# v2_sportbookprediction_automl

January 19, 2025

# 1 LeoVegas Prediction Analysis

This notebook demonstrates an end-to-end workflow for predicting turnover in a sportsbook dataset.

### 1.0.1 Objectives

- Conduct data cleaning and preprocessing.
- Perform exploratory data analysis (EDA).
- Engineer features to enhance predictive power.
- Apply advanced machine learning and time-series models.
- Implement causal inference to derive actionable insights.
- Use conformal predictions for reliable uncertainty estimation.

---

### 1.0.2 Prediction target

The prediction target for this task is: total_turn_over_EUR: The total amount of money bet by LeoVegas customers in a given time frame or event.

Why This Target? Business Relevance:

Total turnover directly reflects customer engagement and revenue generation, making it critical for strategic planning and decision-making. It enables forecasting for operational scaling, marketing budgets, and identifying high-turnover events. Stakeholder Impact:

Insights into betting patterns and trends can inform targeted promotions, resource allocation, and risk management. Predicting turnover provides actionable insights for league-specific marketing and customer segmentation.

#### Workflow to Create a Predictive Model

1. Data Understanding and Exploration Goals:

Identify patterns, seasonality, and trends in the data. Understand features influencing turnover, such as time, event, and league. Actions:

Perform exploratory data analysis (EDA) to uncover trends, anomalies, and correlations. Visualize turnover against features like hour, day_of_week, league, and event_country. Tools:

Pandas for data manipulation. Matplotlib and Seaborn for visualization.

2. Feature Engineering Purpose:

Create informative features to improve model performance. Examples:

Time-based features: Extract hour, day_of_week, month, and is_weekend from bet_placement_hour. Event-specific features: Calculate time_to_event (difference between eventStartDate and bet_placement_hour). Encode event_country and league using target or frequency encoding. Rolling and lag features: Add lag_1_turnover and rolling_3_turnover to capture temporal dependencies.

3. Modeling Approach We will explore two approaches:

Time Series Models:

Use models like SARIMA or Prophet to handle sequential dependencies and seasonal trends. Ideal for capturing long-term seasonality in turnover. Machine Learning Models:

Use tree-based models (e.g., LightGBM, XGBoost) for feature-rich tabular data. Handle non-linear relationships and interactions among features. AutoML:

Use FLAML or H2O AutoML to automate model selection and hyperparameter tuning.

4. Evaluation Metrics:

Root Mean Squared Error (RMSE): Measures average prediction error. Mean Absolute Error (MAE): Measures average absolute error. $R^2$ (Coefficient of Determination): Explains how much variance is captured by the model. Validation Strategy:

Time-based split: Ensure the training set precedes the test set to mimic real-world scenarios.

5. Uncertainty Quantification Why?

Provide stakeholders with prediction confidence intervals to aid in risk management. How?

Use conformal prediction via MAPIE or residual-based methods to quantify prediction uncertainty.

6. Visualization and Reporting Purpose:

Present results in a stakeholder-friendly manner. Deliverables:

Line plots of actual vs. predicted turnover. Confidence intervals to highlight uncertainty. Feature importance to explain model behavior.

[ ]:

```
```

## 1.1 1. Import Libraries

[ ]:

```
```

[1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
```

```python
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.statespace.sarimax import SARIMAX
from dowhy import CausalModel
from mapie.regression import MapieRegressor
```

## 1.2 2. Load and Preprocess Data

[2]:
```python
# Load dataset
data = pd.read_csv('../data/dataset.csv')
data
```

[2]:
```
          bet_placement_hour     eventId  \
0        2023-12-01 00:00:00  1020157185
1        2023-12-01 00:00:00  1020211480
2        2023-12-01 00:00:00  1020030708
3        2023-12-01 00:00:00  1020285783
4        2023-12-01 00:00:00  1020030708
…                        …          …
297282   2024-11-30 23:00:00  1021436280
297283   2024-11-30 23:00:00  1021851569
297284   2024-11-30 23:00:00  1022220957
297285   2024-11-30 23:00:00  1021703587
297286   2024-11-30 23:00:00  1021436335

                                                eventName  \
0        Oklahoma Sooners - Arkansas-Pine Bluff Golden …
1                LSU Tigers (W) - Virginia Tech Hokies (W)
2                             Miami Heat - Indiana Pacers
3               Union de Mar del Plata - Tomas De Rocamora
4                             Miami Heat - Indiana Pacers
…                                                     …
297282                       Utah Jazz - Dallas Mavericks
297283   Texas A&M Corpus Christi Islanders - Prairie V…
297284        (6) Houston Cougars - San Diego State Aztecs
297285         Winthrop Eagles (W) - Air Force Falcons (W)
297286              Detroit Pistons - Philadelphia 76ers

              eventStartDate event_country league  number_of_bets  \
0        2023-12-01 01:00:00         NCAAB    NaN               3
1        2023-12-01 02:00:00         NCAAW    NaN               4
2        2023-12-01 00:42:54           NBA    NaN             136
3        2023-12-01 00:00:00     Argentina    TNA             133
4        2023-12-01 00:30:00           NBA    NaN              25
…                        …           …    …             …
297282   2024-12-01 02:30:00           NBA    NaN             589
297283   2024-11-30 21:32:00         NCAAB    NaN              19
```

```
297284  2024-12-01 00:06:00         NCAAB    NaN               67
297285  2024-11-30 21:30:00         NCAAW    NaN               16
297286  2024-12-01 00:11:00           NBA    NaN               81

        total_turn_over_EUR
0                 49.582521
1                711.310730
2               5989.053830
3               1372.405046
4              18843.904216
…                         …
297282          4740.255659
297283           284.810951
297284             7.021176
297285             8.907591
297286           345.635463

[297287 rows x 8 columns]
```

[3]:
```
print(data.columns)
```

```
Index(['bet_placement_hour', 'eventId', 'eventName', 'eventStartDate',
       'event_country', 'league', 'number_of_bets', 'total_turn_over_EUR'],
      dtype='object')
```

## 2 Which should be target?

- total_number_over_EUR or number_of_bets?Conformal predicteion

1. Predicting number_of_bets Why Choose This? It reflects customer engagement and platform activity, which is critical for marketing, operational planning, and user retention strategies. Useful for forecasting workload on systems during peak times (e.g., popular games). Helps identify patterns in betting volume, which can influence promotions and product offerings. When to Choose This? If the primary goal is to analyze user behavior and optimize platform performance or engagement strategies.
2. Predicting total_turn_over_EUR Why Choose This? It directly ties to revenue and financial metrics. Helps in assessing risks and profitability by forecasting high-value betting periods. Useful for managing financial reserves or payouts during peak betting times. When to Choose This? If the primary goal is to manage monetary exposure or assess revenue trend

*Recommendation*

Since this is for a sports betting business, and both targets have unique importance, consider the following:

- If the focus is on operational planning and engagement: Use number_of_bets.
- If the focus is on financial management or revenue forecasting: Use total_turn_over_EUR.

[4]:
```python
# Identify problematic rows
print(data['bet_placement_hour'].head(10))  # Replace with the actual column␣
↪name
```

```
0    2023-12-01 00:00:00
1    2023-12-01 00:00:00
2    2023-12-01 00:00:00
3    2023-12-01 00:00:00
4    2023-12-01 00:00:00
5    2023-12-01 00:00:00
6    2023-12-01 00:00:00
7    2023-12-01 00:00:00
8    2023-12-01 00:00:00
9    2023-12-01 00:00:00
Name: bet_placement_hour, dtype: object
```

[5]:
```python
data['bet_placement_hour'] = pd.to_datetime(data['bet_placement_hour'],␣
↪errors='coerce')
data['eventStartDate'] = pd.to_datetime(data['eventStartDate'], errors='coerce')

# Check for invalid conversions
print(data[data['bet_placement_hour'].isna()])
print(data[data['eventStartDate'].isna()])
```

```
Empty DataFrame
Columns: [bet_placement_hour, eventId, eventName, eventStartDate, event_country,
league, number_of_bets, total_turn_over_EUR]
Index: []
        bet_placement_hour     eventId  \
17     2023-12-01 00:00:00  1020030708
27     2023-12-01 00:00:00  1020279406
31     2023-12-01 00:00:00  1020030714
36     2023-12-01 00:00:00  1020030711
40     2023-12-01 00:00:00  1020289801
…                    …           …
245956 2024-10-08 10:00:00  1020185517
248021 2024-10-11 13:00:00  1020185517
248253 2024-10-11 19:00:00  1020185517
248298 2024-10-11 20:00:00  1020185517
249957 2024-10-13 18:00:00  1020185517


                                      eventName eventStartDate  \
17                     Miami Heat - Indiana Pacers            NaT
27                   Barrio Parque - Gepu San Luis            NaT
31       Cleveland Cavaliers - Portland Trail Blazers            NaT
36                 New York Knicks - Detroit Pistons            NaT
40           Club Atlético Aguada - Urupan Basketball            NaT
…                                              …              …
```

```
245956                          WNBA Championship 2024           NaT
248021                          WNBA Championship 2024           NaT
248253                          WNBA Championship 2024           NaT
248298                          WNBA Championship 2024           NaT
249957                          WNBA Championship 2024           NaT

       event_country         league  number_of_bets  total_turn_over_EUR
17               NBA            NaN              55          6525.092734
27         Argentina            TNA             118          3234.949108
31               NBA            NaN            1070         10066.951183
36               NBA            NaN             249          8841.198662
40           Uruguay  Liga Uruguaya              23            24.228039
…                  …              …               …                    …
245956          WNBA            NaN               9             2.690213
248021          WNBA            NaN               8          2353.741367
248253          WNBA            NaN               1            41.882348
248298          WNBA            NaN              71            14.394768
249957          WNBA            NaN               5          2209.183403

[31166 rows x 8 columns]
```

[6]:

```python
# Check and handle invalid eventStartDate entries
data['bet_placement_hour'] = pd.to_datetime(data['bet_placement_hour'],␣
 ↪errors='coerce')
data['eventStartDate'] = pd.to_datetime(data['eventStartDate'], errors='coerce')

# Fill missing eventStartDate with bet_placement_hour
data['eventStartDate'] = data['eventStartDate'].
 ↪fillna(data['bet_placement_hour'])

# Fill missing league values
data['league'] = data['league'].fillna('Unknown')

# Drop any remaining invalid rows
data = data.dropna()

# Verify the processed dataset
print(data.head())
print(data.isna().sum())
```

```
  bet_placement_hour     eventId  \
0         2023-12-01  1020157185
1         2023-12-01  1020211480
2         2023-12-01  1020030708
3         2023-12-01  1020285783
4         2023-12-01  1020030708

                                          eventName      eventStartDate  \
```

```
0  Oklahoma Sooners - Arkansas-Pine Bluff Golden … 2023-12-01 01:00:00
1          LSU Tigers (W) - Virginia Tech Hokies (W) 2023-12-01 02:00:00
2                         Miami Heat - Indiana Pacers 2023-12-01 00:42:54
3         Union de Mar del Plata - Tomas De Rocamora 2023-12-01 00:00:00
4                         Miami Heat - Indiana Pacers 2023-12-01 00:30:00

  event_country   league  number_of_bets  total_turn_over_EUR
0         NCAAB  Unknown               3            49.582521
1         NCAAW  Unknown               4           711.310730
2           NBA  Unknown             136          5989.053830
3     Argentina      TNA             133          1372.405046
4           NBA  Unknown              25         18843.904216
bet_placement_hour     0
eventId                0
eventName              0
eventStartDate         0
event_country          0
league                 0
number_of_bets         0
total_turn_over_EUR    0
dtype: int64
```

[7]:

```
# Handle outliers
q_low = data['total_turn_over_EUR'].quantile(0.01)
q_high = data['total_turn_over_EUR'].quantile(0.99)
data = data[(data['total_turn_over_EUR'] >= q_low) &␣
 ↪(data['total_turn_over_EUR'] <= q_high)]
```

[8]:

```
data
```

[8]:

```
        bet_placement_hour     eventId  \
0       2023-12-01 00:00:00  1020157185
1       2023-12-01 00:00:00  1020211480
2       2023-12-01 00:00:00  1020030708
3       2023-12-01 00:00:00  1020285783
4       2023-12-01 00:00:00  1020030708
…                       …           …
297282 2024-11-30 23:00:00  1021436280
297283 2024-11-30 23:00:00  1021851569
297284 2024-11-30 23:00:00  1022220957
297285 2024-11-30 23:00:00  1021703587
297286 2024-11-30 23:00:00  1021436335

                                                eventName      eventStartDate  \
0       Oklahoma Sooners - Arkansas-Pine Bluff Golden … 2023-12-01 01:00:00
1               LSU Tigers (W) - Virginia Tech Hokies (W) 2023-12-01 02:00:00
2                             Miami Heat - Indiana Pacers 2023-12-01 00:42:54
```

```
3              Union de Mar del Plata - Tomas De Rocamora 2023-12-01 00:00:00
4                             Miami Heat - Indiana Pacers 2023-12-01 00:30:00
…                                                    …                  …
297282                        Utah Jazz - Dallas Mavericks 2024-12-01 02:30:00
297283  Texas A&M Corpus Christi Islanders - Prairie V… 2024-11-30 21:32:00
297284       (6) Houston Cougars - San Diego State Aztecs 2024-12-01 00:06:00
297285        Winthrop Eagles (W) - Air Force Falcons (W) 2024-11-30 21:30:00
297286               Detroit Pistons - Philadelphia 76ers 2024-12-01 00:11:00

       event_country   league  number_of_bets  total_turn_over_EUR
0              NCAAB  Unknown               3            49.582521
1              NCAAW  Unknown               4           711.310730
2                NBA  Unknown             136          5989.053830
3          Argentina      TNA             133          1372.405046
4                NBA  Unknown              25         18843.904216
…                …       …             …                    …
297282           NBA  Unknown             589          4740.255659
297283         NCAAB  Unknown              19           284.810951
297284         NCAAB  Unknown              67             7.021176
297285         NCAAW  Unknown              16             8.907591
297286           NBA  Unknown              81           345.635463

[291341 rows x 8 columns]
```

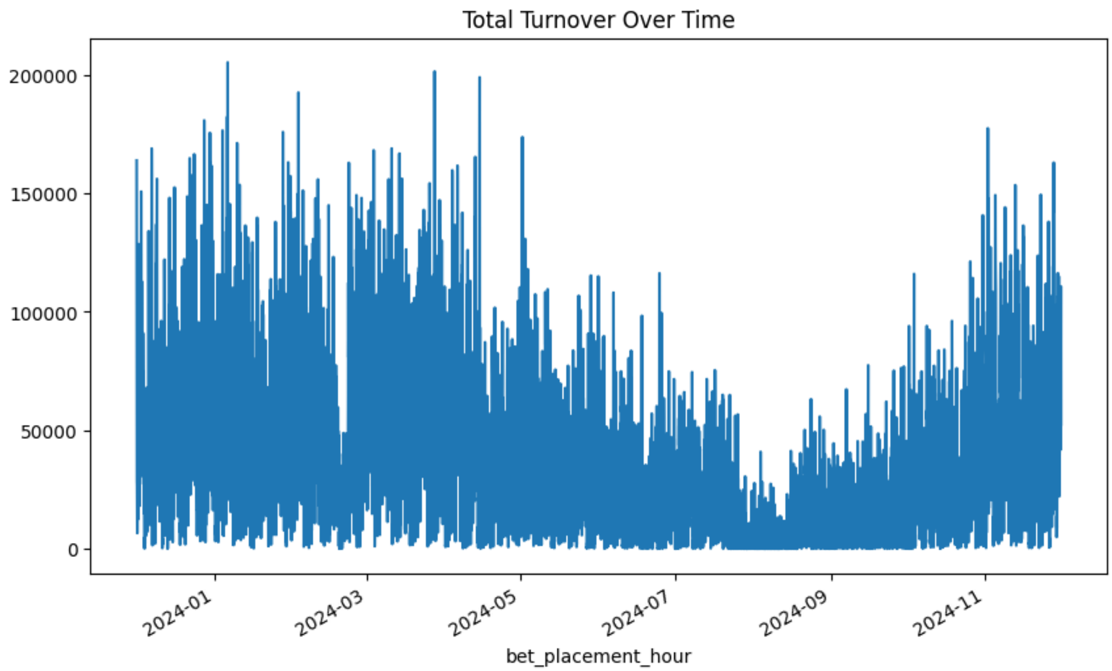
[9]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 291341 entries, 0 to 297286
Data columns (total 8 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   bet_placement_hour   291341 non-null  datetime64[ns]
 1   eventId              291341 non-null  int64
 2   eventName            291341 non-null  object
 3   eventStartDate       291341 non-null  datetime64[ns]
 4   event_country        291341 non-null  object
 5   league               291341 non-null  object
 6   number_of_bets       291341 non-null  int64
 7   total_turn_over_EUR  291341 non-null  float64
dtypes: datetime64[ns](2), float64(1), int64(2), object(3)
memory usage: 20.0+ MB
```

## 2.1 3. Exploratory Data Analysis (EDA)

[10]:
```python
# Time-series visualization
time_series = data.groupby('bet_placement_hour')['total_turn_over_EUR'].sum()
time_series.plot(figsize=(10, 6))
```

```
plt.title('Total Turnover Over Time')
plt.show()
```

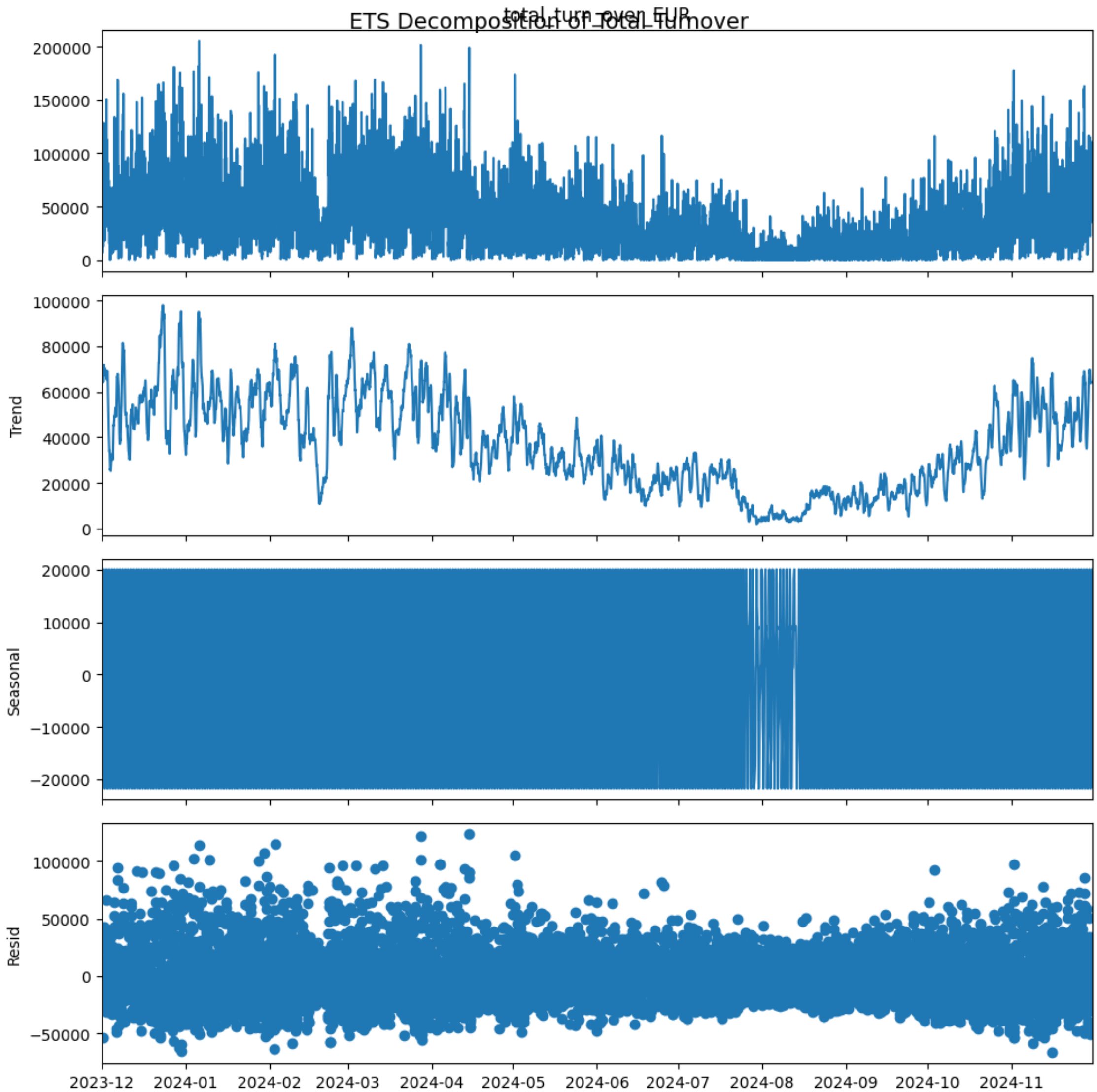



[11]:
```
# conduct ETS decomposition
from statsmodels.tsa.seasonal import seasonal_decompose

# Aggregate total turnover by hour for time-series decomposition
time_series = data.groupby('bet_placement_hour')['total_turn_over_EUR'].sum()

# Perform ETS decomposition
decomposition = seasonal_decompose(time_series, model='additive', period=24)  #␣
↪Assuming hourly periodicity

# Plot the decomposed components
plt.rcParams.update({'figure.figsize': (10, 10)})
decomposition.plot()
plt.suptitle('ETS Decomposition of Total Turnover', fontsize=14)
plt.show()
```




[12]:
```
from statsmodels.tsa.seasonal import seasonal_decompose

# Perform ETS decomposition
decomposition = seasonal_decompose(time_series, model='additive', period=24)  #␣
↪Assuming hourly periodicity

# Plot the decomposed components with improved aesthetics
plt.figure(figsize=(12, 10))

# Original series
plt.subplot(4, 1, 1)
plt.plot(decomposition.observed, color='blue')
plt.title('Original Series: Total Turnover Over Time', fontsize=14)
```

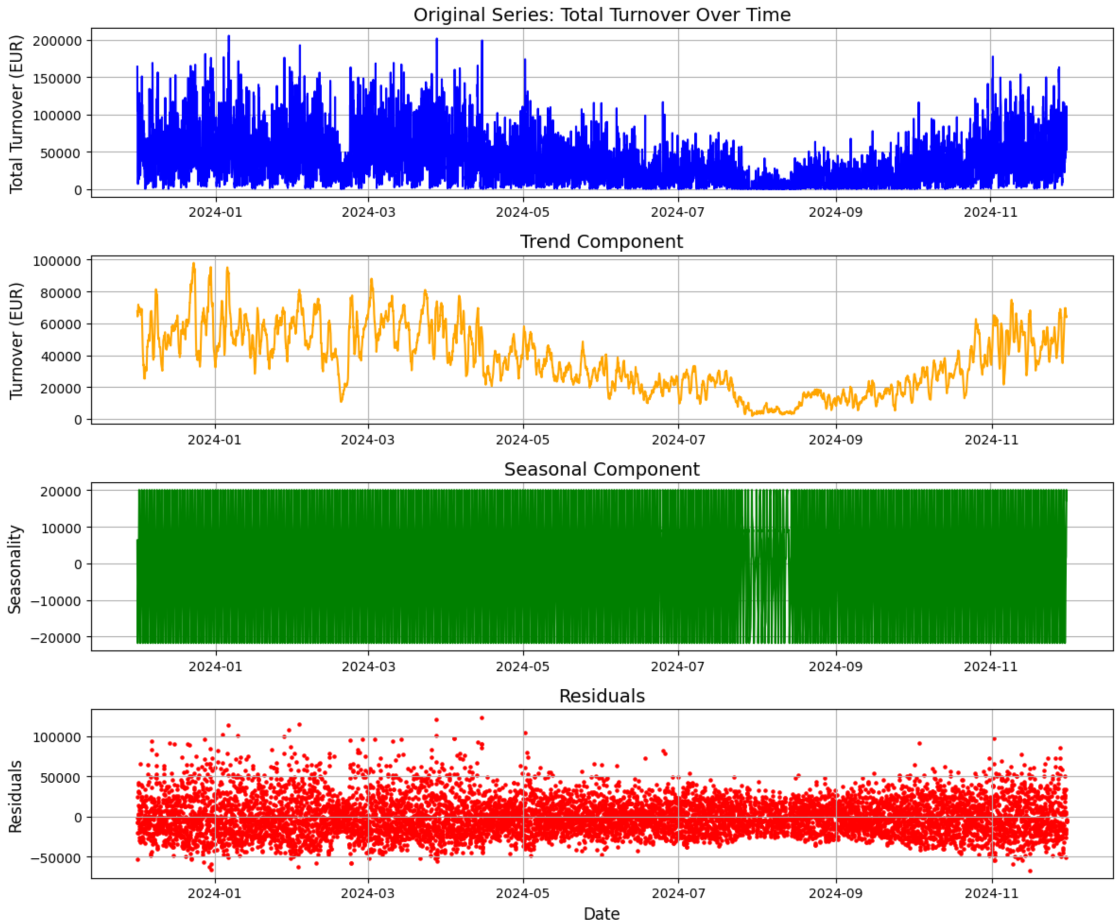
```python
plt.ylabel('Total Turnover (EUR)', fontsize=12)
plt.grid(True)

# Trend component
plt.subplot(4, 1, 2)
plt.plot(decomposition.trend, color='orange')
plt.title('Trend Component', fontsize=14)
plt.ylabel('Turnover (EUR)', fontsize=12)
plt.grid(True)

# Seasonal component
plt.subplot(4, 1, 3)
plt.plot(decomposition.seasonal, color='green')
plt.title('Seasonal Component', fontsize=14)
plt.ylabel('Seasonality', fontsize=12)
plt.grid(True)

# Residual component
plt.subplot(4, 1, 4)
plt.scatter(time_series.index, decomposition.resid, color='red', s=5)
plt.title('Residuals', fontsize=14)
plt.ylabel('Residuals', fontsize=12)
plt.xlabel('Date', fontsize=12)
plt.grid(True)

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




### 2.1.1 Explanation of the ETS Decomposition for a Business Decision Maker

The visualization breaks down the total turnover over time into its components: Original Series, Trend, Seasonality, and Residuals. Here's how each component can provide actionable insights for decision-making. - Trend (T): Determines whether revenue (turnover) is growing, declining, or stable over time. - Seasonality (S) Helps predict repeated patterns, such as daily or weekly customer behavior. - Residuals (R): Quantify randomness or noise, highlighting factors not captured by trend or seasonality. #### 1. Original Series: Total Turnover Over Time What it shows:

The raw total turnover data across the observed period. High fluctuations in turnover, with visible peaks and troughs. Turnover rises significantly toward the end of 2024. Implications for Business:

High-activity periods: Increased turnover during specific months, such as late 2024, may correspond to important basketball seasons or promotional events. Volatility management: Large fluctuations indicate the need for dynamic resource allocation (e.g., server capacity, customer support) to handle surges.

**2. Trend Component** *What it shows:*

Long-term growth or decline in turnover. A dip in mid-2024 followed by a strong recovery towards the end of 2024. Implications for Business:

*Market Analysis:* The mid-year dip might indicate an off-season or reduced customer engagement. The end-of-year growth suggests an opportunity to launch targeted promotions or campaigns to capitalize on peak betting activity. Strategic Planning: Use the trend data to forecast long-term performance and align marketing strategies with growth phases.

**3. Seasonal Component** *What it shows:*

Repeated patterns within the data, likely reflecting periodic betting behavior. For example, peaks and troughs in the seasonal component might align with daily game schedules or weekly betting trends.

*Implications for Business:*

Customer Behavior: Predictable seasonal patterns highlight customer engagement linked to events (e.g., evening games or weekend matches). Targeted Promotions: Schedule campaigns during high-activity periods to maximize customer engagement and revenue. Operational Efficiency: Allocate resources (e.g., marketing budgets or support teams) during high-demand hours or days.

**4. Residuals** *What it shows:*

Noise or randomness in the data after removing trend and seasonality. Large residuals suggest external factors affecting turnover that are not captured by the model. Implications for Business:

Unexplained Variations: Investigate large residuals to identify potential drivers, such as unexpected events (e.g., a championship or technical issues).

Model Refinement: The randomness indicates opportunities for improving predictive models by incorporating more external data (e.g., player stats, event popularity).

*Key Takeaways for Decision-Making* High-Activity Periods:

Focus efforts during late 2024 to leverage increased customer engagement. Plan promotional campaigns during periods of seasonal peaks. Market and Customer Insights:

Use the trend and seasonal data to understand when and why customers engage in betting. Align marketing strategies to maximize ROI during growth periods. Operational Adjustments:

Ensure the company's infrastructure can handle peak loads during high-turnover periods. Identify and address unexplained residuals to mitigate risks (e.g., unexpected surges or drops in turnover).

*How This Analysis Adds Value* For a business decision-maker, this decomposition provides a clear breakdown of patterns in customer behavior and operational needs. It ensures decisions are:

Data-Driven: Leverage turnover trends for revenue forecasting and budget planning. Customer-Centric: Align promotions and resources with periods of high engagement. Risk-Aware: Proactively address fluctuations and unexplained variations to maintain stable operations.

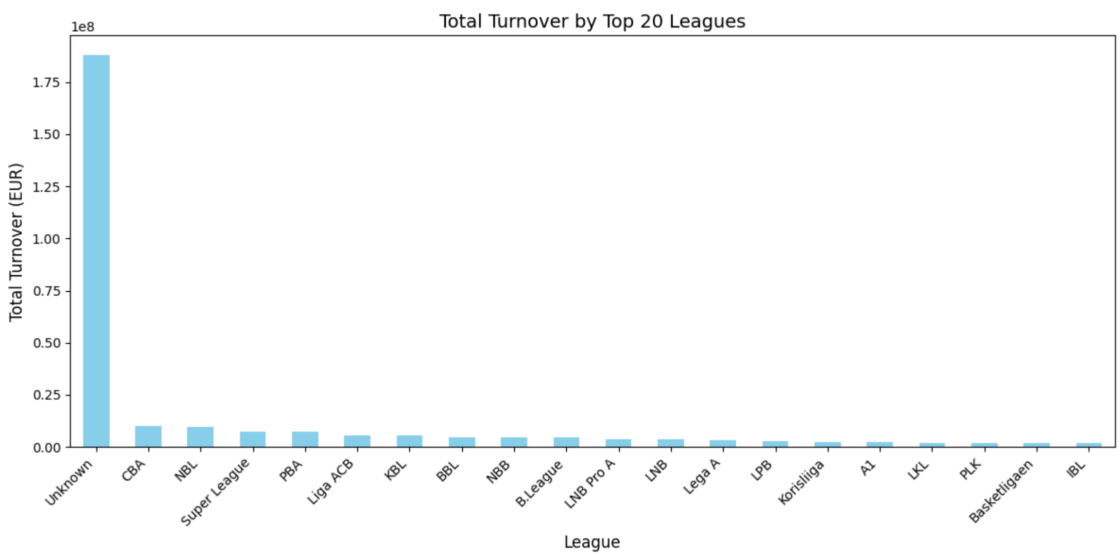
[13]:
```
import matplotlib.pyplot as plt

# Aggregate total turnover by league
```

```
league_turnover = data.groupby('league')['total_turn_over_EUR'].sum().
↪sort_values(ascending=False)

# Display only the top 20 leagues by total turnover
top_leagues = league_turnover.head(20)

# Plot the total turnover for top leagues
plt.figure(figsize=(12, 6))
top_leagues.plot(kind='bar', color='skyblue')
plt.title('Total Turnover by Top 20 Leagues', fontsize=14)
plt.xlabel('League', fontsize=12)
plt.ylabel('Total Turnover (EUR)', fontsize=12)
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.tight_layout()
plt.show()
```

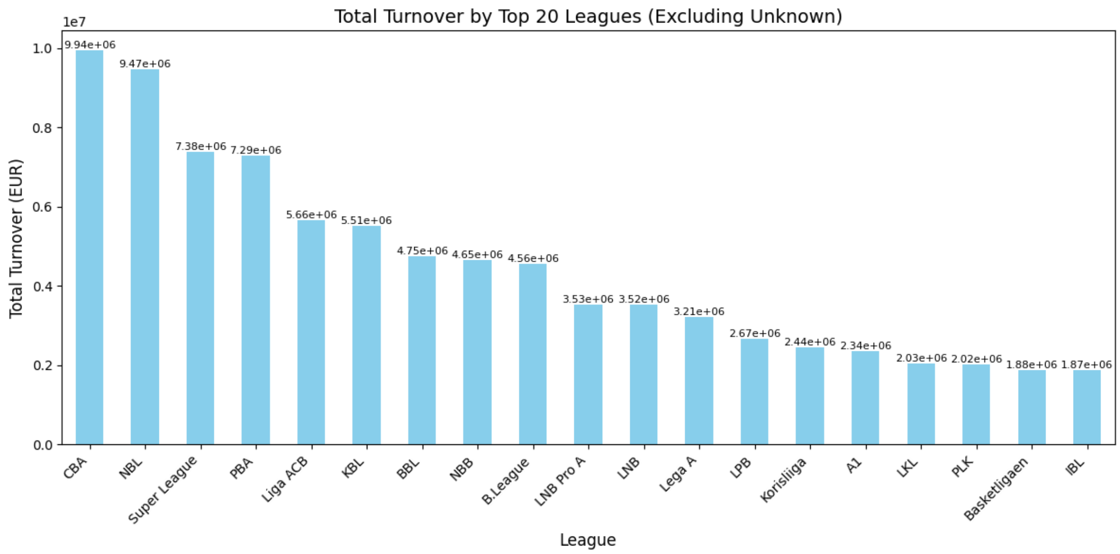



[14]:
```
# exclude unknown
# Exclude the 'Unknown' category
top_leagues_no_unknown = top_leagues[top_leagues.index != 'Unknown']

# Plot the total turnover for the top leagues (excluding Unknown)
ax = top_leagues_no_unknown.plot(kind='bar', color='skyblue', figsize=(12, 6))
for i, value in enumerate(top_leagues_no_unknown):
    ax.text(i, value, f'{value:.2e}', ha='center', va='bottom', fontsize=8)

plt.title('Total Turnover by Top 20 Leagues (Excluding Unknown)', fontsize=14)
plt.xlabel('League', fontsize=12)
plt.ylabel('Total Turnover (EUR)', fontsize=12)
```

```python
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.tight_layout()
plt.show()
```




### 2.1.2 Interpreting the Graph for Business Decision Makers

The graph displays the top 20 leagues (excluding "Unknown") ranked by their total turnover. Turnover here refers to the total amount of money wagered by customers on events in these leagues. Each bar represents the aggregate turnover for a league, giving insights into where the most significant betting activity occurs.

**Key Observations** *Highest Turnover Leagues:*

- CBA (Chinese Basketball Association) and NBL (National Basketball League) lead in total turnover, indicating they are the most popular leagues among customers.
- These leagues generate nearly €10 million each, representing major revenue streams.

*Long-Tail Distribution:*

The turnover drops significantly after the top 5 leagues (CBA, NBL, Super League, PBA, Liga ACB), highlighting a "long-tail" effect where most revenue comes from a few leagues, while others generate comparatively small amounts.

*Diverse Regional Preferences:*

- Leagues from different countries (e.g., China, Europe, and the Americas) are represented, suggesting that customer preferences are geographically distributed.
- The presence of leagues like KBL (Korean Basketball League) and BBL (British Basketball League) suggests opportunities to tailor marketing efforts regionally.

### 2.1.3 Recommendations for Earning More Revenue

To maximize revenue, focus on the following strategies:

1. Double Down on High-Turnover Leagues

Why? The top leagues (e.g., CBA, NBL) already drive the majority of turnover. Enhancing offerings for these leagues can increase engagement and revenue. How? Offer specialized promotions or bonuses for popular games in these leagues. Expand betting options (e.g., prop bets, live betting) to attract more wagers.

3. Target Marketing by Regional Preferences

Why? Different leagues appeal to different customer segments based on their location and interests. How? Focus marketing campaigns on regions where these leagues are most popular (e.g., promote CBA games in Asian markets). Use localized advertising during peak game seasons.

5. Explore Growth Opportunities in Mid-Tier Leagues Why? Mid-tier leagues (e.g., Super League, PBA, Liga ACB) have strong potential for growth as they already have significant turnover but less competition compared to top leagues. How? Partner with these leagues to create exclusive promotions. Invest in educating customers about lesser-known leagues to grow interest and engagement.

6. Seasonal Campaigns Why? Betting turnover often correlates with league schedules and major events. How? Focus promotional campaigns during playoffs, championships, and other high-visibility events for these leagues. Predict seasonal peaks using historical turnover data to allocate marketing resources effectively.

7. Long-Tail Strategy for Lower-Tier Leagues

Why? While smaller leagues (e.g., Basketligan, IBL) contribute less individually, collectively they provide an opportunity to grow aggregate turnover. How? Offer niche promotions or bundle smaller leagues with major ones in betting campaigns. Use targeted incentives like higher odds for these leagues to attract attention. Operational and Strategic Suggestions Diversify Betting Options:

Introduce more live betting options for high-turnover leagues like CBA and NBL, which can drive impulsive wagering during games. Offer specialized bets like player performance or quarter-by-quarter outcomes. Leverage Customer Data:

Analyze customer preferences for betting patterns in these leagues to personalize promotions. For example, identify customers who consistently bet on CBA games and offer them loyalty rewards. Monitor Emerging Trends:

Identify leagues that show consistent growth over time (e.g., mid-tier leagues like Liga ACB or BBL). Invest in growing these markets through sponsorships or strategic partnerships. Expand International Presence:

Partner with local broadcasters or sports organizations in regions where these leagues are popular to drive engagement.

*Expected Impact on Revenue*

Enhanced Engagement: By focusing on popular leagues, customer engagement will likely increase, driving higher turnover and subsequent revenue. Regional Growth: Targeting geographically diverse leagues can help expand the customer base. New Customer Acquisition: Promoting smaller leagues and offering unique bets can attract new customers and keep existing ones engaged. By aligning strategies with the insights from this chart, the business can significantly boost both turnover and profitability. Let me know if you'd like a deeper dive into forecasting specific growth opportunities!

### 2.2 4. Feature Engineering

[15]:
```
# Add time-based features
data['hour'] = data['bet_placement_hour'].dt.hour
data['day_of_week'] = data['bet_placement_hour'].dt.dayofweek
data['is_weekend'] = data['day_of_week'].isin([5, 6]).astype(int)
data['month'] = data['bet_placement_hour'].dt.month

# Calculate time to event
data['time_to_event'] = (data['eventStartDate'] - data['bet_placement_hour']).
 ↪dt.total_seconds()

# Add lag and rolling features
data['lag_1_turnover'] = data['total_turn_over_EUR'].shift(1)
data['rolling_3_turnover'] = data['total_turn_over_EUR'].rolling(window=3).
 ↪mean()
data.dropna(inplace=True)
```

*Purpose of Feature Engineering in This Context*

The goal of feature engineering in this example is to create new features from the existing data to better capture the relationships and patterns in the dataset. These engineered features help machine learning models and data analysis tools make more accurate predictions or gain deeper insights into customer behavior and betting trends.

*Purpose of Each Feature*

Time-Based Features:

hour: Extracts the hour of the day from the bet_placement_hour column.

Purpose: Captures the time-of-day betting trends. For example, betting might peak during evening games or specific hours.

day_of_week: Extracts the day of the week (e.g., Monday=0, Sunday=6).

Purpose: Identifies day-of-week patterns. For instance, betting might be higher during weekends or weekdays when major games are played.

is_weekend: Encodes whether the day is a weekend (1 for Saturday and Sunday, 0 otherwise). Purpose: Helps separate weekend-specific betting behavior, which could differ significantly from weekday behavior.

month: Extracts the month of the year.

Purpose: Captures seasonal patterns. Some months may have more betting activity due to playoffs, tournaments, or holidays.

*Event-Based Feature:*

time_to_event: Calculates the time difference (in seconds) between the eventStartDate and the bet_placement_hour.

Purpose: Tracks whether bets are placed early (pre-game) or closer to or during the event (live betting). This can help model different customer behaviors.

*Lag and Rolling Features:*

lag_1_turnover: The total turnover from the previous time step (lag of 1). Purpose: Captures the immediate past turnover to help models predict current turnover based on recent trends. rolling_3_turnover: The 3-period rolling average of the total turnover.

Purpose: Smooths out short-term fluctuations and captures broader trends over time, providing a more stable input for predictions.

Drop Missing Values (dropna):

Purpose: Ensures the dataset remains clean by removing rows where lagged or rolling features result in missing values (common at the beginning of time series).

## 2.3 5. AutoML and compare various baseline models

Here's a comprehensive workflow to build an AutoML pipeline for predicting total turnover. We'll use a combination of time series models and machine learning models and evaluate their performance to determine the best predictive model.

Step 1: Define the Prediction Target We aim to predict total_turn_over_EUR, the total amount of money bet by customers, based on historical data and engineered features like time, event details, and league-specific information.

Why?

Predicting total turnover helps optimize marketing, operations, and resource allocation. Businesses can anticipate high-demand periods and focus efforts on specific leagues or events.

## 2.4 Automl with flaml

- (just to see which models migth work better as baseline)

[16]:
```
#!pip install h2o scikit-learn matplotlib pandas
#!pip install mapie  # For conformal prediction
```

[17]:
```
#!pip install pycaret
#data.info()
```

[18]:
```
## Automl

#!pip install flaml
```

[19]:
```
# Import libraries
import pandas as pd
```

```python
import numpy as np
from flaml import AutoML
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from mapie.regression import MapieRegressor
import matplotlib.pyplot as plt


# Train-Test Split
features = ['hour', 'day_of_week', 'is_weekend', 'month', 'time_to_event',␣
↪'lag_1_turnover', 'rolling_3_turnover']
target = 'total_turn_over_EUR'
X = data[features]
y = data[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[20]:
```python
#automl and model selection
# FLAML AutoML
automl = AutoML()
automl_settings = {
    "time_budget": 120,  # Time in seconds
    "metric": "rmse",
    "task": "regression",
}
automl.fit(X_train=X_train, y_train=y_train, **automl_settings)

# Best Model
best_model = automl.model
print(f"Best Model: {best_model}")

# Predictions
predictions = automl.predict(X_test)
```

```
[flaml.automl.logger: 01-19 18:59:23] {1728} INFO - task = regression
[flaml.automl.logger: 01-19 18:59:23] {1739} INFO - Evaluation method: holdout
[flaml.automl.logger: 01-19 18:59:23] {1838} INFO - Minimizing error metric:
rmse
[flaml.automl.logger: 01-19 18:59:23] {1955} INFO - List of ML learners in
AutoML Run: ['lgbm', 'rf', 'xgboost', 'extra_tree', 'xgb_limitdepth', 'sgd']
[flaml.automl.logger: 01-19 18:59:23] {2258} INFO - iteration 0, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:23] {2393} INFO - Estimated sufficient time
budget=35873s. Estimated necessary time budget=256s.
[flaml.automl.logger: 01-19 18:59:23] {2442} INFO -  at 0.7s,   estimator lgbm's
best error=2431.1176,  best estimator lgbm's best error=2431.1176
[flaml.automl.logger: 01-19 18:59:23] {2258} INFO - iteration 1, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:23] {2442} INFO -  at 0.8s,   estimator lgbm's
```

```
best error=2431.1176,  best estimator lgbm's best error=2431.1176
[flaml.automl.logger: 01-19 18:59:23] {2258} INFO - iteration 2, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:23] {2442} INFO -  at 0.8s,   estimator lgbm's
best error=2083.1656,  best estimator lgbm's best error=2083.1656
[flaml.automl.logger: 01-19 18:59:23] {2258} INFO - iteration 3, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:23] {2442} INFO -  at 0.9s,   estimator lgbm's
best error=1824.5452,  best estimator lgbm's best error=1824.5452
[flaml.automl.logger: 01-19 18:59:23] {2258} INFO - iteration 4, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:24] {2442} INFO -  at 1.0s,   estimator lgbm's
best error=1824.5452,  best estimator lgbm's best error=1824.5452
[flaml.automl.logger: 01-19 18:59:24] {2258} INFO - iteration 5, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:24] {2442} INFO -  at 1.0s,   estimator lgbm's
best error=1824.5452,  best estimator lgbm's best error=1824.5452
[flaml.automl.logger: 01-19 18:59:24] {2258} INFO - iteration 6, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:24] {2442} INFO -  at 1.0s,   estimator lgbm's
best error=1824.5452,  best estimator lgbm's best error=1824.5452
[flaml.automl.logger: 01-19 18:59:24] {2258} INFO - iteration 7, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:24] {2442} INFO -  at 1.0s,   estimator lgbm's
best error=1824.5452,  best estimator lgbm's best error=1824.5452
[flaml.automl.logger: 01-19 18:59:24] {2258} INFO - iteration 8, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:24] {2442} INFO -  at 1.1s,   estimator lgbm's
best error=1773.8544,  best estimator lgbm's best error=1773.8544
[flaml.automl.logger: 01-19 18:59:24] {2258} INFO - iteration 9, current learner
lgbm
[flaml.automl.logger: 01-19 18:59:24] {2442} INFO -  at 1.1s,   estimator lgbm's
best error=1773.8544,  best estimator lgbm's best error=1773.8544
[flaml.automl.logger: 01-19 18:59:24] {2258} INFO - iteration 10, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:24] {2442} INFO -  at 1.4s,   estimator lgbm's
best error=1773.8544,  best estimator lgbm's best error=1773.8544
[flaml.automl.logger: 01-19 18:59:24] {2258} INFO - iteration 11, current
learner sgd
[flaml.automl.logger: 01-19 18:59:25] {2442} INFO -  at 2.3s,   estimator sgd's
best error=2910.7588,   best estimator lgbm's best error=1773.8544
[flaml.automl.logger: 01-19 18:59:25] {2258} INFO - iteration 12, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:25] {2442} INFO -  at 2.6s,   estimator lgbm's
best error=1739.0755,  best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:25] {2258} INFO - iteration 13, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:28] {2442} INFO -  at 5.2s,   estimator
```

```
xgboost's best error=2691.1756,       best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:28] {2258} INFO - iteration 14, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:28] {2442} INFO -  at 5.3s,   estimator
extra_tree's best error=2287.3051,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:28] {2258} INFO - iteration 15, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:28] {2442} INFO -  at 5.3s,   estimator
extra_tree's best error=2009.3647,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:28] {2258} INFO - iteration 16, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:28] {2442} INFO -  at 5.4s,   estimator
extra_tree's best error=2009.3647,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:28] {2258} INFO - iteration 17, current
learner sgd
[flaml.automl.logger: 01-19 18:59:29] {2442} INFO -  at 6.6s,   estimator sgd's
best error=2817.4128,   best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:29] {2258} INFO - iteration 18, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:29] {2442} INFO -  at 6.9s,   estimator lgbm's
best error=1739.0755,  best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:29] {2258} INFO - iteration 19, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:30] {2442} INFO -  at 7.0s,   estimator
extra_tree's best error=2009.3647,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:30] {2258} INFO - iteration 20, current
learner rf
[flaml.automl.logger: 01-19 18:59:30] {2442} INFO -  at 7.1s,   estimator rf's
best error=2098.4166,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:30] {2258} INFO - iteration 21, current
learner rf
[flaml.automl.logger: 01-19 18:59:30] {2442} INFO -  at 7.2s,   estimator rf's
best error=1840.8234,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:30] {2258} INFO - iteration 22, current
learner rf
[flaml.automl.logger: 01-19 18:59:30] {2442} INFO -  at 7.3s,   estimator rf's
best error=1840.8234,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:30] {2258} INFO - iteration 23, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:30] {2442} INFO -  at 7.7s,   estimator lgbm's
best error=1739.0755,  best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:30] {2258} INFO - iteration 24, current
learner rf
[flaml.automl.logger: 01-19 18:59:30] {2442} INFO -  at 7.8s,   estimator rf's
best error=1799.7169,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:30] {2258} INFO - iteration 25, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:30] {2442} INFO -  at 7.9s,   estimator
```

```
extra_tree's best error=1917.9546,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:30] {2258} INFO - iteration 26, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:31] {2442} INFO -  at 8.0s,   estimator
extra_tree's best error=1917.9546,    best estimator lgbm's best error=1739.0755
[flaml.automl.logger: 01-19 18:59:31] {2258} INFO - iteration 27, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:31] {2442} INFO -  at 8.4s,   estimator lgbm's
best error=1737.7987,  best estimator lgbm's best error=1737.7987
[flaml.automl.logger: 01-19 18:59:31] {2258} INFO - iteration 28, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:31] {2442} INFO -  at 8.5s,   estimator
xgboost's best error=2686.0867,       best estimator lgbm's best error=1737.7987
[flaml.automl.logger: 01-19 18:59:31] {2258} INFO - iteration 29, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:31] {2442} INFO -  at 8.5s,   estimator
extra_tree's best error=1917.9546,    best estimator lgbm's best error=1737.7987
[flaml.automl.logger: 01-19 18:59:31] {2258} INFO - iteration 30, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:31] {2442} INFO -  at 8.6s,   estimator
extra_tree's best error=1917.9546,    best estimator lgbm's best error=1737.7987
[flaml.automl.logger: 01-19 18:59:31] {2258} INFO - iteration 31, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:32] {2442} INFO -  at 9.5s,   estimator lgbm's
best error=1722.8537,  best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:32] {2258} INFO - iteration 32, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:33] {2442} INFO -  at 10.1s,  estimator lgbm's
best error=1722.8537,  best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:33] {2258} INFO - iteration 33, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:33] {2442} INFO -  at 10.2s,  estimator lgbm's
best error=1722.8537,  best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:33] {2258} INFO - iteration 34, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:33] {2442} INFO -  at 10.3s,  estimator
extra_tree's best error=1917.9546,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:33] {2258} INFO - iteration 35, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:33] {2442} INFO -  at 10.9s,  estimator lgbm's
best error=1722.8537,  best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:33] {2258} INFO - iteration 36, current
learner rf
[flaml.automl.logger: 01-19 18:59:34] {2442} INFO -  at 11.1s,  estimator rf's
best error=1799.7169,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:34] {2258} INFO - iteration 37, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:34] {2442} INFO -  at 11.2s,  estimator
```

```
extra_tree's best error=1819.1521,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:34] {2258} INFO - iteration 38, current
learner rf
[flaml.automl.logger: 01-19 18:59:33] {2442} INFO -  at 10.7s,  estimator rf's
best error=1799.7169,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:33] {2258} INFO - iteration 39, current
learner rf
[flaml.automl.logger: 01-19 18:59:33] {2442} INFO -  at 10.9s,  estimator rf's
best error=1799.7169,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:33] {2258} INFO - iteration 40, current
learner rf
[flaml.automl.logger: 01-19 18:59:34] {2442} INFO -  at 11.1s,  estimator rf's
best error=1799.7169,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:34] {2258} INFO - iteration 41, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:34] {2442} INFO -  at 11.2s,  estimator
extra_tree's best error=1805.5553,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:34] {2258} INFO - iteration 42, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:34] {2442} INFO -  at 11.4s,  estimator
extra_tree's best error=1805.5553,    best estimator lgbm's best error=1722.8537
[flaml.automl.logger: 01-19 18:59:34] {2258} INFO - iteration 43, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:34] {2442} INFO -  at 11.6s,  estimator lgbm's
best error=1715.3318,  best estimator lgbm's best error=1715.3318
[flaml.automl.logger: 01-19 18:59:34] {2258} INFO - iteration 44, current
learner rf
[flaml.automl.logger: 01-19 18:59:34] {2442} INFO -  at 11.8s,  estimator rf's
best error=1769.1912,    best estimator lgbm's best error=1715.3318
[flaml.automl.logger: 01-19 18:59:34] {2258} INFO - iteration 45, current
learner sgd
[flaml.automl.logger: 01-19 18:59:35] {2442} INFO -  at 12.8s,  estimator sgd's
best error=2792.3622,   best estimator lgbm's best error=1715.3318
[flaml.automl.logger: 01-19 18:59:35] {2258} INFO - iteration 46, current
learner rf
[flaml.automl.logger: 01-19 18:59:36] {2442} INFO -  at 13.0s,  estimator rf's
best error=1769.1912,    best estimator lgbm's best error=1715.3318
[flaml.automl.logger: 01-19 18:59:36] {2258} INFO - iteration 47, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:36] {2442} INFO -  at 13.2s,  estimator lgbm's
best error=1715.3318,  best estimator lgbm's best error=1715.3318
[flaml.automl.logger: 01-19 18:59:36] {2258} INFO - iteration 48, current
learner rf
[flaml.automl.logger: 01-19 18:59:36] {2442} INFO -  at 13.3s,  estimator rf's
best error=1769.1912,    best estimator lgbm's best error=1715.3318
[flaml.automl.logger: 01-19 18:59:36] {2258} INFO - iteration 49, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:37] {2442} INFO -  at 14.8s,  estimator lgbm's
```

```
best error=1715.3318,  best estimator lgbm's best error=1715.3318
[flaml.automl.logger: 01-19 18:59:37] {2258} INFO - iteration 50, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:38] {2442} INFO -  at 15.1s,  estimator lgbm's
best error=1709.3814,  best estimator lgbm's best error=1709.3814
[flaml.automl.logger: 01-19 18:59:38] {2258} INFO - iteration 51, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:38] {2442} INFO -  at 15.3s,  estimator
extra_tree's best error=1805.5553,    best estimator lgbm's best error=1709.3814
[flaml.automl.logger: 01-19 18:59:38] {2258} INFO - iteration 52, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:39] {2442} INFO -  at 16.0s,  estimator lgbm's
best error=1675.2370,  best estimator lgbm's best error=1675.2370
[flaml.automl.logger: 01-19 18:59:39] {2258} INFO - iteration 53, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:39] {2442} INFO -  at 16.9s,  estimator lgbm's
best error=1675.2370,  best estimator lgbm's best error=1675.2370
[flaml.automl.logger: 01-19 18:59:39] {2258} INFO - iteration 54, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:40] {2442} INFO -  at 17.5s,  estimator lgbm's
best error=1675.2370,  best estimator lgbm's best error=1675.2370
[flaml.automl.logger: 01-19 18:59:40] {2258} INFO - iteration 55, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:41] {2442} INFO -  at 18.2s,  estimator lgbm's
best error=1675.2370,  best estimator lgbm's best error=1675.2370
[flaml.automl.logger: 01-19 18:59:41] {2258} INFO - iteration 56, current
learner rf
[flaml.automl.logger: 01-19 18:59:41] {2442} INFO -  at 18.5s,  estimator rf's
best error=1763.8602,    best estimator lgbm's best error=1675.2370
[flaml.automl.logger: 01-19 18:59:41] {2258} INFO - iteration 57, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:42] {2442} INFO -  at 19.8s,  estimator lgbm's
best error=1673.5106,  best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:42] {2258} INFO - iteration 58, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:43] {2442} INFO -  at 20.5s,  estimator lgbm's
best error=1673.5106,  best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:43] {2258} INFO - iteration 59, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:44] {2442} INFO -  at 21.8s,  estimator lgbm's
best error=1673.5106,  best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:44] {2258} INFO - iteration 60, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:45] {2442} INFO -  at 22.5s,  estimator lgbm's
best error=1673.5106,  best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:45] {2258} INFO - iteration 61, current
learner rf
[flaml.automl.logger: 01-19 18:59:45] {2442} INFO -  at 22.7s,  estimator rf's
```

```
best error=1763.8602,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:45] {2258} INFO - iteration 62, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:46] {2442} INFO -  at 23.2s,  estimator lgbm's
best error=1673.5106,  best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:46] {2258} INFO - iteration 63, current
learner rf
[flaml.automl.logger: 01-19 18:59:46] {2442} INFO -  at 23.6s,  estimator rf's
best error=1732.2962,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:46] {2258} INFO - iteration 64, current
learner rf
[flaml.automl.logger: 01-19 18:59:46] {2442} INFO -  at 23.9s,  estimator rf's
best error=1732.2962,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:46] {2258} INFO - iteration 65, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:53] {2442} INFO -  at 30.0s,  estimator lgbm's
best error=1673.5106,  best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:53] {2258} INFO - iteration 66, current
learner rf
[flaml.automl.logger: 01-19 18:59:53] {2442} INFO -  at 30.6s,  estimator rf's
best error=1719.7439,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:53] {2258} INFO - iteration 67, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:53] {2442} INFO -  at 30.8s,  estimator
extra_tree's best error=1801.0672,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:53] {2258} INFO - iteration 68, current
learner rf
[flaml.automl.logger: 01-19 18:59:54] {2442} INFO -  at 31.3s,  estimator rf's
best error=1719.7439,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:54] {2258} INFO - iteration 69, current
learner lgbm
[flaml.automl.logger: 01-19 18:59:54] {2442} INFO -  at 31.6s,  estimator lgbm's
best error=1673.5106,  best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:54] {2258} INFO - iteration 70, current
learner rf
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.1s,  estimator rf's
best error=1719.7439,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 71, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.1s,  estimator
xgboost's best error=2434.7151,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 72, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.2s,  estimator
xgboost's best error=1999.9387,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 73, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.3s,  estimator
```

```
xgboost's best error=1999.9387,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 74, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.3s,  estimator
xgboost's best error=1999.9387,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 75, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.4s,  estimator
xgboost's best error=1826.5760,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 76, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.4s,  estimator
xgboost's best error=1789.2486,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 77, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.5s,  estimator
xgboost's best error=1775.9238,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 78, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.5s,  estimator
xgboost's best error=1775.9238,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 79, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:55] {2442} INFO -  at 32.6s,  estimator
xgboost's best error=1775.9238,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:55] {2258} INFO - iteration 80, current
learner rf
[flaml.automl.logger: 01-19 18:59:57] {2442} INFO -  at 34.0s,  estimator rf's
best error=1716.2559,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:57] {2258} INFO - iteration 81, current
learner extra_tree
[flaml.automl.logger: 01-19 18:59:57] {2442} INFO -  at 34.2s,  estimator
extra_tree's best error=1801.0672,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:57] {2258} INFO - iteration 82, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:57] {2442} INFO -  at 34.3s,  estimator
xgboost's best error=1775.9238,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:57] {2258} INFO - iteration 83, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:57] {2442} INFO -  at 34.5s,  estimator
xgboost's best error=1753.4656,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:57] {2258} INFO - iteration 84, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:57] {2442} INFO -  at 34.7s,  estimator
xgboost's best error=1753.4656,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:57] {2258} INFO - iteration 85, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:57] {2442} INFO -  at 34.8s,  estimator
```

```
xgboost's best error=1753.4656,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:57] {2258} INFO - iteration 86, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:57] {2442} INFO -  at 34.9s,  estimator
xgboost's best error=1753.4656,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:57] {2258} INFO - iteration 87, current
learner xgboost
[flaml.automl.logger: 01-19 18:59:58] {2442} INFO -  at 35.0s,  estimator
xgboost's best error=1753.4656,       best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:58] {2258} INFO - iteration 88, current
learner rf
[flaml.automl.logger: 01-19 18:59:59] {2442} INFO -  at 36.4s,  estimator rf's
best error=1716.2559,    best estimator lgbm's best error=1673.5106
[flaml.automl.logger: 01-19 18:59:59] {2258} INFO - iteration 89, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:00] {2442} INFO -  at 37.5s,  estimator lgbm's
best error=1652.4874,  best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:00] {2258} INFO - iteration 90, current
learner extra_tree
[flaml.automl.logger: 01-19 19:00:00] {2442} INFO -  at 37.8s,  estimator
extra_tree's best error=1801.0672,    best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:00] {2258} INFO - iteration 91, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:00] {2442} INFO -  at 37.9s,  estimator
xgboost's best error=1753.4656,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:00] {2258} INFO - iteration 92, current
learner sgd
[flaml.automl.logger: 01-19 19:00:01] {2442} INFO -  at 38.7s,  estimator sgd's
best error=2792.3622,   best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:01] {2258} INFO - iteration 93, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:02] {2442} INFO -  at 39.9s,  estimator
xgboost's best error=1735.1124,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:02] {2258} INFO - iteration 94, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:03] {2442} INFO -  at 40.0s,  estimator
xgboost's best error=1735.1124,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:03] {2258} INFO - iteration 95, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:03] {2442} INFO -  at 40.8s,  estimator lgbm's
best error=1652.4874,  best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:03] {2258} INFO - iteration 96, current
learner extra_tree
[flaml.automl.logger: 01-19 19:00:03] {2442} INFO -  at 40.9s,  estimator
extra_tree's best error=1801.0672,    best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:03] {2258} INFO - iteration 97, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:05] {2442} INFO -  at 42.5s,  estimator lgbm's
```

```
best error=1652.4874,  best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:05] {2258} INFO - iteration 98, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:05] {2442} INFO -  at 42.8s,  estimator
xgboost's best error=1722.4343,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:05] {2258} INFO - iteration 99, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:06] {2442} INFO -  at 43.3s,  estimator
xgboost's best error=1722.4343,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:06] {2258} INFO - iteration 100, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:13] {2442} INFO -  at 50.4s,  estimator lgbm's
best error=1652.4874,  best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:13] {2258} INFO - iteration 101, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:14] {2442} INFO -  at 51.2s,  estimator
xgboost's best error=1722.4343,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:14] {2258} INFO - iteration 102, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:15] {2442} INFO -  at 52.6s,  estimator
xgboost's best error=1684.1104,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:15] {2258} INFO - iteration 103, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:16] {2442} INFO -  at 53.4s,  estimator lgbm's
best error=1652.4874,  best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:16] {2258} INFO - iteration 104, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:19] {2442} INFO -  at 56.0s,  estimator
xgboost's best error=1684.1104,       best estimator lgbm's best error=1652.4874
[flaml.automl.logger: 01-19 19:00:19] {2258} INFO - iteration 105, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:21] {2442} INFO -  at 58.1s,  estimator lgbm's
best error=1642.3438,  best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:21] {2258} INFO - iteration 106, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:21] {2442} INFO -  at 58.2s,  estimator
xgb_limitdepth's best error=1754.7830,       best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:21] {2258} INFO - iteration 107, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:21] {2442} INFO -  at 58.2s,  estimator
xgb_limitdepth's best error=1754.7830,       best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:21] {2258} INFO - iteration 108, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:21] {2442} INFO -  at 58.3s,  estimator
xgb_limitdepth's best error=1754.7830,       best estimator lgbm's best
error=1642.3438
```

```
[flaml.automl.logger: 01-19 19:00:21] {2258} INFO - iteration 109, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:21] {2442} INFO -  at 58.4s,  estimator
xgb_limitdepth's best error=1754.7830,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:21] {2258} INFO - iteration 110, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:21] {2442} INFO -  at 58.5s,  estimator
xgb_limitdepth's best error=1754.7830,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:21] {2258} INFO - iteration 111, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:21] {2442} INFO -  at 58.6s,  estimator
xgb_limitdepth's best error=1686.3763,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:21] {2258} INFO - iteration 112, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:23] {2442} INFO -  at 60.4s,  estimator lgbm's
best error=1642.3438,  best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:23] {2258} INFO - iteration 113, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:23] {2442} INFO -  at 60.5s,  estimator
xgb_limitdepth's best error=1686.3763,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:23] {2258} INFO - iteration 114, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:23] {2442} INFO -  at 60.6s,  estimator
xgb_limitdepth's best error=1686.3763,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:23] {2258} INFO - iteration 115, current
learner sgd
[flaml.automl.logger: 01-19 19:00:24] {2442} INFO -  at 61.6s,  estimator sgd's
best error=2586.4406,   best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:24] {2258} INFO - iteration 116, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:24] {2442} INFO -  at 61.9s,  estimator
xgb_limitdepth's best error=1686.3763,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:24] {2258} INFO - iteration 117, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:29] {2442} INFO -  at 66.4s,  estimator
xgboost's best error=1684.1104,       best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:29] {2258} INFO - iteration 118, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:29] {2442} INFO -  at 66.5s,  estimator
xgb_limitdepth's best error=1677.6135,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:29] {2258} INFO - iteration 119, current
```

```
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:29] {2442} INFO -  at 66.7s,  estimator
xgb_limitdepth's best error=1655.6355,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:29] {2258} INFO - iteration 120, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:29] {2442} INFO -  at 66.8s,  estimator
xgb_limitdepth's best error=1655.6355,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:29] {2258} INFO - iteration 121, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:29] {2442} INFO -  at 66.9s,  estimator
xgb_limitdepth's best error=1655.6355,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:29] {2258} INFO - iteration 122, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:30] {2442} INFO -  at 67.1s,  estimator
xgb_limitdepth's best error=1655.6355,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:30] {2258} INFO - iteration 123, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:30] {2442} INFO -  at 67.4s,  estimator
xgb_limitdepth's best error=1655.6355,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:30] {2258} INFO - iteration 124, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:30] {2442} INFO -  at 67.7s,  estimator
xgb_limitdepth's best error=1655.6355,        best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:30] {2258} INFO - iteration 125, current
learner xgboost
[flaml.automl.logger: 01-19 19:00:33] {2442} INFO -  at 70.7s,  estimator
xgboost's best error=1672.1135,       best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:33] {2258} INFO - iteration 126, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:43] {2442} INFO -  at 80.3s,  estimator lgbm's
best error=1642.3438,  best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:43] {2258} INFO - iteration 127, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:43] {2442} INFO -  at 80.8s,  estimator lgbm's
best error=1642.3438,  best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:43] {2258} INFO - iteration 128, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:44] {2442} INFO -  at 81.3s,  estimator lgbm's
best error=1642.3438,  best estimator lgbm's best error=1642.3438
[flaml.automl.logger: 01-19 19:00:44] {2258} INFO - iteration 129, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:44] {2442} INFO -  at 81.5s,  estimator
```

```
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:44] {2258} INFO - iteration 130, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:44] {2442} INFO -  at 81.7s,  estimator
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:44] {2258} INFO - iteration 131, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:44] {2442} INFO -  at 81.8s,  estimator
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:44] {2258} INFO - iteration 132, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:45] {2442} INFO -  at 82.1s,  estimator
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:45] {2258} INFO - iteration 133, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:45] {2442} INFO -  at 82.2s,  estimator
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1642.3438
[flaml.automl.logger: 01-19 19:00:45] {2258} INFO - iteration 134, current
learner lgbm
[flaml.automl.logger: 01-19 19:00:56] {2442} INFO -  at 93.1s,  estimator lgbm's
best error=1636.7947,  best estimator lgbm's best error=1636.7947
[flaml.automl.logger: 01-19 19:00:56] {2258} INFO - iteration 135, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:56] {2442} INFO -  at 93.7s,  estimator
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1636.7947
[flaml.automl.logger: 01-19 19:00:56] {2258} INFO - iteration 136, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:56] {2442} INFO -  at 93.9s,  estimator
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1636.7947
[flaml.automl.logger: 01-19 19:00:56] {2258} INFO - iteration 137, current
learner sgd
[flaml.automl.logger: 01-19 19:00:57] {2442} INFO -  at 94.0s,  estimator sgd's
best error=2586.4406,   best estimator lgbm's best error=1636.7947
[flaml.automl.logger: 01-19 19:00:57] {2258} INFO - iteration 138, current
learner xgb_limitdepth
[flaml.automl.logger: 01-19 19:00:57] {2442} INFO -  at 94.6s,  estimator
xgb_limitdepth's best error=1655.6355,       best estimator lgbm's best
error=1636.7947
[flaml.automl.logger: 01-19 19:00:57] {2258} INFO - iteration 139, current
learner lgbm
[flaml.automl.logger: 01-19 19:01:23] {2442} INFO -  at 120.4s, estimator lgbm's
```

```
best error=1636.7947,  best estimator lgbm's best error=1636.7947
[flaml.automl.logger: 01-19 19:01:40] {2685} INFO - retrain lgbm for 17.3s
[flaml.automl.logger: 01-19 19:01:40] {2688} INFO - retrained model:
LGBMRegressor(colsample_bytree=0.8322686015889758,
              learning_rate=0.05636857077048371, max_bin=1023,
              min_child_samples=9, n_estimators=2569, n_jobs=-1, num_leaves=12,
              reg_alpha=0.04421265048124132, reg_lambda=0.7858555823984512,
              verbose=-1)
[flaml.automl.logger: 01-19 19:01:40] {1985} INFO - fit succeeded
[flaml.automl.logger: 01-19 19:01:40] {1986} INFO - Time taken to find the best
model: 93.12063360214233
Best Model: <flaml.automl.model.LGBMEstimator object at 0x7fbdaf0982f0>
```

[21]:
```python
# Evaluate performance
rmse = np.sqrt(mean_squared_error(y_test, predictions))
mae = mean_absolute_error(y_test, predictions)
r2 = r2_score(y_test, predictions)

print(f"FLAML AutoML - RMSE: {rmse:.2f}, MAE: {mae:.2f}, R2: {r2:.2f}")
```

```
FLAML AutoML - RMSE: 1645.93, MAE: 719.10, R2: 0.66
```