.5321062	total: 30.1s	remaining: 7.04s
2417:	learn: 0.5319809	total: 30.1s	remaining: 7.03s
2418:	learn: 0.5318750	total: 30.2s	remaining: 7.02s
2419:	learn: 0.5317360	total: 30.2s	remaining: 7.01s
2420:	learn: 0.5316028	total: 30.2s	remaining: 6.99s
2421:	learn: 0.5315111	total: 30.2s	remaining: 6.98s
2422:	learn: 0.5314226	total: 30.2s	remaining: 6.97s
2423:	learn: 0.5313548	total: 30.2s	remaining: 6.96s
2424:	learn: 0.5312881	total: 30.3s	remaining: 6.95s
2425:	learn: 0.5311984	total: 30.3s	remaining: 6.94s
2426:	learn: 0.5311155	total: 30.3s	remaining: 6.92s
2427:	learn: 0.5309769	total: 30.3s	remaining: 6.91s
2428:	learn: 0.5308502	total: 30.3s	remaining: 6.9s
2429:	learn: 0.5307296	total: 30.3s	remaining: 6.89s

2430:	learn: 0.5305801	total: 30.3s	remaining: 6.88s
2431:	learn: 0.5304554	total: 30.3s	remaining: 6.86s
2432:	learn: 0.5303237	total: 30.4s	remaining: 6.85s
2433:	learn: 0.5302248	total: 30.4s	remaining: 6.84s
2434:	learn: 0.5301347	total: 30.4s	remaining: 6.83s
2435:	learn: 0.5300665	total: 30.4s	remaining: 6.81s
2436:	learn: 0.5299697	total: 30.4s	remaining: 6.8s
2437:	learn: 0.5298639	total: 30.4s	remaining: 6.79s
2438:	learn: 0.5297637	total: 30.4s	remaining: 6.78s
2439:	learn: 0.5296839	total: 30.5s	remaining: 6.76s
2440:	learn: 0.5296273	total: 30.5s	remaining: 6.75s
2441:	learn: 0.5294873	total: 30.5s	remaining: 6.74s
2442:	learn: 0.5293611	total: 30.5s	remaining: 6.73s
2443:	learn: 0.5292899	total: 30.5s	remaining: 6.71s
2444:	learn: 0.5292050	total: 30.5s	remaining: 6.7s
2445:	learn: 0.5290900	total: 30.5s	remaining: 6.69s
2446:	learn: 0.5289847	total: 30.5s	remaining: 6.68s
2447:	learn: 0.5288681	total: 30.6s	remaining: 6.66s
2448:	learn: 0.5287382	total: 30.6s	remaining: 6.65s
2449:	learn: 0.5286114	total: 30.6s	remaining: 6.64s
2450:	learn: 0.5284826	total: 30.6s	remaining: 6.63s
2451:	learn: 0.5283591	total: 30.6s	remaining: 6.61s
2452:	learn: 0.5282670	total: 30.6s	remaining: 6.6s
2453:	learn: 0.5281260	total: 30.6s	remaining: 6.59s
2454:	learn: 0.5280273	total: 30.6s	remaining: 6.58s
2455:	learn: 0.5279584	total: 30.7s	remaining: 6.57s
2456:	learn: 0.5278765	total: 30.7s	remaining: 6.55s
2457:	learn: 0.5277473	total: 30.7s	remaining: 6.54s
2458:	learn: 0.5276553	total: 30.7s	remaining: 6.53s
2459:	learn: 0.5275035	total: 30.7s	remaining: 6.51s
2460:	learn: 0.5273798	total: 30.7s	remaining: 6.5s
2461:	learn: 0.5272921	total: 30.7s	remaining: 6.49s
2462:	learn: 0.5271671	total: 30.7s	remaining: 6.48s
2463:	learn: 0.5270397	total: 30.8s	remaining: 6.47s
2464:	learn: 0.5269213	total: 30.8s	remaining: 6.45s
2465:	learn: 0.5267998	total: 30.8s	remaining: 6.44s
2466:	learn: 0.5266613	total: 30.8s	remaining: 6.43s
2467:	learn: 0.5265568	total: 30.8s	remaining: 6.42s
2468:	learn: 0.5264405	total: 30.8s	remaining: 6.41s
2469:	learn: 0.5263409	total: 30.8s	remaining: 6.39s
2470:	learn: 0.5262127	total: 30.9s	remaining: 6.38s
2471:	learn: 0.5261186	total: 30.9s	remaining: 6.37s
2472:	learn: 0.5260086	total: 30.9s	remaining: 6.36s
2473:	learn: 0.5258968	total: 30.9s	remaining: 6.34s
2474:	learn: 0.5257803	total: 30.9s	remaining: 6.33s
2475:	learn: 0.5256538	total: 30.9s	remaining: 6.32s
2476:	learn: 0.5255453	total: 30.9s	remaining: 6.31s
2477:	learn: 0.5254190	total: 31s	remaining: 6.29s
2478:	learn: 0.5252869	total: 31s	remaining: 6.28s
2479:	learn: 0.5251741	total: 31s	remaining: 6.27s
2480:	learn: 0.5250695	total: 31s	remaining: 6.26s
2481:	learn: 0.5249840	total: 31s	remaining: 6.25s
2482:	learn: 0.5249057	total: 31s	remaining: 6.23s
2483:	learn: 0.5248134	total: 31s	remaining: 6.22s

2484:	learn: 0.5246977	total: 31s	remaining: 6.21s
2485:	learn: 0.5245892	total: 31.1s	remaining: 6.2s
2486:	learn: 0.5244816	total: 31.1s	remaining: 6.18s
2487:	learn: 0.5244006	total: 31.1s	remaining: 6.17s
2488:	learn: 0.5243280	total: 31.1s	remaining: 6.16s
2489:	learn: 0.5242402	total: 31.1s	remaining: 6.15s
2490:	learn: 0.5241214	total: 31.1s	remaining: 6.13s
2491:	learn: 0.5239925	total: 31.1s	remaining: 6.12s
2492:	learn: 0.5238696	total: 31.2s	remaining: 6.11s
2493:	learn: 0.5237897	total: 31.2s	remaining: 6.1s
2494:	learn: 0.5236870	total: 31.2s	remaining: 6.08s
2495:	learn: 0.5235873	total: 31.2s	remaining: 6.07s
2496:	learn: 0.5234878	total: 31.2s	remaining: 6.06s
2497:	learn: 0.5233812	total: 31.2s	remaining: 6.05s
2498:	learn: 0.5232336	total: 31.2s	remaining: 6.04s
2499:	learn: 0.5231266	total: 31.2s	remaining: 6.02s
2500:	learn: 0.5230298	total: 31.3s	remaining: 6.01s
2501:	learn: 0.5229007	total: 31.3s	remaining: 6s
2502:	learn: 0.5227984	total: 31.3s	remaining: 5.99s
2503:	learn: 0.5226896	total: 31.3s	remaining: 5.97s
2504:	learn: 0.5226132	total: 31.3s	remaining: 5.96s
2505:	learn: 0.5224354	total: 31.3s	remaining: 5.95s
2506:	learn: 0.5223415	total: 31.3s	remaining: 5.94s
2507:	learn: 0.5222766	total: 31.3s	remaining: 5.92s
2508:	learn: 0.5221760	total: 31.4s	remaining: 5.91s
2509:	learn: 0.5220922	total: 31.4s	remaining: 5.9s
2510:	learn: 0.5219820	total: 31.4s	remaining: 5.89s
2511:	learn: 0.5219137	total: 31.4s	remaining: 5.87s
2512:	learn: 0.5218293	total: 31.4s	remaining: 5.86s
2513:	learn: 0.5217256	total: 31.4s	remaining: 5.85s
2514:	learn: 0.5216040	total: 31.4s	remaining: 5.84s
2515:	learn: 0.5214395	total: 31.4s	remaining: 5.82s
2516:	learn: 0.5213616	total: 31.5s	remaining: 5.81s
2517:	learn: 0.5212622	total: 31.5s	remaining: 5.8s
2518:	learn: 0.5211463	total: 31.5s	remaining: 5.79s
2519:	learn: 0.5210702	total: 31.5s	remaining: 5.77s
2520:	learn: 0.5209822	total: 31.5s	remaining: 5.76s
2521:	learn: 0.5209096	total: 31.5s	remaining: 5.75s
2522:	learn: 0.5207810	total: 31.5s	remaining: 5.74s
2523:	learn: 0.5206775	total: 31.5s	remaining: 5.72s
2524:	learn: 0.5205447	total: 31.6s	remaining: 5.71s
2525:	learn: 0.5204214	total: 31.6s	remaining: 5.7s
2526:	learn: 0.5203214	total: 31.6s	remaining: 5.69s
2527:	learn: 0.5202317	total: 31.6s	remaining: 5.67s
2528:	learn: 0.5201284	total: 31.6s	remaining: 5.66s
2529:	learn: 0.5200020	total: 31.6s	remaining: 5.65s
2530:	learn: 0.5198559	total: 31.6s	remaining: 5.64s
2531:	learn: 0.5197489	total: 31.6s	remaining: 5.62s
2532:	learn: 0.5196909	total: 31.7s	remaining: 5.61s
2533:	learn: 0.5195613	total: 31.7s	remaining: 5.6s
2534:	learn: 0.5194140	total: 31.7s	remaining: 5.59s
2535:	learn: 0.5193480	total: 31.7s	remaining: 5.57s
2536:	learn: 0.5192035	total: 31.7s	remaining: 5.56s
2537:	learn: 0.5191188	total: 31.7s	remaining: 5.55s

2538:	learn: 0.5190374	total: 31.7s	remaining: 5.54s
2539:	learn: 0.5188820	total: 31.7s	remaining: 5.52s
2540:	learn: 0.5188337	total: 31.7s	remaining: 5.51s
2541:	learn: 0.5187529	total: 31.8s	remaining: 5.5s
2542:	learn: 0.5186067	total: 31.8s	remaining: 5.49s
2543:	learn: 0.5185348	total: 31.8s	remaining: 5.47s
2544:	learn: 0.5184356	total: 31.8s	remaining: 5.46s
2545:	learn: 0.5183482	total: 31.8s	remaining: 5.45s
2546:	learn: 0.5182733	total: 31.8s	remaining: 5.43s
2547:	learn: 0.5181810	total: 31.8s	remaining: 5.42s
2548:	learn: 0.5180431	total: 31.8s	remaining: 5.41s
2549:	learn: 0.5179481	total: 31.9s	remaining: 5.4s
2550:	learn: 0.5178717	total: 31.9s	remaining: 5.38s
2551:	learn: 0.5177235	total: 31.9s	remaining: 5.37s
2552:	learn: 0.5176276	total: 31.9s	remaining: 5.36s
2553:	learn: 0.5175049	total: 31.9s	remaining: 5.35s
2554:	learn: 0.5174180	total: 31.9s	remaining: 5.33s
2555:	learn: 0.5173213	total: 31.9s	remaining: 5.32s
2556:	learn: 0.5172349	total: 31.9s	remaining: 5.31s
2557:	learn: 0.5171116	total: 32s	remaining: 5.3s
2558:	learn: 0.5170015	total: 32s	remaining: 5.28s
2559:	learn: 0.5168657	total: 32s	remaining: 5.27s
2560:	learn: 0.5167833	total: 32s	remaining: 5.26s
2561:	learn: 0.5166957	total: 32s	remaining: 5.25s
2562:	learn: 0.5166013	total: 32s	remaining: 5.24s
2563:	learn: 0.5164563	total: 32s	remaining: 5.22s
2564:	learn: 0.5163747	total: 32s	remaining: 5.21s
2565:	learn: 0.5162618	total: 32.1s	remaining: 5.2s
2566:	learn: 0.5161261	total: 32.1s	remaining: 5.18s
2567:	learn: 0.5160464	total: 32.1s	remaining: 5.17s
2568:	learn: 0.5159194	total: 32.1s	remaining: 5.16s
2569:	learn: 0.5158312	total: 32.1s	remaining: 5.15s
2570:	learn: 0.5157031	total: 32.1s	remaining: 5.13s
2571:	learn: 0.5156188	total: 32.1s	remaining: 5.12s
2572:	learn: 0.5155247	total: 32.1s	remaining: 5.11s
2573:	learn: 0.5154132	total: 32.2s	remaining: 5.1s
2574:	learn: 0.5153372	total: 32.2s	remaining: 5.08s
2575:	learn: 0.5152438	total: 32.2s	remaining: 5.07s
2576:	learn: 0.5151683	total: 32.2s	remaining: 5.06s
2577:	learn: 0.5150495	total: 32.2s	remaining: 5.05s
2578:	learn: 0.5149234	total: 32.2s	remaining: 5.04s
2579:	learn: 0.5148370	total: 32.2s	remaining: 5.02s
2580:	learn: 0.5147295	total: 32.2s	remaining: 5.01s
2581:	learn: 0.5146215	total: 32.3s	remaining: 5s
2582:	learn: 0.5145259	total: 32.3s	remaining: 4.99s
2583:	learn: 0.5144122	total: 32.3s	remaining: 4.97s
2584:	learn: 0.5142906	total: 32.3s	remaining: 4.96s
2585:	learn: 0.5141546	total: 32.3s	remaining: 4.95s
2586:	learn: 0.5141005	total: 32.3s	remaining: 4.93s
2587:	learn: 0.5140097	total: 32.3s	remaining: 4.92s
2588:	learn: 0.5138888	total: 32.4s	remaining: 4.91s
2589:	learn: 0.5137844	total: 32.4s	remaining: 4.9s
2590:	learn: 0.5136901	total: 32.4s	remaining: 4.88s
2591:	learn: 0.5136103	total: 32.4s	remaining: 4.87s

2592:	learn: 0.5134912	total: 32.4s	remaining: 4.86s
2593:	learn: 0.5133637	total: 32.4s	remaining: 4.85s
2594:	learn: 0.5132620	total: 32.4s	remaining: 4.84s
2595:	learn: 0.5131638	total: 32.4s	remaining: 4.82s
2596:	learn: 0.5130672	total: 32.5s	remaining: 4.81s
2597:	learn: 0.5128885	total: 32.5s	remaining: 4.8s
2598:	learn: 0.5127614	total: 32.5s	remaining: 4.79s
2599:	learn: 0.5126749	total: 32.5s	remaining: 4.77s
2600:	learn: 0.5125591	total: 32.5s	remaining: 4.76s
2601:	learn: 0.5124264	total: 32.5s	remaining: 4.75s
2602:	learn: 0.5123233	total: 32.5s	remaining: 4.74s
2603:	learn: 0.5122501	total: 32.5s	remaining: 4.72s
2604:	learn: 0.5121146	total: 32.6s	remaining: 4.71s
2605:	learn: 0.5120280	total: 32.6s	remaining: 4.7s
2606:	learn: 0.5119267	total: 32.6s	remaining: 4.69s
2607:	learn: 0.5118277	total: 32.6s	remaining: 4.67s
2608:	learn: 0.5117100	total: 32.6s	remaining: 4.66s
2609:	learn: 0.5116260	total: 32.6s	remaining: 4.65s
2610:	learn: 0.5115012	total: 32.6s	remaining: 4.64s
2611:	learn: 0.5114214	total: 32.6s	remaining: 4.62s
2612:	learn: 0.5112707	total: 32.7s	remaining: 4.61s
2613:	learn: 0.5111868	total: 32.7s	remaining: 4.6s
2614:	learn: 0.5111058	total: 32.7s	remaining: 4.59s
2615:	learn: 0.5109335	total: 32.7s	remaining: 4.57s
2616:	learn: 0.5108322	total: 32.7s	remaining: 4.56s
2617:	learn: 0.5106990	total: 32.7s	remaining: 4.55s
2618:	learn: 0.5105993	total: 32.7s	remaining: 4.54s
2619:	learn: 0.5104965	total: 32.7s	remaining: 4.52s
2620:	learn: 0.5103864	total: 32.8s	remaining: 4.51s
2621:	learn: 0.5102999	total: 32.8s	remaining: 4.5s
2622:	learn: 0.5102032	total: 32.8s	remaining: 4.49s
2623:	learn: 0.5101005	total: 32.8s	remaining: 4.47s
2624:	learn: 0.5099962	total: 32.8s	remaining: 4.46s
2625:	learn: 0.5099158	total: 32.8s	remaining: 4.45s
2626:	learn: 0.5098535	total: 32.8s	remaining: 4.44s
2627:	learn: 0.5097375	total: 32.9s	remaining: 4.42s
2628:	learn: 0.5096703	total: 32.9s	remaining: 4.41s
2629:	learn: 0.5095749	total: 32.9s	remaining: 4.4s
2630:	learn: 0.5094646	total: 32.9s	remaining: 4.39s
2631:	learn: 0.5093469	total: 32.9s	remaining: 4.38s
2632:	learn: 0.5092575	total: 32.9s	remaining: 4.36s
2633:	learn: 0.5091624	total: 32.9s	remaining: 4.35s
2634:	learn: 0.5090862	total: 32.9s	remaining: 4.34s
2635:	learn: 0.5089671	total: 33s	remaining: 4.33s
2636:	learn: 0.5088731	total: 33s	remaining: 4.31s
2637:	learn: 0.5087896	total: 33s	remaining: 4.3s
2638:	learn: 0.5087071	total: 33s	remaining: 4.29s
2639:	learn: 0.5085706	total: 33s	remaining: 4.27s
2640:	learn: 0.5084883	total: 33s	remaining: 4.26s
2641:	learn: 0.5084107	total: 33s	remaining: 4.25s
2642:	learn: 0.5083269	total: 33s	remaining: 4.24s
2643:	learn: 0.5082495	total: 33s	remaining: 4.22s
2644:	learn: 0.5081341	total: 33.1s	remaining: 4.21s
2645:	learn: 0.5080535	total: 33.1s	remaining: 4.2s

2646:	learn: 0.5079978	total: 33.1s	remaining: 4.19s
2647:	learn: 0.5079358	total: 33.1s	remaining: 4.18s
2648:	learn: 0.5078149	total: 33.1s	remaining: 4.16s
2649:	learn: 0.5077111	total: 33.1s	remaining: 4.15s
2650:	learn: 0.5076211	total: 33.2s	remaining: 4.14s
2651:	learn: 0.5075049	total: 33.2s	remaining: 4.13s
2652:	learn: 0.5074214	total: 33.2s	remaining: 4.11s
2653:	learn: 0.5073110	total: 33.2s	remaining: 4.1s
2654:	learn: 0.5072235	total: 33.2s	remaining: 4.09s
2655:	learn: 0.5071282	total: 33.2s	remaining: 4.08s
2656:	learn: 0.5070471	total: 33.2s	remaining: 4.06s
2657:	learn: 0.5069560	total: 33.2s	remaining: 4.05s
2658:	learn: 0.5068438	total: 33.3s	remaining: 4.04s
2659:	learn: 0.5067555	total: 33.3s	remaining: 4.03s
2660:	learn: 0.5066515	total: 33.3s	remaining: 4.01s
2661:	learn: 0.5065951	total: 33.3s	remaining: 4s
2662:	learn: 0.5064810	total: 33.3s	remaining: 3.99s
2663:	learn: 0.5063761	total: 33.3s	remaining: 3.98s
2664:	learn: 0.5062681	total: 33.3s	remaining: 3.96s
2665:	learn: 0.5061621	total: 33.3s	remaining: 3.95s
2666:	learn: 0.5060694	total: 33.4s	remaining: 3.94s
2667:	learn: 0.5059730	total: 33.4s	remaining: 3.93s
2668:	learn: 0.5058694	total: 33.4s	remaining: 3.91s
2669:	learn: 0.5057783	total: 33.4s	remaining: 3.9s
2670:	learn: 0.5056712	total: 33.4s	remaining: 3.89s
2671:	learn: 0.5055938	total: 33.4s	remaining: 3.88s
2672:	learn: 0.5055257	total: 33.4s	remaining: 3.87s
2673:	learn: 0.5054197	total: 33.4s	remaining: 3.85s
2674:	learn: 0.5053311	total: 33.5s	remaining: 3.84s
2675:	learn: 0.5052495	total: 33.5s	remaining: 3.83s
2676:	learn: 0.5051767	total: 33.5s	remaining: 3.81s
2677:	learn: 0.5051172	total: 33.5s	remaining: 3.8s
2678:	learn: 0.5050091	total: 33.5s	remaining: 3.79s
2679:	learn: 0.5049473	total: 33.5s	remaining: 3.78s
2680:	learn: 0.5048563	total: 33.5s	remaining: 3.77s
2681:	learn: 0.5047429	total: 33.5s	remaining: 3.75s
2682:	learn: 0.5046706	total: 33.6s	remaining: 3.74s
2683:	learn: 0.5046000	total: 33.6s	remaining: 3.73s
2684:	learn: 0.5045298	total: 33.6s	remaining: 3.71s
2685:	learn: 0.5044589	total: 33.6s	remaining: 3.7s
2686:	learn: 0.5043694	total: 33.6s	remaining: 3.69s
2687:	learn: 0.5042973	total: 33.6s	remaining: 3.68s
2688:	learn: 0.5041837	total: 33.6s	remaining: 3.67s
2689:	learn: 0.5041226	total: 33.7s	remaining: 3.65s
2690:	learn: 0.5040432	total: 33.7s	remaining: 3.64s
2691:	learn: 0.5039602	total: 33.7s	remaining: 3.63s
2692:	learn: 0.5038803	total: 33.7s	remaining: 3.62s
2693:	learn: 0.5037970	total: 33.7s	remaining: 3.6s
2694:	learn: 0.5037166	total: 33.7s	remaining: 3.59s
2695:	learn: 0.5036259	total: 33.7s	remaining: 3.58s
2696:	learn: 0.5035556	total: 33.7s	remaining: 3.56s
2697:	learn: 0.5034827	total: 33.8s	remaining: 3.55s
2698:	learn: 0.5034175	total: 33.8s	remaining: 3.54s
2699:	learn: 0.5033406	total: 33.8s	remaining: 3.53s

2700:	learn: 0.5032846	total: 33.8s	remaining: 3.52s
2701:	learn: 0.5031940	total: 33.8s	remaining: 3.5s
2702:	learn: 0.5031386	total: 33.8s	remaining: 3.49s
2703:	learn: 0.5030294	total: 33.8s	remaining: 3.48s
2704:	learn: 0.5029614	total: 33.8s	remaining: 3.47s
2705:	learn: 0.5028871	total: 33.9s	remaining: 3.45s
2706:	learn: 0.5028436	total: 33.9s	remaining: 3.44s
2707:	learn: 0.5027402	total: 33.9s	remaining: 3.43s
2708:	learn: 0.5026480	total: 33.9s	remaining: 3.42s
2709:	learn: 0.5025439	total: 33.9s	remaining: 3.4s
2710:	learn: 0.5024771	total: 33.9s	remaining: 3.39s
2711:	learn: 0.5023723	total: 33.9s	remaining: 3.38s
2712:	learn: 0.5022829	total: 33.9s	remaining: 3.37s
2713:	learn: 0.5021991	total: 34s	remaining: 3.35s
2714:	learn: 0.5021356	total: 34s	remaining: 3.34s
2715:	learn: 0.5020619	total: 34s	remaining: 3.33s
2716:	learn: 0.5020065	total: 34s	remaining: 3.32s
2717:	learn: 0.5019177	total: 34s	remaining: 3.3s
2718:	learn: 0.5018247	total: 34s	remaining: 3.29s
2719:	learn: 0.5017012	total: 34s	remaining: 3.28s
2720:	learn: 0.5016022	total: 34.1s	remaining: 3.27s
2721:	learn: 0.5015083	total: 34.1s	remaining: 3.25s
2722:	learn: 0.5014276	total: 34.1s	remaining: 3.24s
2723:	learn: 0.5013344	total: 34.1s	remaining: 3.23s
2724:	learn: 0.5012865	total: 34.1s	remaining: 3.22s
2725:	learn: 0.5012319	total: 34.1s	remaining: 3.2s
2726:	learn: 0.5011550	total: 34.1s	remaining: 3.19s
2727:	learn: 0.5010442	total: 34.1s	remaining: 3.18s
2728:	learn: 0.5009398	total: 34.2s	remaining: 3.17s
2729:	learn: 0.5008428	total: 34.2s	remaining: 3.15s
2730:	learn: 0.5007787	total: 34.2s	remaining: 3.14s
2731:	learn: 0.5006956	total: 34.2s	remaining: 3.13s
2732:	learn: 0.5006233	total: 34.2s	remaining: 3.12s
2733:	learn: 0.5005240	total: 34.2s	remaining: 3.1s
2734:	learn: 0.5004087	total: 34.2s	remaining: 3.09s
2735:	learn: 0.5003235	total: 34.2s	remaining: 3.08s
2736:	learn: 0.5002454	total: 34.3s	remaining: 3.07s
2737:	learn: 0.5001291	total: 34.3s	remaining: 3.05s
2738:	learn: 0.5000729	total: 34.3s	remaining: 3.04s
2739:	learn: 0.4999726	total: 34.3s	remaining: 3.03s
2740:	learn: 0.4999055	total: 34.3s	remaining: 3.02s
2741:	learn: 0.4998387	total: 34.3s	remaining: 3s
2742:	learn: 0.4997720	total: 34.3s	remaining: 2.99s
2743:	learn: 0.4996833	total: 34.3s	remaining: 2.98s
2744:	learn: 0.4995997	total: 34.4s	remaining: 2.97s
2745:	learn: 0.4994896	total: 34.4s	remaining: 2.95s
2746:	learn: 0.4993827	total: 34.4s	remaining: 2.94s
2747:	learn: 0.4993027	total: 34.4s	remaining: 2.93s
2748:	learn: 0.4992329	total: 34.4s	remaining: 2.92s
2749:	learn: 0.4991372	total: 34.4s	remaining: 2.9s
2750:	learn: 0.4990409	total: 34.4s	remaining: 2.89s
2751:	learn: 0.4989452	total: 34.4s	remaining: 2.88s
2752:	learn: 0.4988378	total: 34.5s	remaining: 2.87s
2753:	learn: 0.4987565	total: 34.5s	remaining: 2.85s

2754:	learn: 0.4986758	total: 34.5s	remaining: 2.84s
2755:	learn: 0.4986115	total: 34.5s	remaining: 2.83s
2756:	learn: 0.4984634	total: 34.5s	remaining: 2.82s
2757:	learn: 0.4983836	total: 34.5s	remaining: 2.8s
2758:	learn: 0.4982885	total: 34.5s	remaining: 2.79s
2759:	learn: 0.4982058	total: 34.6s	remaining: 2.78s
2760:	learn: 0.4981115	total: 34.6s	remaining: 2.77s
2761:	learn: 0.4980393	total: 34.6s	remaining: 2.75s
2762:	learn: 0.4979467	total: 34.6s	remaining: 2.74s
2763:	learn: 0.4978743	total: 34.6s	remaining: 2.73s
2764:	learn: 0.4977796	total: 34.6s	remaining: 2.72s
2765:	learn: 0.4977333	total: 34.6s	remaining: 2.7s
2766:	learn: 0.4976287	total: 34.6s	remaining: 2.69s
2767:	learn: 0.4975318	total: 34.6s	remaining: 2.68s
2768:	learn: 0.4974722	total: 34.7s	remaining: 2.67s
2769:	learn: 0.4974154	total: 34.7s	remaining: 2.65s
2770:	learn: 0.4973477	total: 34.7s	remaining: 2.64s
2771:	learn: 0.4972516	total: 34.7s	remaining: 2.63s
2772:	learn: 0.4971683	total: 34.7s	remaining: 2.62s
2773:	learn: 0.4970910	total: 34.7s	remaining: 2.6s
2774:	learn: 0.4969866	total: 34.7s	remaining: 2.59s
2775:	learn: 0.4968943	total: 34.7s	remaining: 2.58s
2776:	learn: 0.4968118	total: 34.8s	remaining: 2.57s
2777:	learn: 0.4967264	total: 34.8s	remaining: 2.55s
2778:	learn: 0.4966418	total: 34.8s	remaining: 2.54s
2779:	learn: 0.4965676	total: 34.8s	remaining: 2.53s
2780:	learn: 0.4964542	total: 34.8s	remaining: 2.52s
2781:	learn: 0.4963842	total: 34.8s	remaining: 2.5s
2782:	learn: 0.4962769	total: 34.8s	remaining: 2.49s
2783:	learn: 0.4961845	total: 34.9s	remaining: 2.48s
2784:	learn: 0.4960977	total: 34.9s	remaining: 2.47s
2785:	learn: 0.4960187	total: 34.9s	remaining: 2.45s
2786:	learn: 0.4959688	total: 34.9s	remaining: 2.44s
2787:	learn: 0.4958920	total: 34.9s	remaining: 2.43s
2788:	learn: 0.4957910	total: 34.9s	remaining: 2.42s
2789:	learn: 0.4957077	total: 34.9s	remaining: 2.4s
2790:	learn: 0.4956484	total: 34.9s	remaining: 2.39s
2791:	learn: 0.4955515	total: 35s	remaining: 2.38s
2792:	learn: 0.4954019	total: 35s	remaining: 2.37s
2793:	learn: 0.4953133	total: 35s	remaining: 2.35s
2794:	learn: 0.4952459	total: 35s	remaining: 2.34s
2795:	learn: 0.4951679	total: 35s	remaining: 2.33s
2796:	learn: 0.4950674	total: 35s	remaining: 2.32s
2797:	learn: 0.4949728	total: 35s	remaining: 2.3s
2798:	learn: 0.4948826	total: 35s	remaining: 2.29s
2799:	learn: 0.4947902	total: 35.1s	remaining: 2.28s
2800:	learn: 0.4946819	total: 35.1s	remaining: 2.27s
2801:	learn: 0.4946094	total: 35.1s	remaining: 2.25s
2802:	learn: 0.4945358	total: 35.1s	remaining: 2.24s
2803:	learn: 0.4944843	total: 35.1s	remaining: 2.23s
2804:	learn: 0.4943767	total: 35.1s	remaining: 2.22s
2805:	learn: 0.4942594	total: 35.1s	remaining: 2.2s
2806:	learn: 0.4941875	total: 35.2s	remaining: 2.19s
2807:	learn: 0.4940872	total: 35.2s	remaining: 2.18s

2808:	learn: 0.4940017	total: 35.2s	remaining: 2.17s
2809:	learn: 0.4939371	total: 35.2s	remaining: 2.15s
2810:	learn: 0.4938647	total: 35.2s	remaining: 2.14s
2811:	learn: 0.4937921	total: 35.2s	remaining: 2.13s
2812:	learn: 0.4937033	total: 35.2s	remaining: 2.12s
2813:	learn: 0.4936056	total: 35.3s	remaining: 2.1s
2814:	learn: 0.4935398	total: 35.3s	remaining: 2.09s
2815:	learn: 0.4934743	total: 35.3s	remaining: 2.08s
2816:	learn: 0.4933669	total: 35.3s	remaining: 2.07s
2817:	learn: 0.4932727	total: 35.3s	remaining: 2.05s
2818:	learn: 0.4931459	total: 35.3s	remaining: 2.04s
2819:	learn: 0.4930453	total: 35.3s	remaining: 2.03s
2820:	learn: 0.4929625	total: 35.3s	remaining: 2.02s
2821:	learn: 0.4929034	total: 35.4s	remaining: 2s
2822:	learn: 0.4927997	total: 35.4s	remaining: 1.99s
2823:	learn: 0.4926947	total: 35.4s	remaining: 1.98s
2824:	learn: 0.4926182	total: 35.4s	remaining: 1.97s
2825:	learn: 0.4925277	total: 35.4s	remaining: 1.95s
2826:	learn: 0.4924463	total: 35.4s	remaining: 1.94s
2827:	learn: 0.4923420	total: 35.4s	remaining: 1.93s
2828:	learn: 0.4922776	total: 35.4s	remaining: 1.92s
2829:	learn: 0.4921628	total: 35.5s	remaining: 1.9s
2830:	learn: 0.4920756	total: 35.5s	remaining: 1.89s
2831:	learn: 0.4919720	total: 35.5s	remaining: 1.88s
2832:	learn: 0.4918686	total: 35.5s	remaining: 1.87s
2833:	learn: 0.4917453	total: 35.5s	remaining: 1.85s
2834:	learn: 0.4916649	total: 35.5s	remaining: 1.84s
2835:	learn: 0.4915792	total: 35.5s	remaining: 1.83s
2836:	learn: 0.4915013	total: 35.5s	remaining: 1.82s
2837:	learn: 0.4914355	total: 35.6s	remaining: 1.8s
2838:	learn: 0.4913585	total: 35.6s	remaining: 1.79s
2839:	learn: 0.4912275	total: 35.6s	remaining: 1.78s
2840:	learn: 0.4911698	total: 35.6s	remaining: 1.77s
2841:	learn: 0.4910963	total: 35.6s	remaining: 1.75s
2842:	learn: 0.4910015	total: 35.6s	remaining: 1.74s
2843:	learn: 0.4909290	total: 35.6s	remaining: 1.73s
2844:	learn: 0.4908526	total: 35.6s	remaining: 1.72s
2845:	learn: 0.4907493	total: 35.7s	remaining: 1.7s
2846:	learn: 0.4906295	total: 35.7s	remaining: 1.69s
2847:	learn: 0.4905452	total: 35.7s	remaining: 1.68s
2848:	learn: 0.4904898	total: 35.7s	remaining: 1.67s
2849:	learn: 0.4903886	total: 35.7s	remaining: 1.65s
2850:	learn: 0.4903161	total: 35.7s	remaining: 1.64s
2851:	learn: 0.4902839	total: 35.7s	remaining: 1.63s
2852:	learn: 0.4902047	total: 35.8s	remaining: 1.62s
2853:	learn: 0.4901185	total: 35.8s	remaining: 1.6s
2854:	learn: 0.4900276	total: 35.8s	remaining: 1.59s
2855:	learn: 0.4899331	total: 35.8s	remaining: 1.58s
2856:	learn: 0.4898429	total: 35.8s	remaining: 1.57s
2857:	learn: 0.4897621	total: 35.8s	remaining: 1.55s
2858:	learn: 0.4896732	total: 35.8s	remaining: 1.54s
2859:	learn: 0.4895570	total: 35.8s	remaining: 1.53s
2860:	learn: 0.4894891	total: 35.9s	remaining: 1.52s
2861:	learn: 0.4893744	total: 35.9s	remaining: 1.5s

2862:	learn: 0.4892992	total: 35.9s	remaining: 1.49s
2863:	learn: 0.4892216	total: 35.9s	remaining: 1.48s
2864:	learn: 0.4891420	total: 35.9s	remaining: 1.47s
2865:	learn: 0.4890615	total: 35.9s	remaining: 1.45s
2866:	learn: 0.4889848	total: 35.9s	remaining: 1.44s
2867:	learn: 0.4889335	total: 35.9s	remaining: 1.43s
2868:	learn: 0.4888370	total: 36s	remaining: 1.42s
2869:	learn: 0.4887736	total: 36s	remaining: 1.4s
2870:	learn: 0.4886817	total: 36s	remaining: 1.39s
2871:	learn: 0.4886033	total: 36s	remaining: 1.38s
2872:	learn: 0.4884925	total: 36s	remaining: 1.37s
2873:	learn: 0.4884011	total: 36s	remaining: 1.35s
2874:	learn: 0.4883677	total: 36s	remaining: 1.34s
2875:	learn: 0.4883245	total: 36.1s	remaining: 1.33s
2876:	learn: 0.4882380	total: 36.1s	remaining: 1.32s
2877:	learn: 0.4881302	total: 36.1s	remaining: 1.3s
2878:	learn: 0.4880521	total: 36.1s	remaining: 1.29s
2879:	learn: 0.4879740	total: 36.1s	remaining: 1.28s
2880:	learn: 0.4878808	total: 36.1s	remaining: 1.27s
2881:	learn: 0.4877915	total: 36.1s	remaining: 1.25s
2882:	learn: 0.4877116	total: 36.1s	remaining: 1.24s
2883:	learn: 0.4876394	total: 36.2s	remaining: 1.23s
2884:	learn: 0.4875543	total: 36.2s	remaining: 1.22s
2885:	learn: 0.4874911	total: 36.2s	remaining: 1.2s
2886:	learn: 0.4873957	total: 36.2s	remaining: 1.19s
2887:	learn: 0.4873107	total: 36.2s	remaining: 1.18s
2888:	learn: 0.4872344	total: 36.2s	remaining: 1.17s
2889:	learn: 0.4871667	total: 36.2s	remaining: 1.15s
2890:	learn: 0.4870871	total: 36.3s	remaining: 1.14s
2891:	learn: 0.4870098	total: 36.3s	remaining: 1.13s
2892:	learn: 0.4869556	total: 36.3s	remaining: 1.12s
2893:	learn: 0.4868653	total: 36.3s	remaining: 1.1s
2894:	learn: 0.4867807	total: 36.3s	remaining: 1.09s
2895:	learn: 0.4867079	total: 36.3s	remaining: 1.08s
2896:	learn: 0.4866099	total: 36.3s	remaining: 1.07s
2897:	learn: 0.4865315	total: 36.3s	remaining: 1.05s
2898:	learn: 0.4864620	total: 36.4s	remaining: 1.04s
2899:	learn: 0.4863849	total: 36.4s	remaining: 1.03s
2900:	learn: 0.4863110	total: 36.4s	remaining: 1.01s
2901:	learn: 0.4861807	total: 36.4s	remaining: 1s
2902:	learn: 0.4860795	total: 36.4s	remaining: 991ms
2903:	learn: 0.4859931	total: 36.4s	remaining: 978ms
2904:	learn: 0.4859162	total: 36.4s	remaining: 966ms
2905:	learn: 0.4858274	total: 36.5s	remaining: 953ms
2906:	learn: 0.4857797	total: 36.5s	remaining: 941ms
2907:	learn: 0.4856828	total: 36.5s	remaining: 928ms
2908:	learn: 0.4856326	total: 36.5s	remaining: 916ms
2909:	learn: 0.4855318	total: 36.5s	remaining: 903ms
2910:	learn: 0.4854555	total: 36.5s	remaining: 891ms
2911:	learn: 0.4853891	total: 36.5s	remaining: 878ms
2912:	learn: 0.4853188	total: 36.5s	remaining: 866ms
2913:	learn: 0.4852601	total: 36.6s	remaining: 853ms
2914:	learn: 0.4851946	total: 36.6s	remaining: 841ms
2915:	learn: 0.4851203	total: 36.6s	remaining: 828ms

2916:	learn: 0.4850305	total: 36.6s	remaining: 816ms
2917:	learn: 0.4849788	total: 36.6s	remaining: 803ms
2918:	learn: 0.4849169	total: 36.6s	remaining: 791ms
2919:	learn: 0.4848341	total: 36.6s	remaining: 778ms
2920:	learn: 0.4847673	total: 36.7s	remaining: 766ms
2921:	learn: 0.4847039	total: 36.7s	remaining: 753ms
2922:	learn: 0.4846342	total: 36.7s	remaining: 741ms
2923:	learn: 0.4845736	total: 36.7s	remaining: 728ms
2924:	learn: 0.4844452	total: 36.7s	remaining: 715ms
2925:	learn: 0.4843814	total: 36.7s	remaining: 703ms
2926:	learn: 0.4843260	total: 36.7s	remaining: 690ms
2927:	learn: 0.4842625	total: 36.8s	remaining: 678ms
2928:	learn: 0.4841727	total: 36.8s	remaining: 665ms
2929:	learn: 0.4840921	total: 36.8s	remaining: 653ms
2930:	learn: 0.4840375	total: 36.8s	remaining: 640ms
2931:	learn: 0.4839896	total: 36.8s	remaining: 628ms
2932:	learn: 0.4838459	total: 36.8s	remaining: 615ms
2933:	learn: 0.4837228	total: 36.9s	remaining: 603ms
2934:	learn: 0.4836410	total: 36.9s	remaining: 590ms
2935:	learn: 0.4835590	total: 36.9s	remaining: 578ms
2936:	learn: 0.4834718	total: 36.9s	remaining: 565ms
2937:	learn: 0.4833798	total: 36.9s	remaining: 553ms
2938:	learn: 0.4833062	total: 36.9s	remaining: 540ms
2939:	learn: 0.4832260	total: 36.9s	remaining: 528ms
2940:	learn: 0.4831361	total: 37s	remaining: 515ms
2941:	learn: 0.4830356	total: 37s	remaining: 503ms
2942:	learn: 0.4829604	total: 37s	remaining: 490ms
2943:	learn: 0.4829031	total: 37s	remaining: 478ms
2944:	learn: 0.4828064	total: 37s	remaining: 465ms
2945:	learn: 0.4827425	total: 37s	remaining: 452ms
2946:	learn: 0.4826352	total: 37s	remaining: 440ms
2947:	learn: 0.4825600	total: 37.1s	remaining: 427ms
2948:	learn: 0.4824793	total: 37.1s	remaining: 415ms
2949:	learn: 0.4824070	total: 37.1s	remaining: 402ms
2950:	learn: 0.4823171	total: 37.1s	remaining: 390ms
2951:	learn: 0.4821899	total: 37.1s	remaining: 377ms
2952:	learn: 0.4821134	total: 37.1s	remaining: 365ms
2953:	learn: 0.4820103	total: 37.1s	remaining: 352ms
2954:	learn: 0.4819230	total: 37.1s	remaining: 339ms
2955:	learn: 0.4818362	total: 37.2s	remaining: 327ms
2956:	learn: 0.4817405	total: 37.2s	remaining: 314ms
2957:	learn: 0.4816659	total: 37.2s	remaining: 302ms
2958:	learn: 0.4815963	total: 37.2s	remaining: 289ms
2959:	learn: 0.4815086	total: 37.2s	remaining: 277ms
2960:	learn: 0.4814232	total: 37.2s	remaining: 264ms
2961:	learn: 0.4813610	total: 37.2s	remaining: 251ms
2962:	learn: 0.4812639	total: 37.2s	remaining: 239ms
2963:	learn: 0.4811703	total: 37.3s	remaining: 226ms
2964:	learn: 0.4811107	total: 37.3s	remaining: 214ms
2965:	learn: 0.4810464	total: 37.3s	remaining: 201ms
2966:	learn: 0.4809792	total: 37.3s	remaining: 189ms
2967:	learn: 0.4809485	total: 37.3s	remaining: 176ms
2968:	learn: 0.4808935	total: 37.3s	remaining: 163ms
2969:	learn: 0.4808261	total: 37.3s	remaining: 151ms

2970:	learn: 0.4807343	total: 37.4s	remaining: 138ms
2971:	learn: 0.4806500	total: 37.4s	remaining: 126ms
2972:	learn: 0.4805680	total: 37.4s	remaining: 113ms
2973:	learn: 0.4804684	total: 37.4s	remaining: 101ms
2974:	learn: 0.4803890	total: 37.4s	remaining: 88ms
2975:	learn: 0.4803305	total: 37.4s	remaining: 75.4ms
2976:	learn: 0.4802638	total: 37.4s	remaining: 62.9ms
2977:	learn: 0.4802048	total: 37.4s	remaining: 50.3ms
2978:	learn: 0.4801317	total: 37.5s	remaining: 37.7ms
2979:	learn: 0.4800626	total: 37.5s	remaining: 25.1ms
2980:	learn: 0.4799773	total: 37.5s	remaining: 12.6ms
2981:	learn: 0.4799080	total: 37.5s	remaining: 0us

CatBoost $R^2 = 0.9992$

4.5 Training Random Forest

```
In [119... # Train Random Forest
rf_model = RandomForestRegressor(
    n_estimators=338,
    max_depth=23,
    min_samples_split=2,
    min_samples_leaf=1,
    random_state=42,
    n_jobs=-1,
    verbose=1
)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_val)
r2_rf = r2_score(y_val, y_pred_rf)
mae_rf = mean_absolute_error(y_val, y_pred_rf)
print(f"Random Forest  $R^2 = {r2_rf:.4f}$ ")
print(f"Random Forest MAE = {mae_rf:.2f}")
has_rf = True
```

```
[Parallel(n_jobs=-1)]: Using backend ThreadingBackend with 16 concurrent worker
s.
```

```
[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 8.1s
```

```
[Parallel(n_jobs=-1)]: Done 168 tasks | elapsed: 47.7s
```

Random Forest $R^2 = 0.9993$

Random Forest MAE = 0.16

```
[Parallel(n_jobs=-1)]: Done 338 out of 338 | elapsed: 1.5min finished
```

```
[Parallel(n_jobs=16)]: Using backend ThreadingBackend with 16 concurrent worker
s.
```

```
[Parallel(n_jobs=16)]: Done 18 tasks | elapsed: 0.0s
```

```
[Parallel(n_jobs=16)]: Done 168 tasks | elapsed: 0.0s
```

```
[Parallel(n_jobs=16)]: Done 338 out of 338 | elapsed: 0.0s finished
```

4.6 Training Gradient Boosting

```
In [120... # Train Gradient Boosting
gb_model = GradientBoostingRegressor(
```

```

n_estimators=500,
learning_rate=0.02,
max_depth=6,
min_samples_split=10,
min_samples_leaf=4,
subsample=0.8,
random_state=42,
verbose=1
)
gb_model.fit(X_train, y_train)
y_pred_gb = gb_model.predict(X_val)
r2_gb = r2_score(y_val, y_pred_gb)
mae_gb = mean_absolute_error(y_val, y_pred_gb)
print(f"Gradient Boosting R² = {r2_gb:.4f}")
print(f"Gradient Boosting MAE = {mae_gb:.2f}")
has_gb = True

```

Iter	Train Loss	OOB Improve	Remaining Time
1	1053.7148	41.9775	14.48m
2	989.5725	-57.1519	14.53m
3	972.5313	124.7679	14.58m
4	932.0776	23.8815	14.47m
5	908.5283	84.4823	14.43m
6	880.0313	57.7712	14.41m
7	839.9049	4.2641	14.35m
8	805.3826	21.0409	14.26m
9	770.4969	12.7098	14.17m
10	744.7120	42.8711	14.07m
20	500.2353	30.0560	13.53m
30	340.4776	-21.0344	13.26m
40	237.3980	22.0279	12.92m
50	159.0446	3.7506	12.69m
60	112.7574	5.0802	12.42m
70	76.7959	4.6581	12.15m
80	55.1111	2.7690	11.91m
90	38.5132	-0.5603	11.66m
100	27.7262	5.3697	11.37m
200	2.3966	0.1543	8.63m
300	0.9636	-0.5478	5.76m
400	0.8293	-0.0055	2.89m
500	0.6621	-0.1458	0.00s

Gradient Boosting R² = 0.9990
Gradient Boosting MAE = 0.58

4.7 Model R² Score Comparison

```

In [121]: # Create comparison table
model_comparison = pd.DataFrame({
    'Model': ['LightGBM', 'XGBoost', 'CatBoost', 'Random Forest', 'Gradient Boosting'],
    'R² Score': [r2_lgb, r2_xgb, r2_cat, r2_rf, r2_gb],
    'MAE': [mean_absolute_error(y_val, y_pred_lgb),
            mean_absolute_error(y_val, y_pred_xgb),

```

```

        mean_absolute_error(y_val, y_pred_cat),
        mae_rf,
        mae_gb]
    })

# Sort by R2 score
model_comparison = model_comparison.sort_values('R2 Score', ascending=False).r
print("\nModel Performance Comparison:")
print("="*60)
print(model_comparison.to_string(index=False))
print("="*60)

```

Model Performance Comparison:

```

=====
      Model  R2 Score      MAE
LightGBM   0.999356  0.345339
Random Forest 0.999323 0.161787
CatBoost   0.999221 0.375600
XGBoost    0.999086 0.354335
Gradient Boosting 0.998986 0.581315
=====

```

```

In [122... import matplotlib.pyplot as plt
import seaborn as sns

# Set style
sns.set_style("whitegrid")
plt.figure(figsize=(14, 6))

# Plot 1: R2 Score Comparison
plt.subplot(1, 2, 1)
colors = ['#2ecc71', '#3498db', '#9b59b6', '#e74c3c', '#f39c12']
bars = plt.barh(model_comparison['Model'], model_comparison['R2 Score'], color=
plt.xlabel('R2 Score', fontsize=12, fontweight='bold')
plt.ylabel('Model', fontsize=12, fontweight='bold')
plt.title('Model R2 Score Comparison', fontsize=14, fontweight='bold')
plt.xlim(0, 1.0)

# Add value labels on bars
for i, (model, score) in enumerate(zip(model_comparison['Model'], model_compar
    plt.text(score + 0.01, i, f'{score:.4f}', va='center', fontweight='bold')

# Plot 2: MAE Comparison
plt.subplot(1, 2, 2)
bars = plt.barh(model_comparison['Model'], model_comparison['MAE'], color=col
plt.xlabel('Mean Absolute Error', fontsize=12, fontweight='bold')
plt.ylabel('Model', fontsize=12, fontweight='bold')
plt.title('Model MAE Comparison (Lower is Better)', fontsize=14, fontweight='b

# Add value labels on bars
for i, (model, mae) in enumerate(zip(model_comparison['Model'], model_comparis
    plt.text(mae + 0.2, i, f'{mae:.2f}', va='center', fontweight='bold')

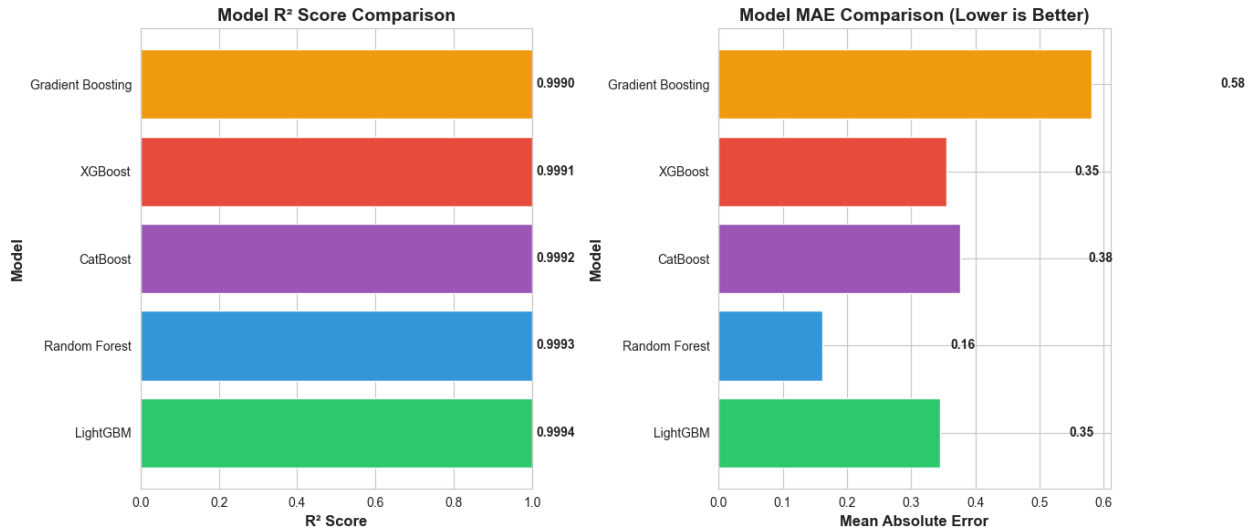
plt.tight_layout()

```



```
plt.show()
```

```
print(f"\n✓ Best Model: {model_comparison.iloc[0]['Model']} (R² = {model_compa
```



✓ Best Model: LightGBM (R² = 0.9994)

4.8 Hyperparameter Tuning of best three models using Bayesian Optimization

4.8.1 LightGBM Hyperparameter Tuning

```
In [123... # lgb_param_dist = {
#     'n_estimators': (1500, 3500),
#     'learning_rate': (0.005, 0.15),
#     'max_depth': (3, 15),
#     'num_leaves': (20, 200),
#     'min_child_samples': (5, 100),
#     'subsample': (0.5, 1.0),
#     'colsample_bytree': (0.5, 1.0),
#     'reg_alpha': (1e-9, 10.0),
#     'reg_lambda': (1e-9, 10.0),
#     'min_split_gain': (0.0, 1.0),
#     'min_child_weight': (1e-5, 1e2),
#     'random_state': 42,
#     'verbose': -1,
#     'n_jobs': -1
# }

# lgb_random_search = RandomizedSearchCV(
#     lgb.LGBMRegressor(random_state=42, verbose=-1),
#     param_distributions=lgb_param_dist,
#     n_iter=100,
#     cv=3,
#     scoring='r2',
#     random_state=42,
```

```
#     n_jobs=-1,
#     verbose=2
# )

# lgb_random_search.fit(X_train, y_train)
# print("LightGBM Best Params:", lgb_random_search.best_params_)
# print("LightGBM Best R2 Score:", lgb_random_search.best_score_)
```

4.8.2 XGBoost Hyperparameter Tuning

```
In [124... # xgb_param_dist = {
#         'n_estimators': ( 1500, 3000),
#         'learning_rate': ( 0.005, 0.15),
#         'max_depth': ( 3, 15),
#         'min_child_weight': ( 1, 15),
#         'subsample': (0.5, 1.0),
#         'colsample_bytree': ( 0.5, 1.0),
#         'colsample_bylevel': ( 0.5, 1.0),
#         'colsample_bynode': ( 0.5, 1.0),
#         'gamma': ( 1e-9, 5.0),
#         'reg_alpha': ( 1e-9, 10.0),
#         'reg_lambda': (1e-9, 10.0),
#         'max_delta_step': (0, 10),
#         'random_state': 42,
#         'tree_method': 'hist',
#         'n_jobs': -1
# }

# xgb_random_search = RandomizedSearchCV(
#     xgb.XGBRegressor(random_state=42, verbosity=0),
#     param_distributions=xgb_param_dist,
#     n_iter=100,
#     cv=3,
#     scoring='r2',
#     random_state=42,
#     n_jobs=-1,
#     verbose=2
# )

# xgb_random_search.fit(X_train, y_train)
# print("XGBoost Best Params:", xgb_random_search.best_params_)
# print("XGBoost Best R2 Score:", xgb_random_search.best_score_)
```

4.8.3 CatBoost Hyperparameter Tuning

```
In [125... # cat_param_dist = {
#         'iterations': randint(1000, 3000),
#         'learning_rate': uniform(0.01, 0.09),
#         'depth': randint(4, 10),
#         'l2_leaf_reg': uniform(1.0, 9.0),
#         'border_count': randint(32, 255),
```

```

#     'random_strength': uniform(1.0, 9.0),
#     'bagging_temperature': uniform(0.0, 1.0),
# }

# cat_random_search = RandomizedSearchCV(
#     CatBoostRegressor(random_state=42, verbose=0),
#     param_distributions=cat_param_dist,
#     n_iter=100,
#     cv=3,
#     scoring='r2',
#     random_state=42,
#     n_jobs=-1,
#     verbose=2
# )

# cat_random_search.fit(X_train, y_train)
# print("CatBoost Best Params:", cat_random_search.best_params_)
# print("CatBoost Best R2 Score:", cat_random_search.best_score_)

```

5. Feature Engineering of Test Data

5.1 Feature Engineering

```

In [126... # Prepare test data
test = sample_submission.copy()
test['book_theater_id'] = test['ID'].apply(lambda x: "_".join(x.split('_')[:2])
test['show_date'] = pd.to_datetime(test['ID'].apply(lambda x: x.split('_')[-1])

```

```

In [127... # Add theater statistics
test = test.merge(theater_performance, left_on='book_theater_id', right_index=
test['avg_audience'].fillna(test['avg_audience'].mean(), inplace=True)
test['min_audience'].fillna(test['min_audience'].mean(), inplace=True)
test['max_audience'].fillna(test['max_audience'].mean(), inplace=True)
test['std_audience'].fillna(test['std_audience'].mean(), inplace=True)

```

```

In [128... # Temporal features
test['year'] = test['show_date'].dt.year
test['month'] = test['show_date'].dt.month
test['day'] = test['show_date'].dt.day
test['day_of_week'] = test['show_date'].dt.day_of_week
test['weekofyear'] = test['show_date'].dt.isocalendar().week
test['day_of_year'] = test['show_date'].dt.dayofyear
test['quarter'] = test['show_date'].dt.quarter
test['day_of_quarter'] = test['show_date'].apply(
    lambda x: (x - pd.Timestamp(x.year, (x.quarter-1)*3 + 1, 1)).days + 1
)
test['is_weekend'] = test['day_of_week'].isin([5, 6]).astype(int)
test['is_month_start'] = test['show_date'].dt.is_month_start.astype(int)
test['is_month_end'] = test['show_date'].dt.is_month_end.astype(int)

```

```
test['is_quarter_start'] = test['show_date'].dt.is_quarter_start.astype(int)
test['is_quarter_end'] = test['show_date'].dt.is_quarter_end.astype(int)
```

```
In [129... # Interaction features
test['theater_x_month'] = test['book_theater_id'].astype(str) + '_' + test['month']
test['theater_x_dow'] = test['book_theater_id'].astype(str) + '_' + test['day_of_week']
test['month_x_dow'] = test['month'].astype(str) + '_' + test['day_of_week'].astype(int)
```

```
In [130... test.columns
```

```
Out[130... Index(['ID', 'audience_count', 'book_theater_id', 'show_date', 'avg_audience',
      'min_audience', 'max_audience', 'std_audience', 'total_shows', 'year',
      'month', 'day', 'day_of_week', 'weekofyear', 'day_of_year', 'quarter',
      'day_of_quarter', 'is_weekend', 'is_month_start', 'is_month_end',
      'is_quarter_start', 'is_quarter_end', 'theater_x_month',
      'theater_x_dow', 'month_x_dow'],
      dtype='object')
```

5.2 Creating Rolling Features in test data

```
In [131... combined = pd.concat([
    bookknow_visits[['book_theater_id', 'show_date', 'audience_count']],
    test[['book_theater_id', 'show_date']].assign(audience_count=np.nan)
], ignore_index=True).sort_values(['book_theater_id', 'show_date']).reset_index()
```

```
In [132... for lag in [1, 2, 3, 7, 14, 21, 28]:
    combined[f'lag_{lag}'] = combined.groupby('book_theater_id')['audience_count'].shift(lag)
```

```
In [133... for window in [3, 7, 14, 21, 28]:
    combined[f'rolling_mean_{window}'] = combined.groupby('book_theater_id')['audience_count'].rolling(window=window, min_periods=1).mean()
    combined[f'rolling_std_{window}'] = combined.groupby('book_theater_id')['audience_count'].rolling(window=window, min_periods=1).std()
    combined[f'rolling_min_{window}'] = combined.groupby('book_theater_id')['audience_count'].rolling(window=window, min_periods=1).min()
    combined[f'rolling_max_{window}'] = combined.groupby('book_theater_id')['audience_count'].rolling(window=window, min_periods=1).max()
```

```
In [134... for span in [3, 7, 14]:
    combined[f'ewm_{span}'] = combined.groupby('book_theater_id')['audience_count'].ewm(span=span, min_periods=1).mean()
```

```
In [135... test_with_lags = combined[combined['audience_count'].isna()].drop(columns=['audience_count'])
test = test.merge(test_with_lags, on=['book_theater_id', 'show_date'], how='left')
```

```
In [136... theater_mapping = dict(zip(le_theater.classes_, le_theater.transform(le_theater.classes_)))
test['book_theater_id_encoded'] = test['book_theater_id'].map(theater_mapping)
```

```
In [137... theater_month_mapping = dict(zip(le_theater_month.classes_, le_theater_month.transform(le_theater_month.classes_)))
```

```
test['theater_x_month_encoded'] = test['theater_x_month'].map(theater_month_ma
```

```
In [138... theater_dow_mapping = dict(zip(le_theater_dow.classes_, le_theater_dow.transfo
test['theater_x_dow_encoded'] = test['theater_x_dow'].map(theater_dow_mapping)
```

```
In [139... month_dow_mapping = dict(zip(le_month_dow.classes_, le_month_dow.transform(le_
test['month_x_dow_encoded'] = test['month_x_dow'].map(month_dow_mapping).fillr
```

```
In [140... test['days_from_avg'] = 0
test['pct_of_avg'] = 1
```

```
In [141... test_features = test[feature_cols]
```

```
In [142... for col in test_features.columns:
    if test_features[col].isnull().sum() > 0:
        median_val = X_train[col].median() if col in X_train.columns else 0
        test_features[col].fillna(median_val, inplace=True)
```

```
In [143... test_features.shape
```

```
Out[143... (38062, 53)
```

6. Making prediction using Ensemble Model

6.1 Finding Optimal Ensemble Weights

```
In [144... # Get validation predictions from the 3 models (LightGBM, Random Forest, CatBo
val_pred_lgb = lgb_model.predict(X_val)
val_pred_rf = rf_model.predict(X_val)
val_pred_cat = cat_model.predict(X_val)

# Stack predictions (only 3 models)
val_predictions = np.column_stack([
    val_pred_lgb,
    val_pred_rf,
    val_pred_cat
])
```

```
[Parallel(n_jobs=16)]: Using backend ThreadingBackend with 16 concurrent worker
s.
[Parallel(n_jobs=16)]: Done 18 tasks      | elapsed:    0.0s
[Parallel(n_jobs=16)]: Done 168 tasks    | elapsed:    0.0s
[Parallel(n_jobs=16)]: Done 338 out of 338 | elapsed:    0.1s finished
```

```
In [145... # Define objective function to minimize (negative R2 or MSE)
def ensemble_objective(weights, predictions, y_true):
```

```

weights = weights / np.sum(weights)
ensemble_pred = np.dot(predictions, weights)
r2 = r2_score(y_true, ensemble_pred)

return -r2

initial_weights = np.array([0.33, 0.33, 0.34])
constraints = ({'type': 'eq', 'fun': lambda w: np.sum(w) - 1})
bounds = [(0, 1) for _ in range(3)]

result = minimize(
    ensemble_objective,
    initial_weights,
    args=(val_predictions, y_val),
    method='SLSQP',
    bounds=bounds,
    constraints=constraints,
    options={'maxiter': 1000, 'ftol': 1e-9}
)

optimal_weights = result.x / np.sum(result.x)

print("OPTIMAL ENSEMBLE WEIGHTS:")
print(f"LightGBM:      {optimal_weights[0]:.4f} ({optimal_weights[0]*100:.2%}")
print(f"Random Forest: {optimal_weights[1]:.4f} ({optimal_weights[1]*100:.2%}")
print(f"CatBoost:      {optimal_weights[2]:.4f} ({optimal_weights[2]*100:.2%}")

```

```

OPTIMAL ENSEMBLE WEIGHTS:
LightGBM:      0.2410 (24.10%)
Random Forest: 0.4188 (41.88%)
CatBoost:      0.3402 (34.02%)

```

```

In [146... # Calculate performance with optimal weights
optimal_ensemble_pred = np.dot(val_predictions, optimal_weights)
optimal_r2 = r2_score(y_val, optimal_ensemble_pred)
optimal_mae = mean_absolute_error(y_val, optimal_ensemble_pred)

```

6.2 Generate Predictions

```

In [147... # Generate predictions from the 3 models (LightGBM, Random Forest, CatBoost)
pred_lgb = lgb_model.predict(test_features)
pred_rf = rf_model.predict(test_features)
pred_cat = cat_model.predict(test_features)

# Stack predictions
test_predictions = np.column_stack([pred_lgb, pred_rf, pred_cat])

# Use optimal weights for ensemble
predictions = np.dot(test_predictions, optimal_weights)
predictions = np.maximum(predictions, 0)

```

```
[Parallel(n_jobs=16)]: Using backend ThreadingBackend with 16 concurrent worker  
S.  
[Parallel(n_jobs=16)]: Done 18 tasks      | elapsed: 0.0s  
[Parallel(n_jobs=16)]: Done 168 tasks    | elapsed: 0.0s  
[Parallel(n_jobs=16)]: Done 338 out of 338 | elapsed: 0.0s finished
```

```
In [148... submission = pd.DataFrame({  
    'ID': sample_submission['ID'],  
    'audience_count': predictions  
})  
  
submission.to_csv('submission.csv', index=False)
```