Forecasting Domestic Tourism Revenue

(Machine Learning Final Report)

2023036299 김진욱
2023076508 박준호
2023094275 조준형

ㄴ

Submission Date: June 1, 2025

Table of Contents

introduction

Body

Conclusion

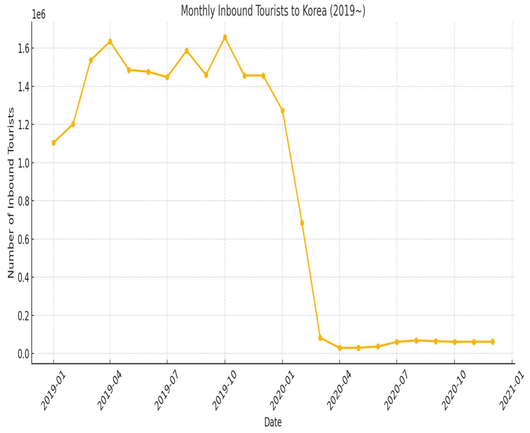
The tourism industry has a positive impact on the national economy in various ways, including job creation, foreign currency earnings, and revitalization of local economies. In particular, South Korea has long been regarded as a highly competitive tourism destination in Asia, and attracting foreign tourists has contributed not only to national branding but also to the spread of Korean culture. However, the number of foreign tourists visiting South Korea—having reached a record high of 17.5 million in 2019—was severely impacted by the COVID-19 pandemic. This decline in tourism led to a significant reduction in national revenue, a ripple effect across tourism-related industries, increased unemployment, and a contraction in job opportunities. In addition, the sharp drop in international visitors resulted in lost opportunities for cultural exchange and left large-scale tourism infrastructure underutilized.

The graphs below illustrate the decline in both the number of tourists and tourism revenue due to the COVID-19 pandemic.

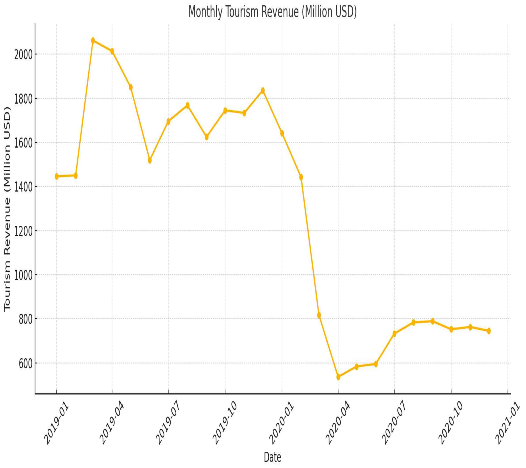
As shown, the tourism industry is highly vulnerable to external shocks, and the extent of the damage can be widespread. Therefore, it is essential to establish proactive response strategies by forecasting changes in tourism revenue through predictive modeling.

We selected this topic with the aim of building a model that considers various scenarios to forecast future revenue and prepare for potential losses when the tourism industry is affected.

(Monthly Foreign Tourists)



(Monthly Tourism Revenue)



This study collected relevant data from credible public institutions to forecast domestic tourism revenue. Major data sources include FRED, the Korea Meteorological Administration, Korea Customs Service, Seoul Open Data Plaza, SEDAILY, Korea Tourism Data Lab, and Statistics Korea.

The dataset used for analysis spans a period of ten years, from January 2015 to

December 2024. The dependent variable is monthly tourism revenue, recorded in units of one thousand U.S. dollars.

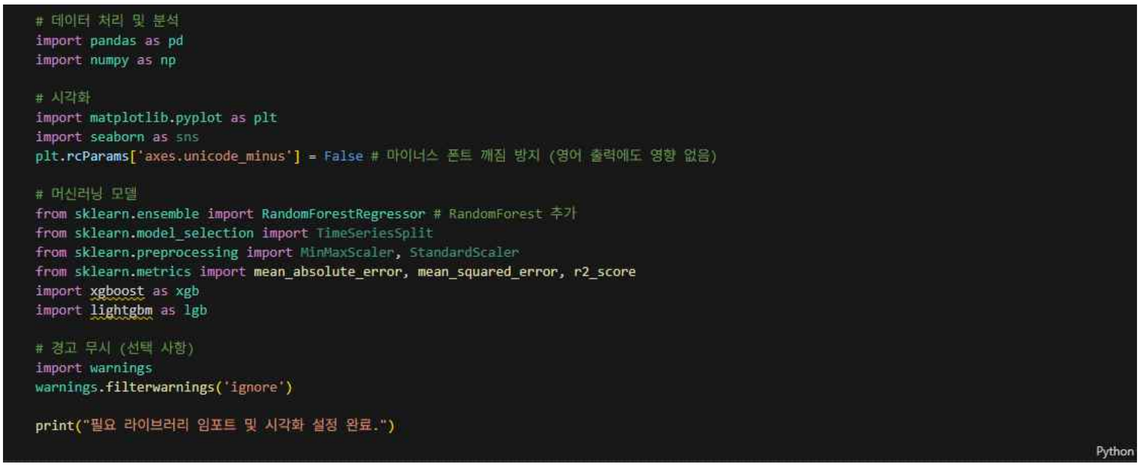
Over 20 independent variables were used. Representative features include the number of inbound tourists, average spending per person, number of arrivals, number of international flight arrivals at Incheon Airport, exchange rates, and global oil prices—all of which can directly influence tourism revenue.

To reflect foreign tourist spending, duty-free foreign transaction amounts and the number of visitors were included. In addition, Google Trends data was used to quantify public interest from key countries.

Considering the time-series nature of the data, lag features and moving average features were generated. For example, lag variables include tourism revenue values from 1, 3, 6, and 12 months prior, and short-term trend indicators were captured using 3-month and 6-month moving averages. These features enable the model to learn seasonal patterns, recurring trends, and the lagged effects of exogenous variables.

Lastly, a binary variable indicating the pandemic period (0 or 1) was added to account for the impact of special events like COVID-19. This variable helps the model distinguish between pandemic and non-pandemic periods in terms of tourism revenue.

Before building the model, the necessary libraries were imported.

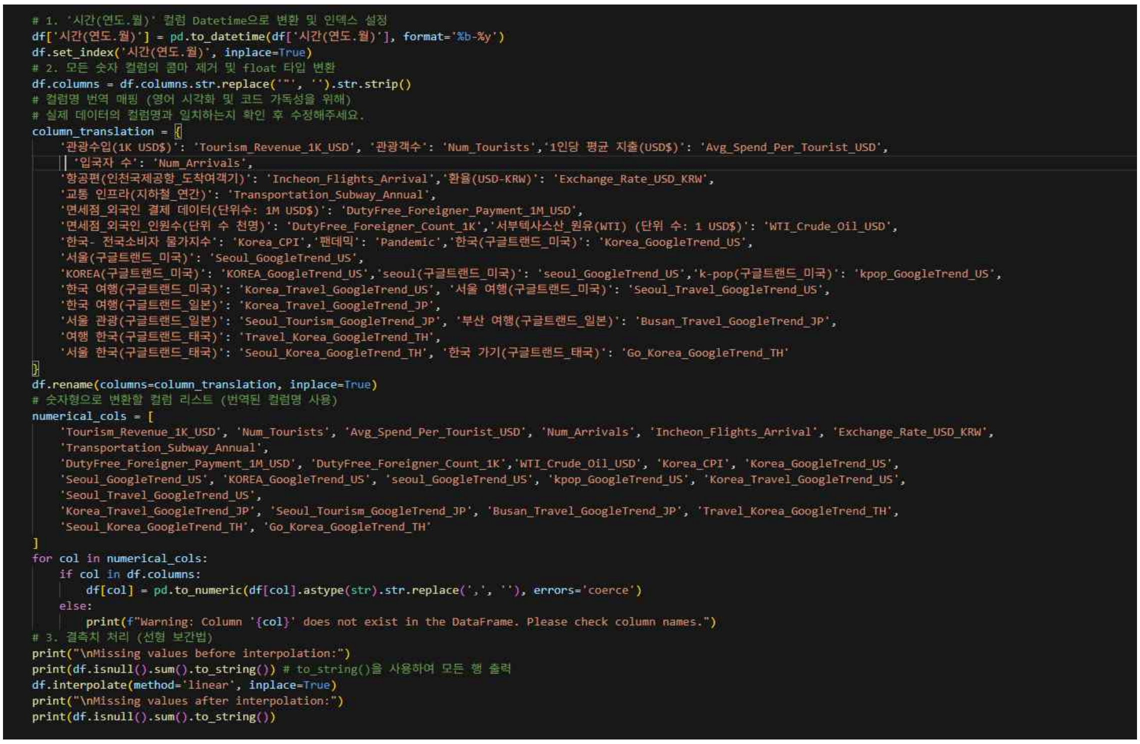


```
# 데이터 처리 및 분석
import pandas as pd
import numpy as np

# 시각화
import matplotlib.pyplot as plt
import seaborn as sns
plt.rcParams['axes.unicode_minus'] = False # 마이너스 폰트 깨짐 방지 (영어 출력에도 영향 없음)

# 머신러닝 모델
from sklearn.ensemble import RandomForestRegressor # RandomForest 추가
from sklearn.model_selection import TimeSeriesSplit
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import xgboost as xgb
import lightgbm as lgb

# 경고 무시 (선택 사항)
import warnings
warnings.filterwarnings('ignore')

print("필요 라이브러리 임포트 및 시각화 설정 완료.")
```

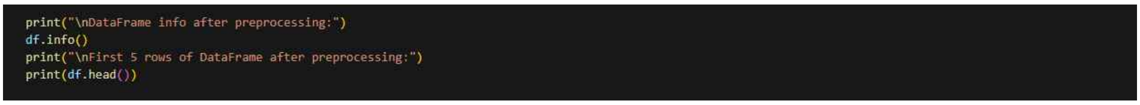
The purpose of the code below is to preprocess the raw data into a format suitable for analysis.

The date information was converted into a time-series index, and all column names originally written in Korean were renamed to English for ease of analysis and visualization.

Additionally, all numerical variables were converted to float type after removing thousand separators, and missing values were handled using linear interpolation.



```python
# 1. '시간(연도.월)' 컬럼 Datetime으로 변환 및 인덱스 설정
df['시간(연도.월)'] = pd.to_datetime(df['시간(연도.월)'], format='%b-%y')
df.set_index('시간(연도.월)', inplace=True)
# 2. 모든 숫자 컬럼의 콤마 제거 및 float 타입 변환
df.columns = df.columns.str.replace('"', '').str.strip()
# 컬럼명 번역 매핑 (영어 시각화 및 코드 가독성을 위해)
# 실제 데이터의 컬럼명과 일치하는지 확인 후 수정해주세요.
column_translation = {
    '관광수입(1K USD$)': 'Tourism_Revenue_1K_USD', '관광객수': 'Num_Tourists','1인당 평균 지출(USD$)': 'Avg_Spend_Per_Tourist_USD',
    | '입국자 수': 'Num_Arrivals',
    '항공편(인천국제공항_도착여객기)': 'Incheon_Flights_Arrival','환율(USD-KRW)': 'Exchange_Rate_USD_KRW',
    '교통 인프라(지하철_연간)': 'Transportation_Subway_Annual',
    '면세점_외국인 결제 데이터(단위수: 1M USD$)': 'DutyFree_Foreigner_Payment_1M_USD',
    '면세점_외국인_인원수(단위 수 천명)': 'DutyFree_Foreigner_Count_1K','서부텍사스산_원유(WTI) (단위 수: 1 USD$)': 'WTI_Crude_Oil_USD',
    '한국- 전국소비자 물가지수': 'Korea_CPI','팬데믹': 'Pandemic','한국(구글트랜드_미국)': 'Korea_GoogleTrend_US',
    '서울(구글트랜드_미국)': 'Seoul_GoogleTrend_US',
    'KOREA(구글트랜드_미국)': 'KOREA_GoogleTrend_US','seoul(구글트랜드_미국)': 'seoul_GoogleTrend_US','k-pop(구글트랜드_미국)': 'kpop_GoogleTrend_US',
    '한국 여행(구글트랜드_미국)': 'Korea_Travel_GoogleTrend_US', '서울 여행(구글트랜드_미국)': 'Seoul_Travel_GoogleTrend_US',
    '한국 여행(구글트랜드_일본)': 'Korea_Travel_GoogleTrend_JP',
    '서울 관광(구글트랜드_일본)': 'Seoul_Tourism_GoogleTrend_JP', '부산 여행(구글트랜드_일본)': 'Busan_Travel_GoogleTrend_JP',
    '여행 한국(구글트랜드_태국)': 'Travel_Korea_GoogleTrend_TH',
    '서울 한국(구글트랜드_태국)': 'Seoul_Korea_GoogleTrend_TH', '한국 가기(구글트랜드_태국)': 'Go_Korea_GoogleTrend_TH'
}
df.rename(columns=column_translation, inplace=True)
# 숫자형으로 변환할 컬럼 리스트 (번역된 컬럼명 사용)
numerical_cols = [
    'Tourism_Revenue_1K_USD', 'Num_Tourists', 'Avg_Spend_Per_Tourist_USD', 'Num_Arrivals', 'Incheon_Flights_Arrival', 'Exchange_Rate_USD_KRW',
    'Transportation_Subway_Annual',
    'DutyFree_Foreigner_Payment_1M_USD', 'DutyFree_Foreigner_Count_1K','WTI_Crude_Oil_USD', 'Korea_CPI', 'Korea_GoogleTrend_US',
    'Seoul_GoogleTrend_US', 'KOREA_GoogleTrend_US', 'seoul_GoogleTrend_US', 'kpop_GoogleTrend_US', 'Korea_Travel_GoogleTrend_US',
    'Seoul_Travel_GoogleTrend_US',
    'Korea_Travel_GoogleTrend_JP', 'Seoul_Tourism_GoogleTrend_JP', 'Busan_Travel_GoogleTrend_JP', 'Travel_Korea_GoogleTrend_TH',
    'Seoul_Korea_GoogleTrend_TH', 'Go_Korea_GoogleTrend_TH'
]
for col in numerical_cols:
    if col in df.columns:
        df[col] = pd.to_numeric(df[col].astype(str).str.replace(',', ''), errors='coerce')
    else:
        print(f"Warning: Column '{col}' does not exist in the DataFrame. Please check column names.")
# 3. 결측치 처리 (선형 보간법)
print("\nMissing values before interpolation:")
print(df.isnull().sum().to_string()) # to_string()을 사용하여 모든 행 출력
df.interpolate(method='linear', inplace=True)
print("\nMissing values after interpolation:")
print(df.isnull().sum().to_string())
```



```python
print("\nDataFrame info after preprocessing:")
df.info()
print("\nFirst 5 rows of DataFrame after preprocessing:")
print(df.head())
```

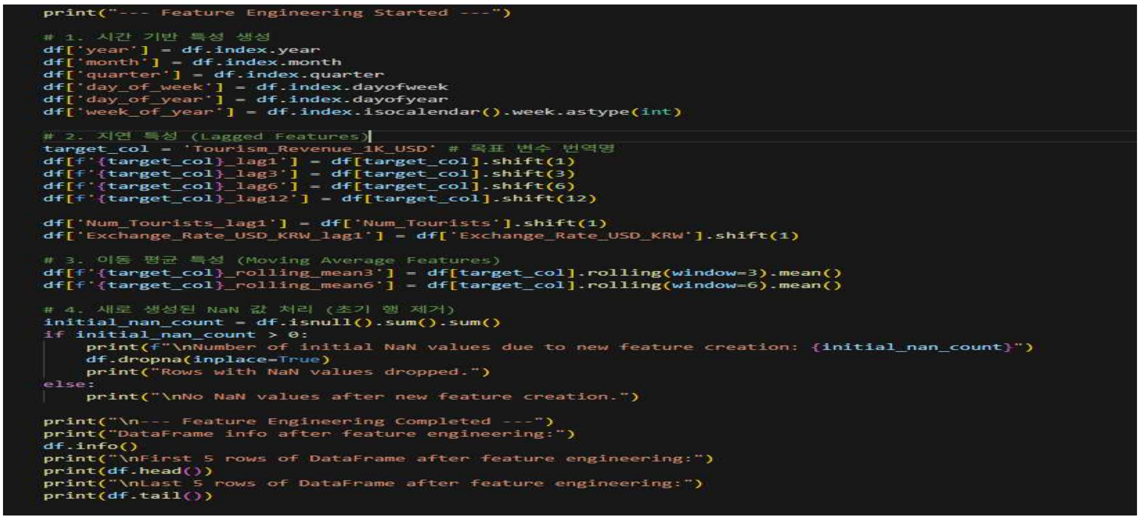
Next, feature engineering was performed to improve the predictive performance of the time-series model.

First, time-based variables such as year, month, quarter, day of the week, and week number were created using the date index.

Then, lag variables were added to reflect past tourism revenue values, along with moving average variables to capture short-term trends.

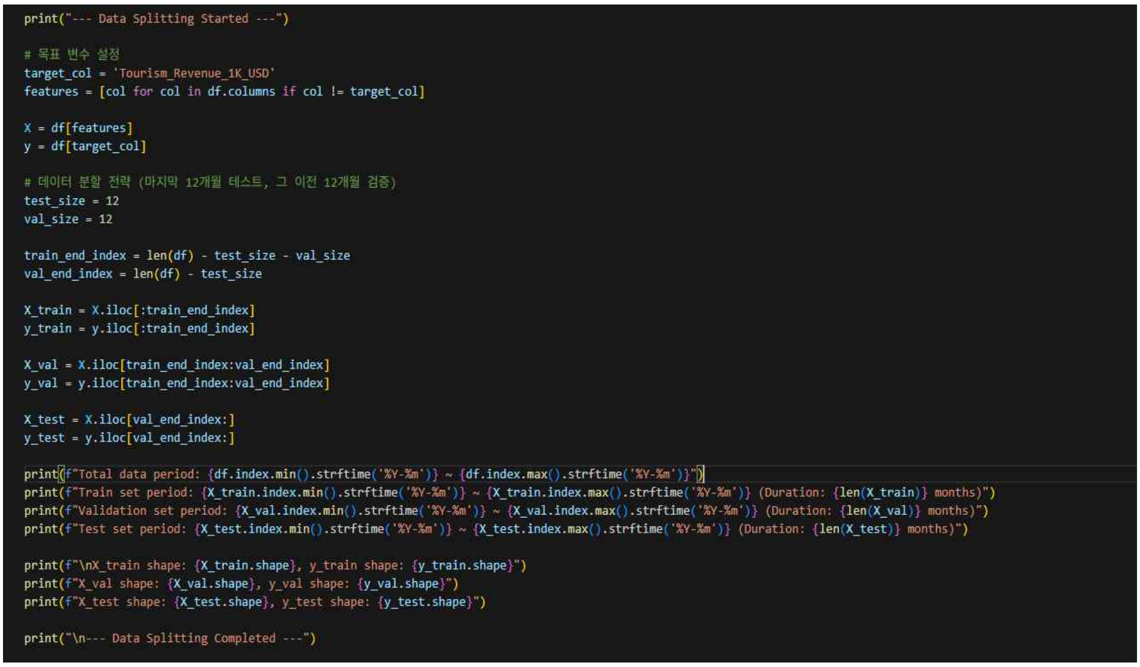
These features help the model better learn seasonality and time-series patterns.

Rows containing initial NaN values caused by these operations were excluded from the training data to ensure accurate predictions.



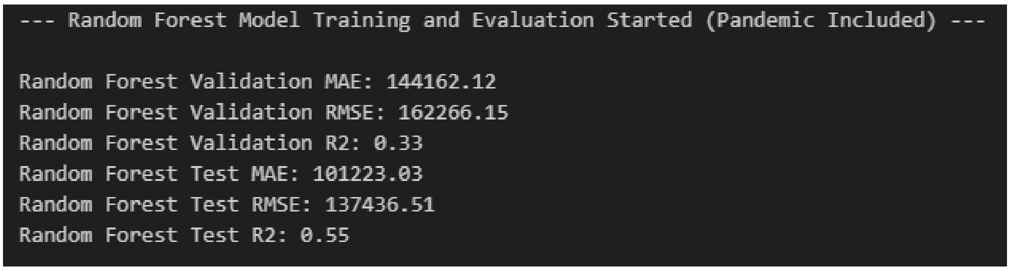
```python
print("--- Feature Engineering Started ---")

# 1. 시간 기반 특성 생성
df['year'] = df.index.year
df['month'] = df.index.month
df['quarter'] = df.index.quarter
df['day_of_week'] = df.index.dayofweek
df['day_of_year'] = df.index.dayofyear
df['week_of_year'] = df.index.isocalendar().week.astype(int)

# 2. 지연 특성 (Lagged Features)
target_col = 'Tourism_Revenue_1K_USD' # 목표 변수 번역명
df[f'{target_col}_lag1'] = df[target_col].shift(1)
df[f'{target_col}_lag3'] = df[target_col].shift(3)
df[f'{target_col}_lag6'] = df[target_col].shift(6)
df[f'{target_col}_lag12'] = df[target_col].shift(12)

df['Num_Tourists_lag1'] = df['Num_Tourists'].shift(1)
df['Exchange_Rate_USD_KRW_lag1'] = df['Exchange_Rate_USD_KRW'].shift(1)

# 3. 이동 평균 특성 (Moving Average Features)
df[f'{target_col}_rolling_mean3'] = df[target_col].rolling(window=3).mean()
df[f'{target_col}_rolling_mean6'] = df[target_col].rolling(window=6).mean()

# 4. 새로 생성된 NaN 값 처리 (초기 행 제거)
initial_nan_count = df.isnull().sum().sum()
if initial_nan_count > 0:
    print(f"\nNumber of initial NaN values due to new feature creation: {initial_nan_count}")
    df.dropna(inplace=True)
    print("Rows with NaN values dropped.")
else:
    print("\nNo NaN values after new feature creation.")

print("\n--- Feature Engineering Completed ---")
print("DataFrame info after feature engineering:")
df.info()
print("\nFirst 5 rows of DataFrame after feature engineering:")
print(df.head())
print("\nLast 5 rows of DataFrame after feature engineering:")
print(df.tail())
```

Fourth, the tourism revenue was set as the dependent variable, while all other variables were treated as independent variables.
The most recent 12 months of data were allocated for testing, and the 12 months preceding that were used for validation.
All remaining earlier data were assigned for training. To prevent data shuffling and maintain temporal integrity, the dataset was split in chronological order.
This ensured a clear separation between training, validation, and testing periods, effectively preventing data leakage.
Finally, the output clearly displayed the duration and size of each dataset, explicitly stating the training, validation, and testing periods to ensure objectivity in model evaluation.

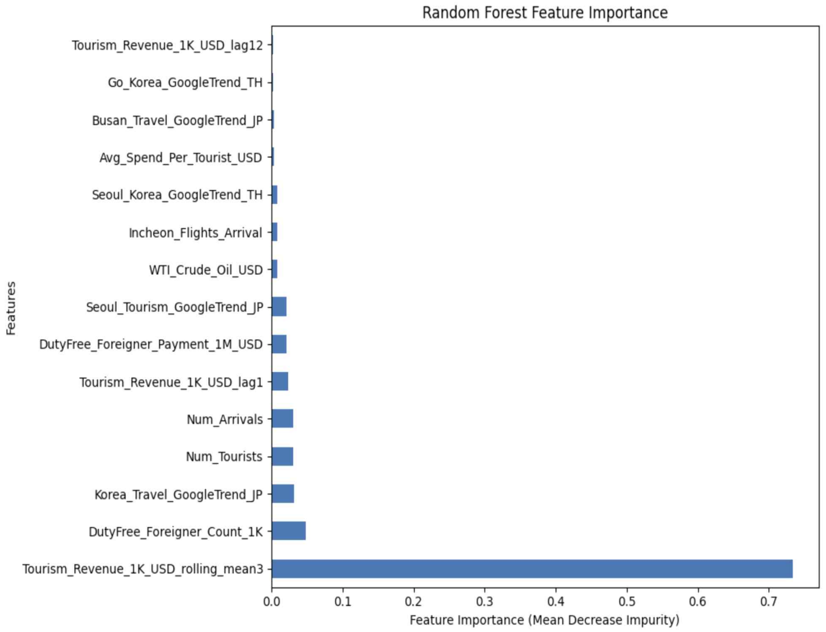


```
print("--- Data Splitting Started ---")

# 목표 변수 설정
target_col = 'Tourism_Revenue_1K_USD'
features = [col for col in df.columns if col != target_col]

X = df[features]
y = df[target_col]

# 데이터 분할 전략 (마지막 12개월 테스트, 그 이전 12개월 검증)
test_size = 12
val_size = 12

train_end_index = len(df) - test_size - val_size
val_end_index = len(df) - test_size

X_train = X.iloc[:train_end_index]
y_train = y.iloc[:train_end_index]

X_val = X.iloc[train_end_index:val_end_index]
y_val = y.iloc[train_end_index:val_end_index]

X_test = X.iloc[val_end_index:]
y_test = y.iloc[val_end_index:]

print(f"Total data period: {df.index.min().strftime('%Y-%m')} ~ {df.index.max().strftime('%Y-%m')}")
print(f"Train set period: {X_train.index.min().strftime('%Y-%m')} ~ {X_train.index.max().strftime('%Y-%m')} (Duration: {len(X_train)} months)")
print(f"Validation set period: {X_val.index.min().strftime('%Y-%m')} ~ {X_val.index.max().strftime('%Y-%m')} (Duration: {len(X_val)} months)")
print(f"Test set period: {X_test.index.min().strftime('%Y-%m')} ~ {X_test.index.max().strftime('%Y-%m')} (Duration: {len(X_test)} months)")

print(f"\nX_train shape: {X_train.shape}, y_train shape: {y_train.shape}")
print(f"X_val shape: {X_val.shape}, y_val shape: {y_val.shape}")
print(f"X_test shape: {X_test.shape}, y_test shape: {y_test.shape}")

print("\n--- Data Splitting Completed ---")
```

First, a Random Forest model was used to predict domestic tourism revenue. Random Forest is an ensemble learning method that builds multiple decision trees and aggregates their results through averaging (for regression) or majority voting (for classification).
The results of the Random Forest model showed relatively low MAE and RMSE values on the test set, along with a higher R² score.
The R² value on the validation set was 0.33, indicating low explanatory power, while the R² on the test set was 0.55, which represents a moderate level of performance for a regression model.
Although lower MAE and RMSE values are desirable, both exceeded 100,000, suggesting that prediction errors are still significant.
Overall, the model demonstrated moderate performance, with some predictive ability on the test set, but limited explanatory power on the validation set, which may point to potential issues with the data.



```
--- Random Forest Model Training and Evaluation Started (Pandemic Included) ---

Random Forest Validation MAE: 144162.12
Random Forest Validation RMSE: 162266.15
Random Forest Validation R2: 0.33
Random Forest Test MAE: 101223.03
Random Forest Test RMSE: 137436.51
Random Forest Test R2: 0.55
```



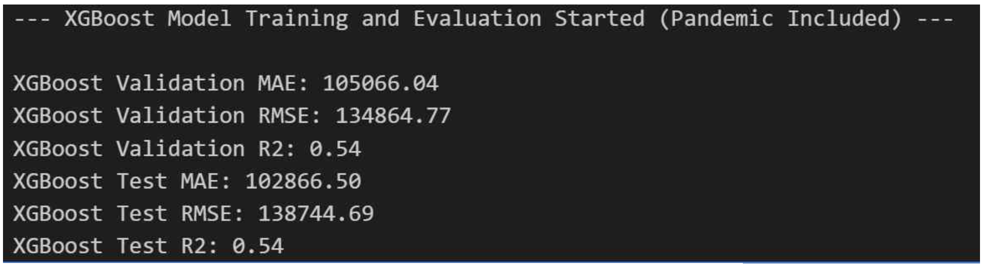
The Random Forest model identified the 3-month moving average of tourism revenue—which reflects short-term trends—as the most important feature by a wide margin. This was followed by the number of foreign visitors to duty-free stores and the level of public interest in Japan. These results suggest that recent revenue trends and indicators representing consumer demand play a key role in predicting tourism revenue.

XGBoost is an efficient implementation of the gradient boosting algorithm, which enhances predictive performance by sequentially learning trees that correct the errors of previous models.

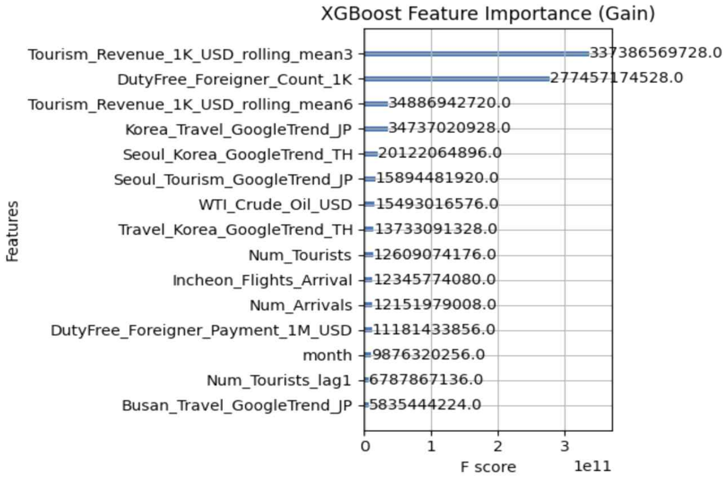
The MAE and RMSE values were similar across both the validation and test sets, and the R² score was consistently 0.54.

This indicates that the data was properly split and the model was trained in a stable and reliable manner.

An R² value of 0.54 means the model explains approximately 54% of the variance in the target variable, representing a reasonably good performance above average.



```
--- XGBoost Model Training and Evaluation Started (Pandemic Included) ---

XGBoost Validation MAE: 105066.04
XGBoost Validation RMSE: 134864.77
XGBoost Validation R2: 0.54
XGBoost Test MAE: 102866.50
XGBoost Test RMSE: 138744.69
XGBoost Test R2: 0.54
```

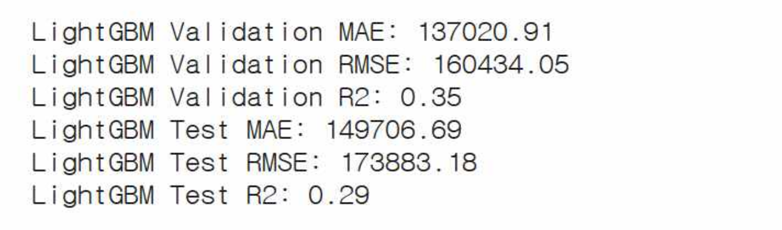


The XGBoost model placed the greatest importance on short- and mid-term trends, while the number of foreign visitors to duty-free stores and overseas interest based on Google Trends also served as key predictive features. In contrast, variables such as regional interest, past tourist arrivals, and monthly seasonality had relatively lower importance.

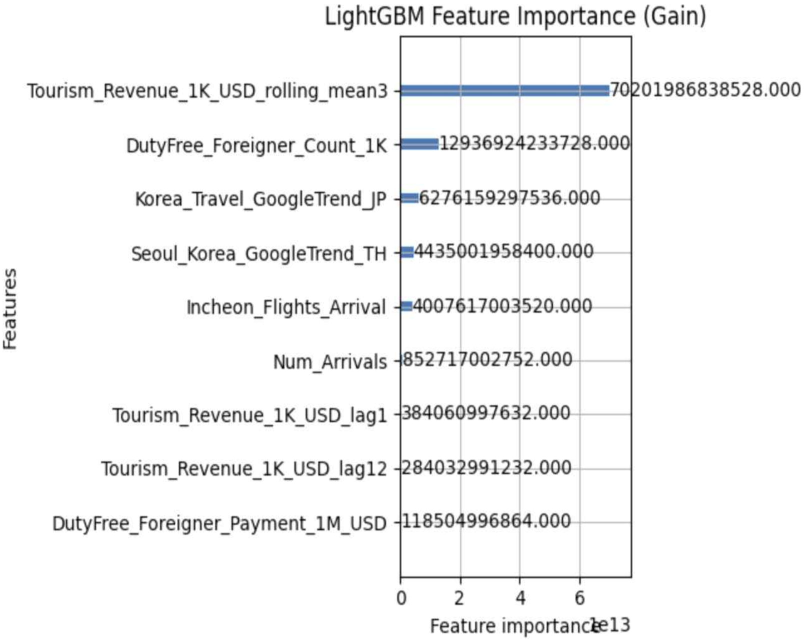
Next, the LightGBM model was tested.

LightGBM, developed by Microsoft, is a gradient boosting framework based on decision trees that is highly efficient for processing large-scale and

high-dimensional data, with fast training speeds.
However, in this case, the model produced larger prediction errors and showed lower accuracy.
The R² scores were 0.35 for the validation set and 0.29 for the test set, indicating weak explanatory power.
These low values suggest that the model was unable to explain even 30% of the variance in the target variable, and its overall predictive performance was significantly lacking.
Although the validation performance was slightly better than the test performance, both were low, implying that the model failed to adequately learn the underlying patterns in the data.



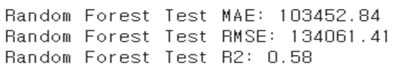
```
LightGBM Validation MAE: 137020.91
LightGBM Validation RMSE: 160434.05
LightGBM Validation R2: 0.35
LightGBM Test MAE: 149706.69
LightGBM Test RMSE: 173883.18
LightGBM Test R2: 0.29
```



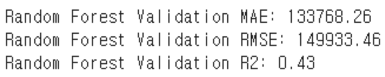
The LightGBM model identified short-term trends in tourism revenue, the number of foreign visitors using duty-free stores, Google Trends-based interest levels, and the number of inbound flights as the most important features for prediction.
This highlights that recent trends, demand indicators, and supply-related factors play a crucial role in forecasting tourism revenue.

Next, the data was split by year—excluding the pandemic period—to allow the model to learn from normal tourism revenue patterns and evaluate its predictive performance on recent years.

Using the Random Forest model first, validation performance improved. Both MAE and RMSE decreased, and the R² score rose from 0.33 to 0.43.
By removing pandemic-period data, the model was able to learn regular patterns more effectively.
Test performance also showed slight improvement: although MAE increased slightly, RMSE decreased, and the R² score increased from 0.55 to 0.58.
Overall, excluding pandemic data contributed to enhancing the model's predictive stability.

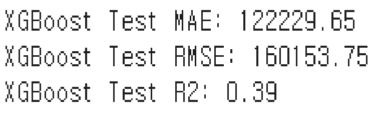


```
Random Forest Test MAE: 103452.84
Random Forest Test RMSE: 134061.41
Random Forest Test R2: 0.58
```

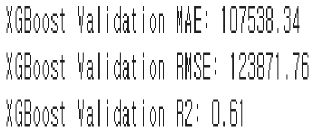


```
Random Forest Validation MAE: 133768.26
Random Forest Validation RMSE: 149933.46
Random Forest Validation R2: 0.43
```

For the XGBoost model, excluding the pandemic data also led to improved validation performance.
The RMSE decreased by approximately 11,000, and the R² score improved from 0.54 to 0.61.
Although the MAE slightly increased, the overall predictive performance became better during validation.
However, performance on the test set declined: both MAE and RMSE increased, and the R² score dropped from 0.54 to 0.39.
This suggests that the test set—representing data from 2024—may reflect a recovery phase following the pandemic, exhibiting patterns different from those in the training data.

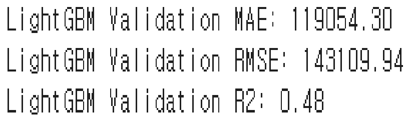


```
XGBoost Test MAE: 122229.65
XGBoost Test RMSE: 160153.75
XGBoost Test R2: 0.39
```

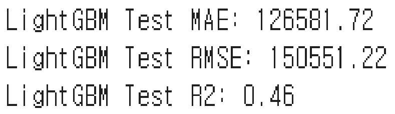


```
XGBoost Validation MAE: 107538.34
XGBoost Validation RMSE: 123871.76
XGBoost Validation R2: 0.61
```

For the LightGBM model, validation performance improved significantly after excluding the pandemic data.
The RMSE decreased by over 17,000, the R² score rose substantially from 0.35 to 0.48, and the MAE decreased by approximately 18,000.
These results indicate that the pandemic data acted as outliers, hindering the model's learning process.
In terms of test performance, the MAE decreased by about 23,000, the R² score improved significantly from 0.29 to 0.46, and the RMSE also dropped by more than 23,000.
Overall, the exclusion of pandemic-period data greatly enhanced the model's ability to make accurate and stable predictions.



```
LightGBM Validation MAE: 119054.30
LightGBM Validation RMSE: 143109.94
LightGBM Validation R2: 0.48
```



```
LightGBM Test MAE: 126581.72
LightGBM Test RMSE: 150551.22
LightGBM Test R2: 0.46
```

When forecasting domestic tourism revenue including the pandemic period, the Random Forest model showed the highest explanatory power on the test set.
Interestingly, its predictions closely matched the actual values during the pandemic, suggesting that the model retained the anomalous patterns from the past and applied them effectively to the test data.
The XGBoost model performed at a similar level to Random Forest, demonstrating balanced performance across both the validation and test sets.
This indicates that the model was able to learn while partially adjusting for past irregularities.
On the other hand, the LightGBM model had the lowest performance among the three. It struggled to effectively capture the volatility of abnormal periods like the pandemic.
In contrast, when excluding pandemic data, XGBoost achieved the best validation performance.
As a boosting-based model, XGBoost effectively captured fine-grained time-series patterns, allowing it to make highly accurate predictions on stable, non-disrupted data.
LightGBM ranked in the middle, showing decent accuracy while benefiting from its lightweight structure and fast learning speed.
Random Forest showed the lowest validation R² score of 0.43, indicating a weaker ability to learn from the normal patterns once the pandemic data was removed.
Regarding test performance, Random Forest achieved the highest R² score of 0.58.

Although its validation performance was weaker, it aligned well with the data pattern at the test point.
LightGBM again showed moderate performance with consistent and stable results, suggesting it was not overfitting and could be considered a reliable model.
XGBoost, despite its strong performance in training and validation, had the lowest test performance.
This reveals its limited adaptability to new, unseen patterns that differed from the training data.

Based on the model performance evaluation results, Random Forest is the most suitable model for practical application. It showed the highest predictive power on the test set and was able to stably forecast trends based on past data, indicating the model's superior generalization ability. In addition, it was less sensitive to outliers such as the pandemic and was easy to implement and apply.
A common limitation among all three models is that the learning data was distorted due to the impact of the pandemic. When the pandemic was included, the R^2 score dropped sharply, and the learned trends failed to reflect reality. There was also an imbalance between validation and test performance. In the case of XGBoost, the difference in R^2 between the validation and test sets was large when the pandemic was excluded. Furthermore, all models had a non-time-series structure and thus could not recognize the temporal order. The use of external variables was also insufficient. While some exogenous variables such as exchange rates, oil prices, and duty-free sales were included, there was no data on events, policy announcements, or diplomatic issues, which have a significant impact on tourism revenue. As a result, the models had difficulty explaining "why" changes occurred and missed important triggers.
In particular, XGBoost and LightGBM lacked model interpretability. Although they had high R^2 scores, it was difficult to explain the reasons behind the predictions.
As for improvements, the pandemic period should be separated and trained as a distinct scenario. A main model based on normal data excluding the pandemic would result in higher predictive accuracy. To address the imbalance between validation and test performance, it is necessary to apply resampling or re-splitting strategies to reduce distributional differences among training, validation, and test sets.
In addition, derivative variables such as month, quarter, year, holiday, and season should be generated to reflect the time-series characteristics in the model.
It is also important to incorporate schedule data provided by agencies such as the Korea Tourism Organization, Ministry of Foreign Affairs, and Ministry of Culture, Sports and Tourism. By including major national holidays, diplomatic news related to Korea, etc., the accuracy of tourism revenue forecasts can be further improved.
In this study, various external variables and time-series data were integrated to accurately predict domestic tourism revenue, and the Random Forest, XGBoost, and LightGBM models were compared and analyzed.
The prediction model can be used to allocate tourism promotion funds and local government budgets based on demand forecasting. It also enables flexible seasonal budget operations and optimization of marketing costs.
The forecasted revenue trends serve as key indicators for business planning and investment feasibility analysis in private sectors such as hotels, airlines, and duty-free shops, and can also be used as reference data for profitability review when developing new tourist destinations or operating experience-based content.
In addition, marketing strategies such as K-content promotion and visa relaxation policies can be designed for countries with increasing foreign tourist inflow.
In particular, event and seasonal marketing strategies can be developed. Based on monthly revenue forecasts, strategies such as seasonal product planning, festival organization, and flight promotions can be established, which gives the model great practical significance.

Data Sources

https://data.kma.go.kr/stcs/grnd/grndTaList.do

https://datalab.visitkorea.or.kr/datalab/portal/main/getMainForm.do

https://www.airport.kr/co_ko/651/subview.do

https://fred.stlouisfed.org/series/DEXKOUS

https://know.tour.go.kr/stat/tourStatSearchDis19Re.do;jsessionid=231763ADE8F7B87A14C675428F508384#

https://data.seoul.go.kr/dataList/264/S/2/datasetView.do

https://www.data.go.kr/data/15142084/fileData.do?recommendDataYn=Y

https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&ssc=tab.nx.all&query=2025%EB%85%84+%EA%B3%B5%ED%9C%B4%EC%9D%BC&oquery=2024%EB%85%84+%EA%B3%B5%ED%9C%B4%EC%9D%BC&tqi=juwE2lqosesssShULqGssssthR-216135&ackey=ktdz57uj

https://www.sedaily.com/NewsView/29THBUEY1Z