FINAL PROJECT REPORT - DATA SCIENCE

PREDICTING TOURIST ARRIVALS

PHẠM CÔNG SỰ

Table of contents

I

Exploratory Data Analysis

Business Understanding

Project problem:

- Tourism is a crucial economic sector that significantly contributes to the economies of numerous countries. However, tourist arrivals are dynamic and susceptible to the influence of various factors. Accurate tourist demand forecasting empowers tourism and hospitality businesses to minimize unnecessary costs and maximize business opportunities..

- This project, inspired by a Kaggle competition, aims to develop a tourist arrival prediction model. The goal is to provide precise insights into Japan's future tourist demand, enabling businesses and relevant organizations to make informed business and development decisions, ultimately contributing to the industry's growth.

- Details: https://www.kaggle.com/competitions/prediction-of-tourist-arrivals/overview

Business Understanding

Objective:

- Predicting tourist arrivals for July 2019.

Factors influencing tourist arrivals:

- Seasonality
- Location
- Destination characteristics: urban centers, rural areas, natural attractions...
- Special events, holidays
- Weather
- Other factors

Business Understanding

Evaluation Criteria:

- The project will be evaluated based on: RMSE
- RMSE (Root Mean Square Error): the primary metric used in the Kaggle competition to assess the accuracy of prediction models
- Interpretation of RMSE:
  - A lower RMSE value indicates higher predictive accuracy of the model, meaning the model's predictions are closer to the actual values.
  - A higher RMSE value indicates lower predictive accuracy of the model, meaning the model's predictions deviate further from the actual values.
- The project will be considered successful if the achieved RMSE is less than or equal to the top 20 results on the Kaggle competition leaderboard at the time of project commencement (RMSE <= 78.33)

Data Understanding

Training data (train_df.csv):

- This dataset comprises 132,192 data points and includes 12 explanatory variables and 1 target variable.

	Field	Definition
1	id	Unique identifier for the record
2	date	Year, month, day
3	tourist_area	
4	spot_facility	
5	area	
6	city	
7	type	
8	category	
9	tourism_index	
10	info	
11	event	
12	weather_index	
13	tourist_arrivals	Number of tourist arrivals (target variable)

Data Understanding

Testing data (test_df.csv):

- This dataset comprises 13,392 data points and includes the same variables as the training data except for the target variable, tourist_arrivals.

	Fields	Definition
1	id	Unique identifier for the record
2	date	Year, month, day
3	tourist_area	
4	spot_facility	
5	area	
6	city	
7	type	
8	category	
9	tourism_index	
10	info	
11	event	
12	weather_index	

Data Analysis

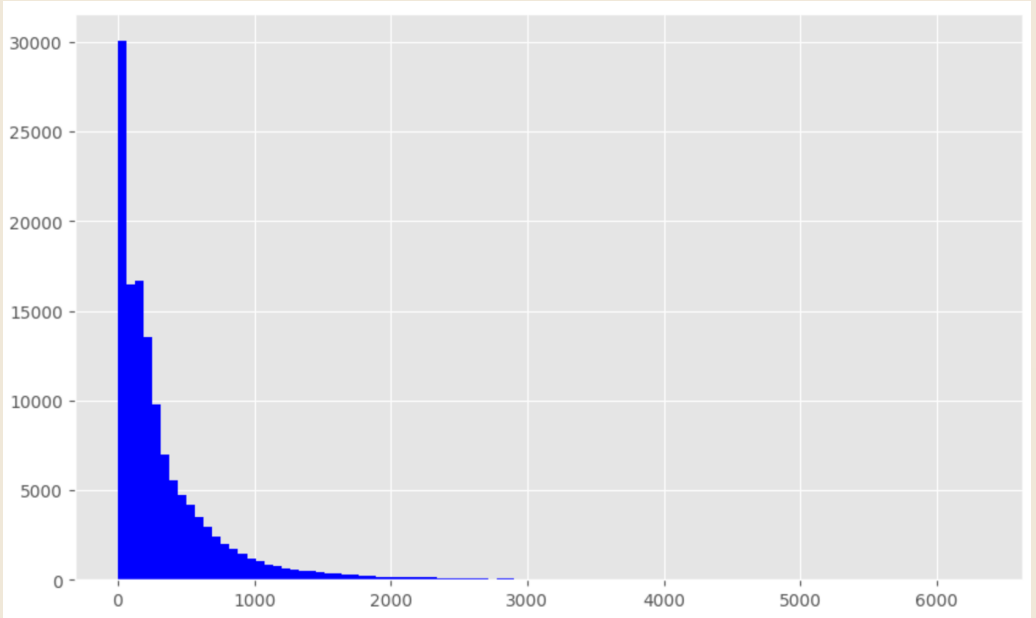
Training data description (train_df.csv):

	dtypes	missing#	missing%	uniques	count
id	int64	0	0.000000	132192	132192
date	datetime64[ns]	0	0.000000	303	132192
tourist_area	int64	0	0.000000	54	132192
spot_facility	object	0	0.000000	8	132192
tourist_arrivals	int64	0	0.000000	3071	132192
area	object	0	0.000000	22	132192
city	object	0	0.000000	16	132192
type	object	0	0.000000	5	132192
category	int64	0	0.000000	17	132192
tourism_index	float64	3992	0.030198	3609	128200
info	object	0	0.000000	4	132192
event	object	0	0.000000	7	132192
weather_index	float64	41040	0.310458	190	91152

Data Analysis

Training data description (train_df.csv):

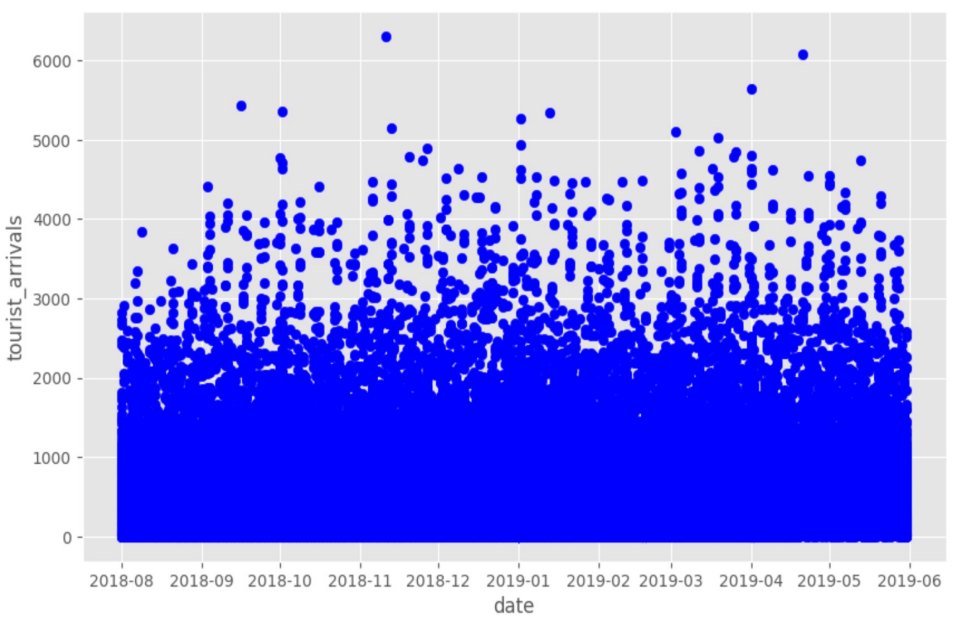
- tourist_arrivals (output)
- Observation: The majority of the values in the dataset lie within the range of 0 to 1000.



Data Analysis

Training data description (train_df.csv):

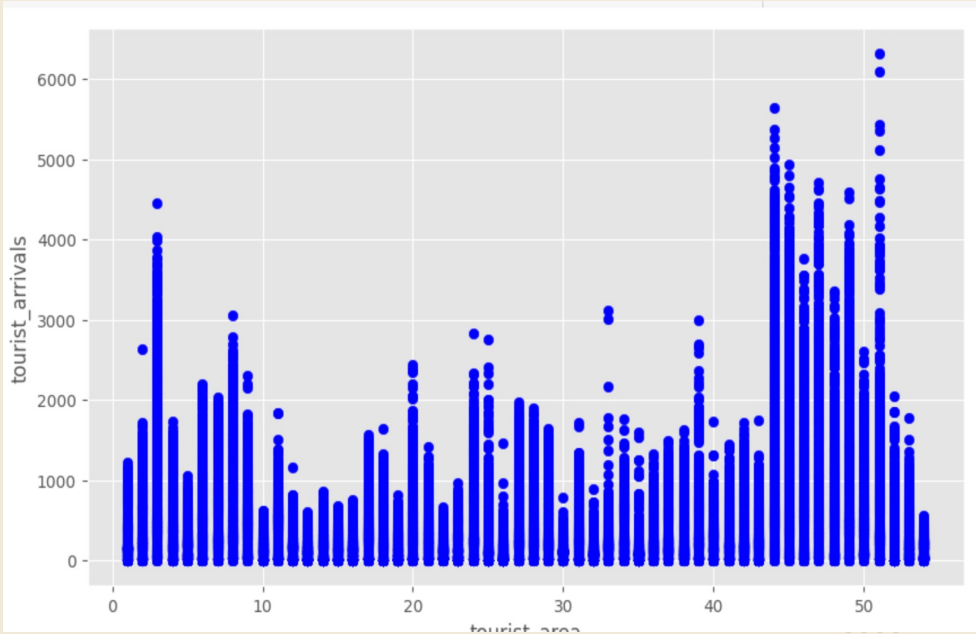
- date
- Observation:
  - Majority of days have tourist arrivals below 3000.
  - Highest number of tourist arrivals is around 6200.
  - tourist_arrivals may depend on date



Data Analysis

Training data description (train_df.csv):

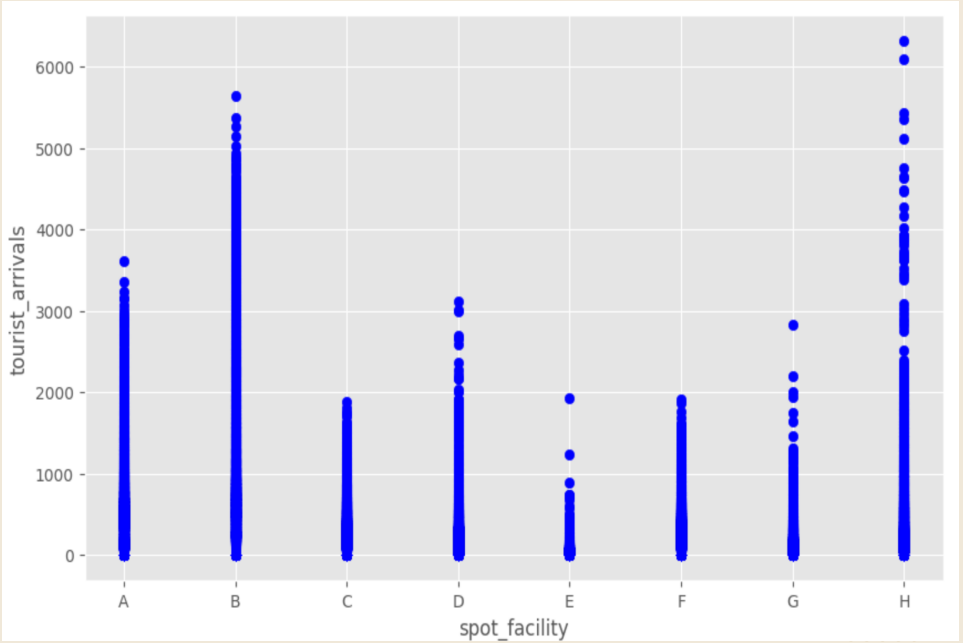
- tourist_area
- Observation:
  - Total of 54 different tourist_area categories
  - Tourists tend to visit tourist_area between 44 and 54
  - tourist_arrivals may depend on tourist_area



Data Analysis

Training data description (train_df.csv):

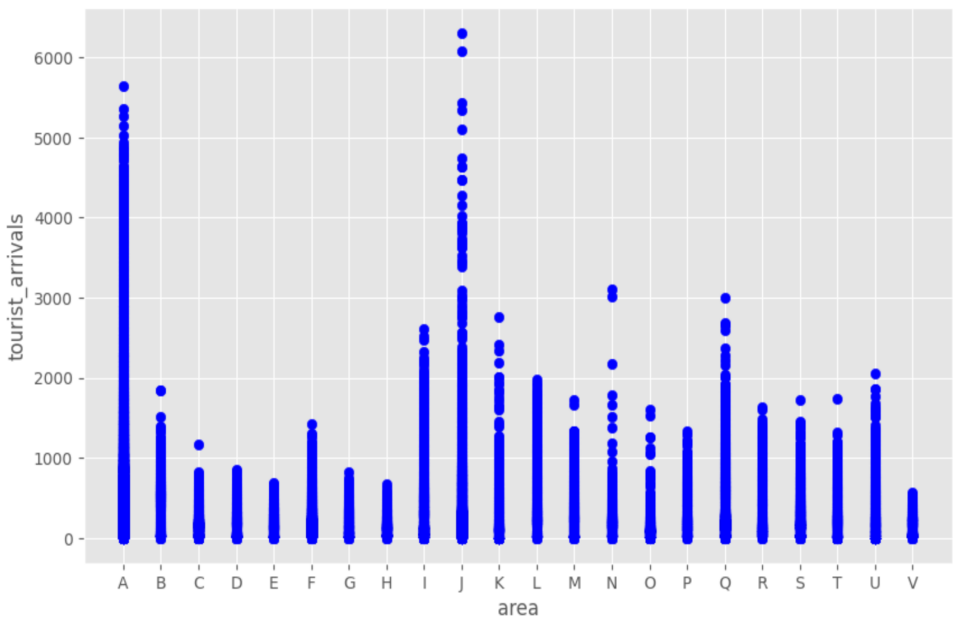
- spot_facility
- Observation:
  - Total of 8 different spot_facility categories (A, B, C, D, E, F, G, H)
  - High tourist_arrivals at spot_facility B and H, low at spot_facility E
  - tourist_arrivals may depend on spot_facility



Data Analysis

Training data description (train_df.csv):

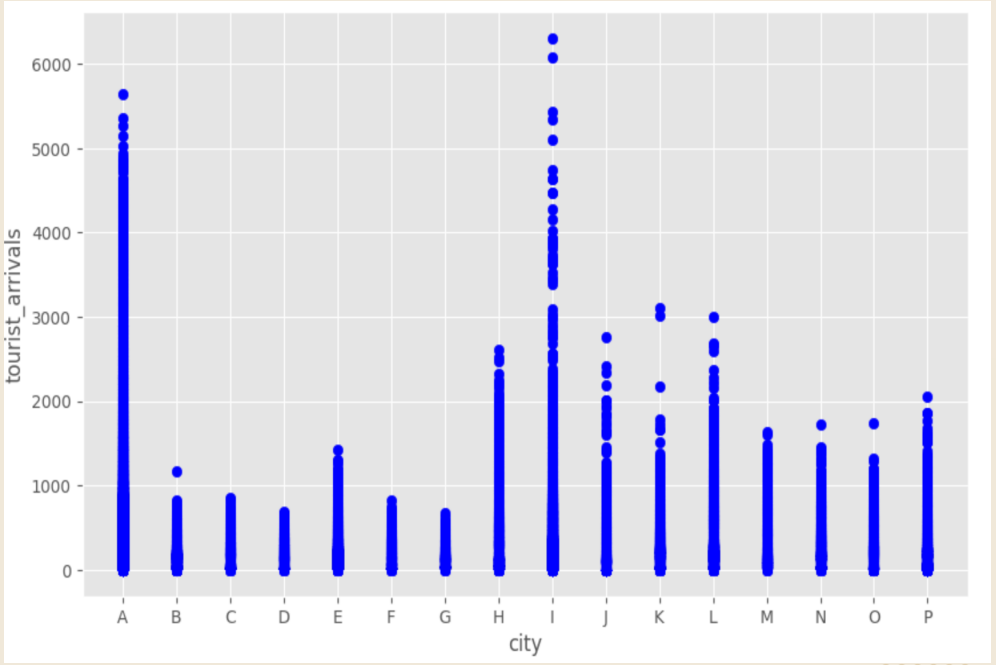
- area
- Observation:
  - Total of 22 different area categories
  - High tourist_arrivals in area A and J, low in C, D, E, G, H, V
  - tourist_arrivals may depend on area



Data Analysis

Training data description (train_df.csv):

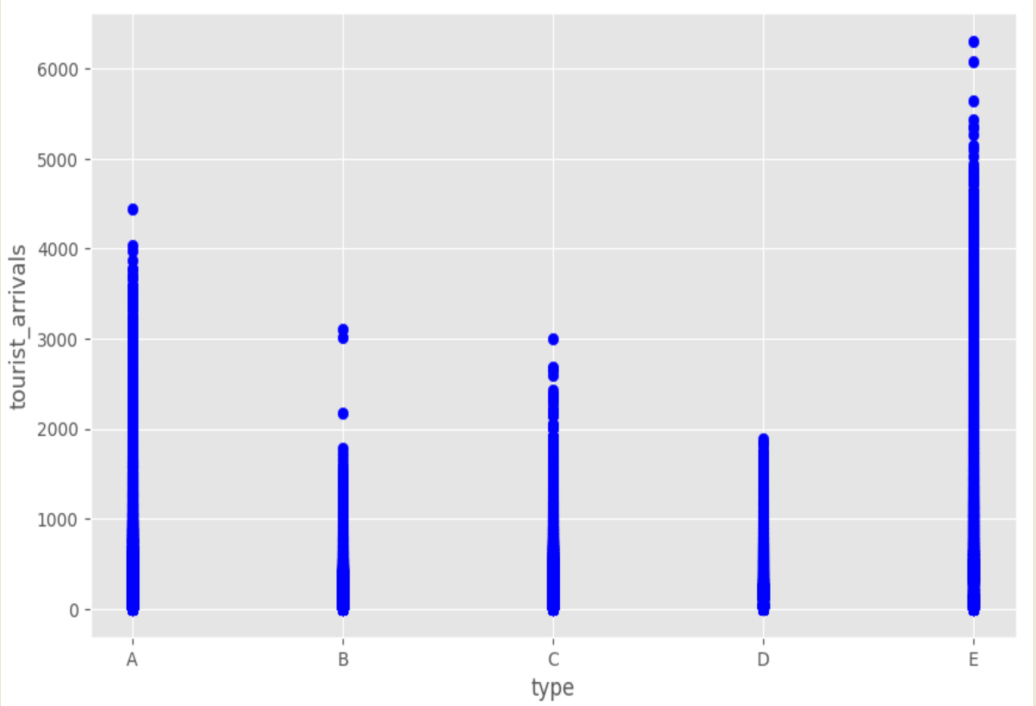
- city
- Observation:
  - Total of 16 different city categories
  - High tourist_arrivals in city A and I, low in B, C, D, F, G
  - tourist_arrivals may depend on city



Data Analysis

Training data description (train_df.csv):

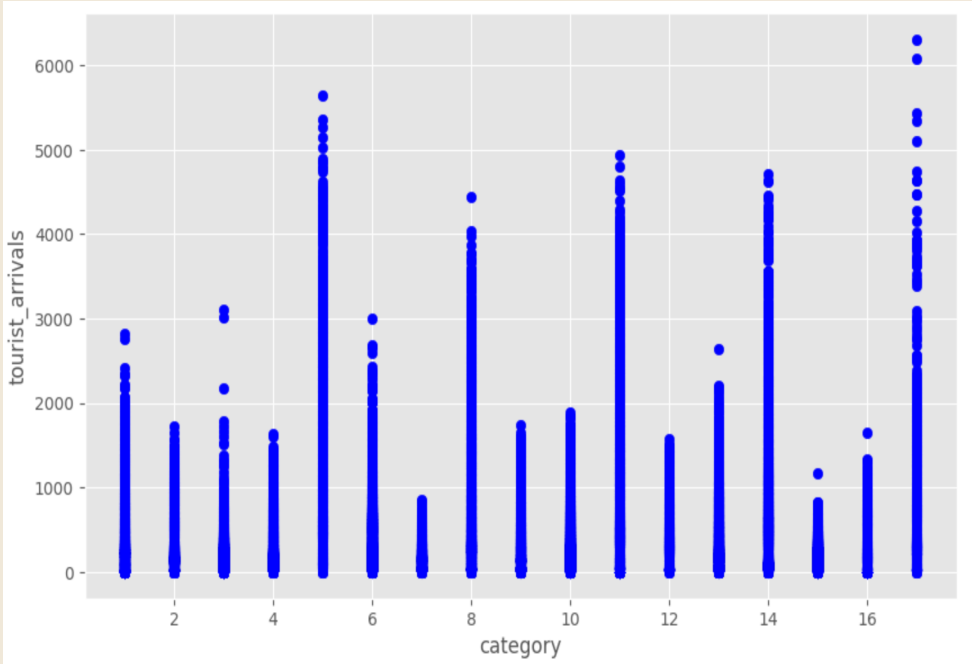
- type
- Observation:
  - Total of 5 different type categories
  - High tourist_arrivals in type E, low in type D
  - tourist_arrivals may depend on type



Data Analysis

Training data description (train_df.csv):

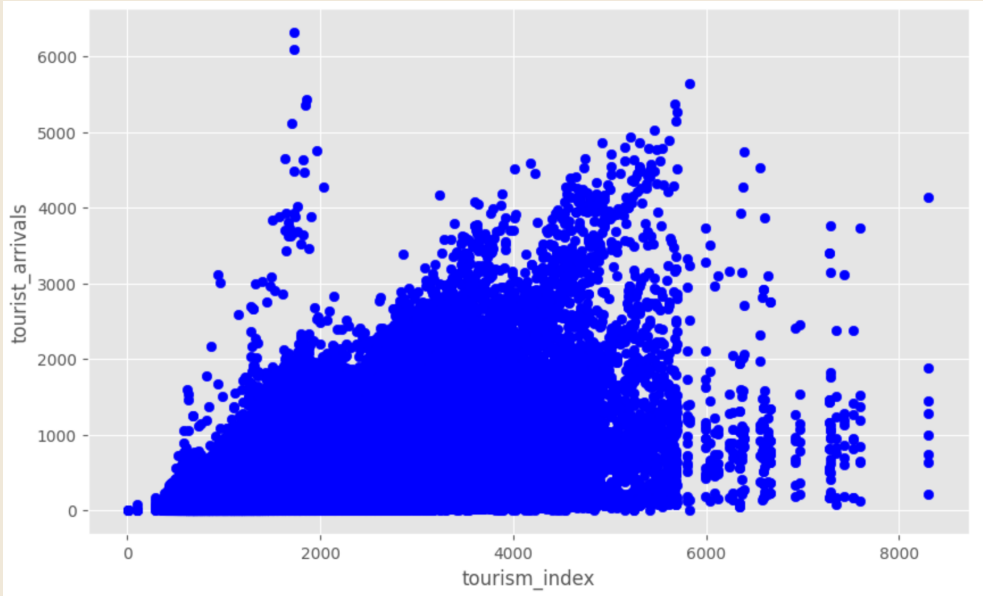
- category
- Observation:
  - Total of 17 different category categories
  - High tourist_arrivals in category 3, 11, 14, 17, low in category 7, 15
  - tourist_arrivals may depend on category



Data Analysis

Training data description (train_df.csv):

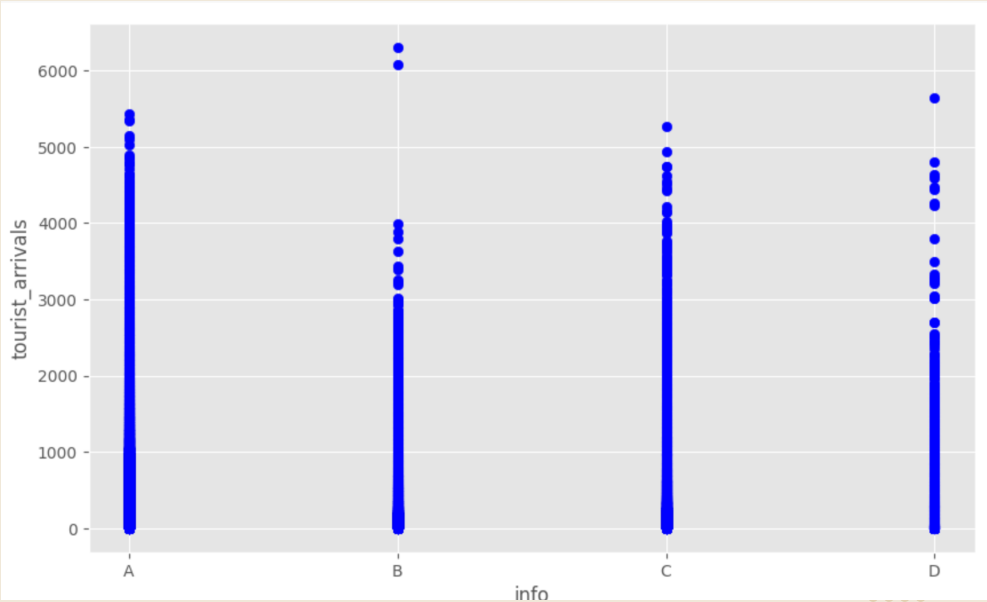
- tourism_index
- Observation:
  - Concentration of tourist_arrivals at tourism_index below 6000
  - tourist_arrivals might be influenced by tourism_index



Data Analysis

Training data description (train_df.csv):

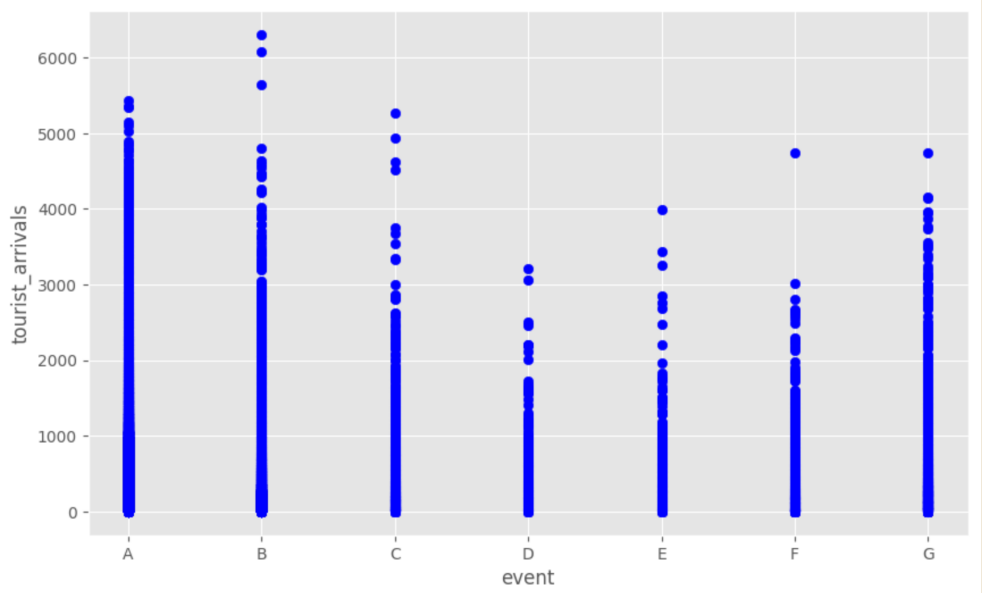
- info
- Observation:
  - Total 4 different info categories
  - High tourist_arrivals in info A and C, low in B and D
  - tourist_arrivals may depend on info



Data Analysis

Training data description (train_df.csv):

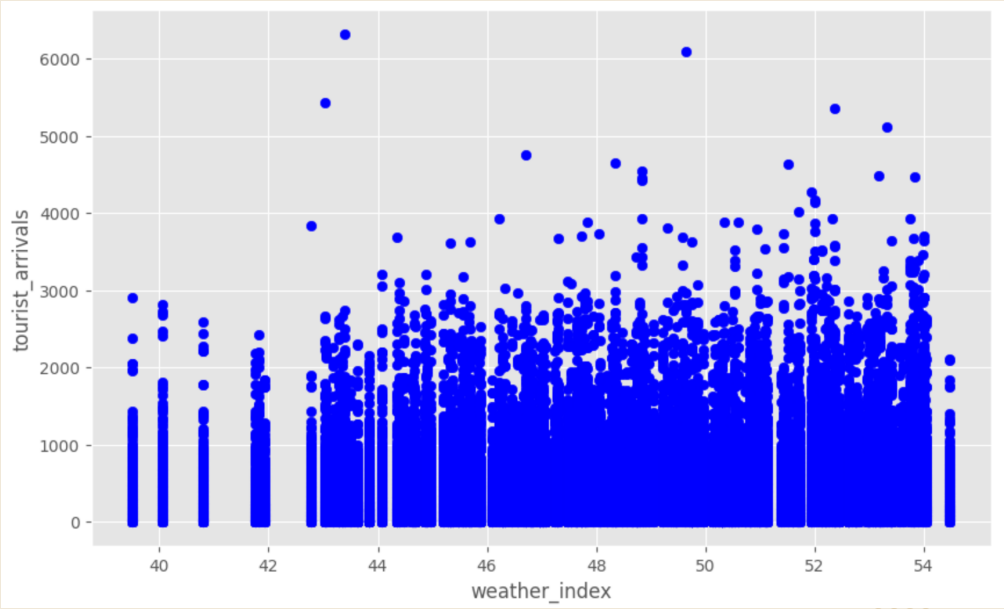
- event
- Observation:
  - Total of 7 different event categories
  - High tourist_arrivals in event A and B, low in D
  - tourist_arrivals may depend on event



Data Analysis

Training data description (train_df.csv):

- weather_index
- Observation:
  - Concentration of tourist_arrivals around weather_index 43 to 54
  - tourist_arrivals may depend on weather_index



Data Analysis

Testing data description (test_df.csv):

	dtypes	missing#	missing%	uniques	count
id	int64	0	0.000000	13392	13392
date	datetime64[ns]	0	0.000000	31	13392
tourist_area	int64	0	0.000000	54	13392
spot_facility	object	0	0.000000	8	13392
area	object	0	0.000000	22	13392
city	object	0	0.000000	16	13392
type	object	0	0.000000	5	13392
category	int64	0	0.000000	17	13392
tourism_index	int64	0	0.000000	1216	13392
info	object	0	0.000000	2	13392
event	object	0	0.000000	3	13392
weather_index	float64	5184	0.387097	19	8208

Data Analysis

Data transformation:

- Extract additional features “year”, “month”, “day” from the original “date” feature.
- Employ Label Encoding to convert categorical data in features: 'spot_facility', 'area', 'city', 'type', 'info', 'event’ into numerical data.

Data Analysis

Data transformation:

train_df

	dtypes	missing#	missing%	uniques	count
id	int64	0	0.000000	132192	132192
date	datetime64[ns]	0	0.000000	303	132192
tourist_area	int64	0	0.000000	54	132192
spot_facility	int64	0	0.000000	8	132192
tourist_arrivals	int64	0	0.000000	3071	132192
area	int64	0	0.000000	22	132192
city	int64	0	0.000000	16	132192
type	int64	0	0.000000	5	132192
category	int64	0	0.000000	17	132192
tourism_index	float64	3992	0.030198	3609	128200
info	int64	0	0.000000	4	132192
event	int64	0	0.000000	7	132192
weather_index	float64	41040	0.310458	190	91152
year	int64	0	0.000000	2	132192
month	int64	0	0.000000	10	132192
day	int64	0	0.000000	31	132192

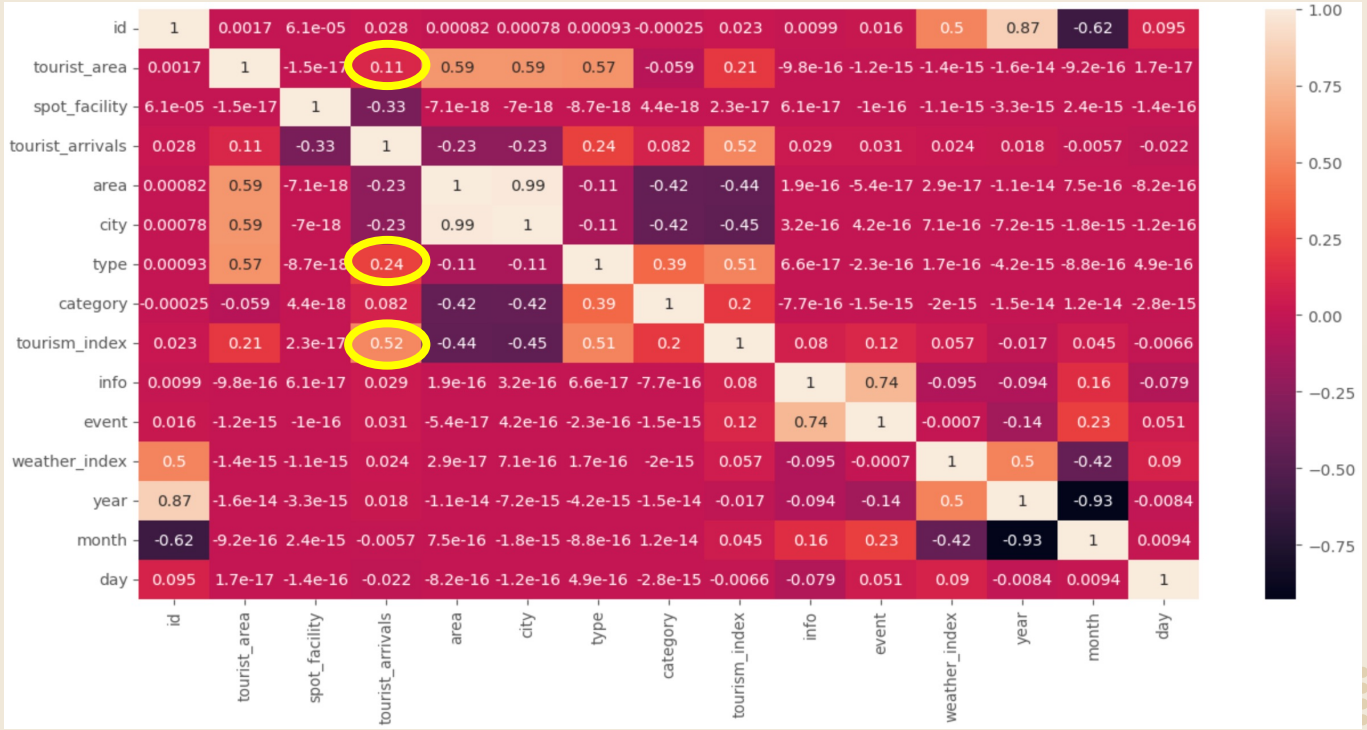
test_df

	dtypes	missing#	missing%	uniques	count
id	int64	0	0.000000	13392	13392
date	datetime64[ns]	0	0.000000	31	13392
tourist_area	int64	0	0.000000	54	13392
spot_facility	int64	0	0.000000	8	13392
area	int64	0	0.000000	22	13392
city	int64	0	0.000000	16	13392
type	int64	0	0.000000	5	13392
category	int64	0	0.000000	17	13392
tourism_index	int64	0	0.000000	1216	13392
info	int64	0	0.000000	2	13392
event	int64	0	0.000000	3	13392
weather_index	float64	5184	0.387097	19	8208
year	int64	0	0.000000	1	13392
month	int64	0	0.000000	1	13392
day	int64	0	0.000000	31	13392

Data Analysis

Correlations between features:

- The features with the highest correlations with "tourist_arrivals" (output) are:
  - tourism_index
  - type
  - tourist_area



Data Analysis

Feature selection:

- Removing “weather_index” feature: since over 30% of the values in both the training and testing datasets are missing
- Removing “id”, “date” features
- Eliminate rows with missing values in “tourism_index”, keeping this field because the percentage of missing values is low and has a high correlation coefficient with “tourist_arivals”

	Feature	Correlation (absolute value)
1	id	0.027750
2	tourist_area	0.112900
3	spot_facility	0.325611
4	area	0.227062
5	city	0.229929
6	type	0.244973
7	category	0.082123
8	tourism_index	0.519752
9	info	0.029452
10	event	0.030675
11	weather_index	0.024104
12	year	0.017656
13	month	0.005725
14	day	0.022258

Problem Solving Method

Algorithm:

- Employ Gradient Boosting, an ensemble machine learning algorithm that combines multiple weak learners to produce a more robust and predictive model.
- Evaluate the performance of Gradient Boosting using three popular frameworks:
  - LightGBM (https://lightgbm.readthedocs.io/en/stable/)
  - XGBoost (https://xgboost.readthedocs.io/en/stable/)
  - CatBoost (https://catboost.ai/)

Problem Solving Method

Technical stack:

- Programming language: Python
- Library/Package/Framework:
  - Data Preprocessing and Manipulation: pandas, numpy, datetime, jpholiday
  - Statistics and Data Analysis: numpy, sklearn
  - Machine Learning: sklearn, keras, optuna, lightgbm, xgboost, catboost
  - Data Visualization: matplotlib, seaborn

II

Model Building and Evaluation

Parameter Setting

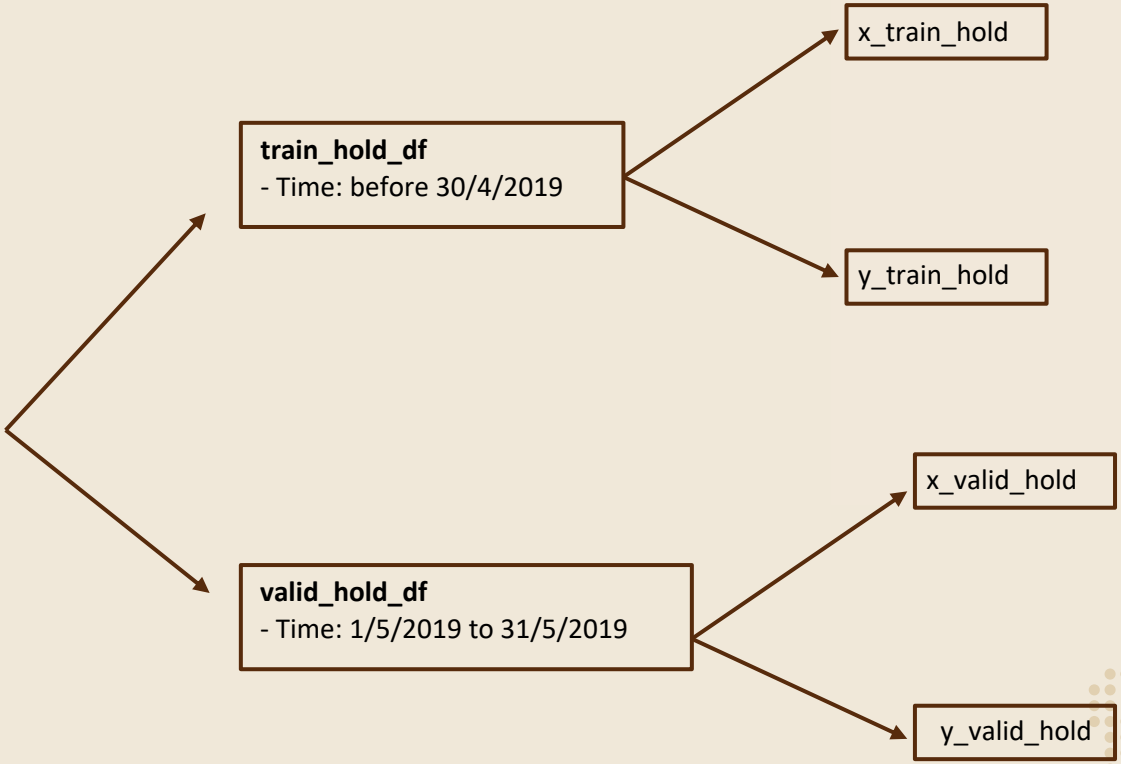
- Parameters to be tuned and adjusted:

Definition	Parameter Name		
	LightGBM	XGBoost	CatBoost
Controlling the proportion of training data used to build each individual tree in the ensemble model	bagging_fraction	subsample	subsample
Learning rate of the model	learning_rate	learning_rate	learning_rate
Number of trees	num_iterations	n_estimators	iterations
Maximum depth of trees	max_depth	max_depth	depth

Training and Evaluation Data Split

Feature	Output
tourist_area	tourist_arrivals
spot_facility	
area	
city	
type	
category	
tourism_index	
info	
event	
year	
month	
day	

train_df



Model Evaluation

Evaluation methods:

- The model should not overfit the training data
- The model with the lowest RMSE on the validation set is considered the best model
- A model is considered satisfactory if its RMSE on the validation set is less than or equal to the evaluation criterion of 78.33 (RMSE <= 78.33)

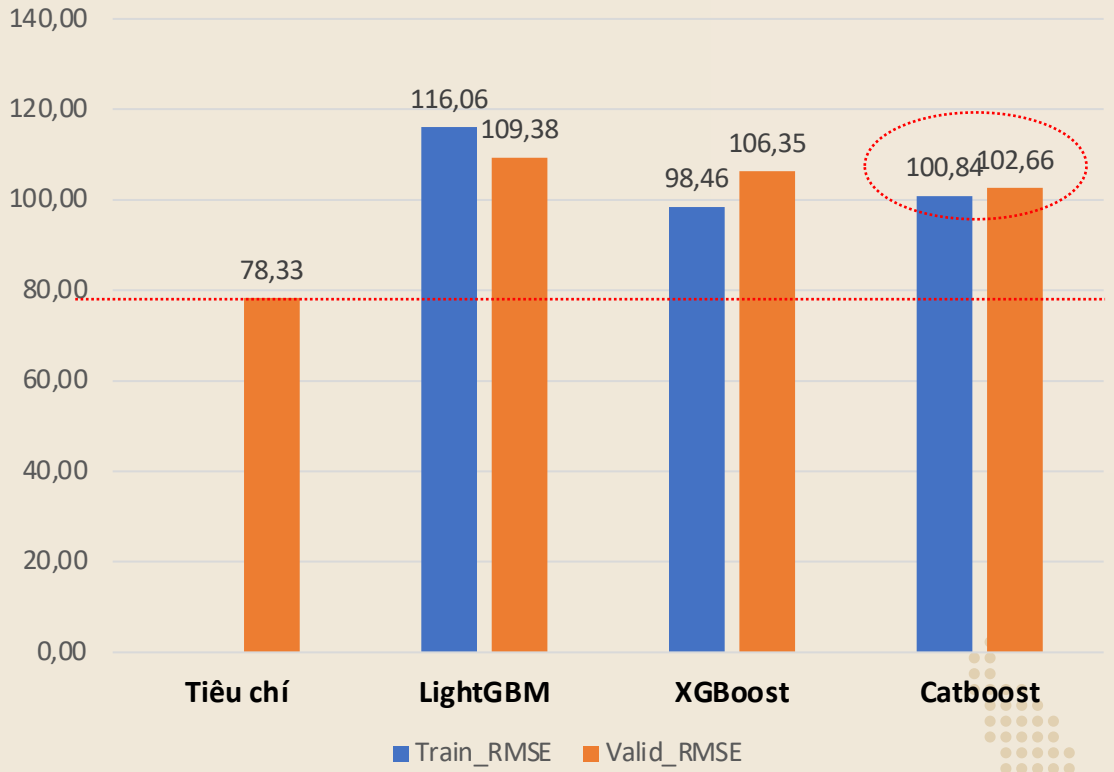
Training models:

- Train LightGBM, XGBoost, and CatBoost models using their default parameter values:
  - subsample
  - learning_rate
  - num_iterations
  - max_depth
- early_stopping_rounds = 100

Model Evaluation

Evaluation results:

- The models did not meet the evaluation criterion of RMSE <= 78.33
- The models did not exhibit significant overfitting on the training data
- CatBoost: Achieved the best performance on the validation set with an RMSE of 102.66
- LightGBM: Achieved the worst performance on the validation set with an RMSE of 109.38



III

Model Improvement

Feature Engineering

Create additional features for the target variable:

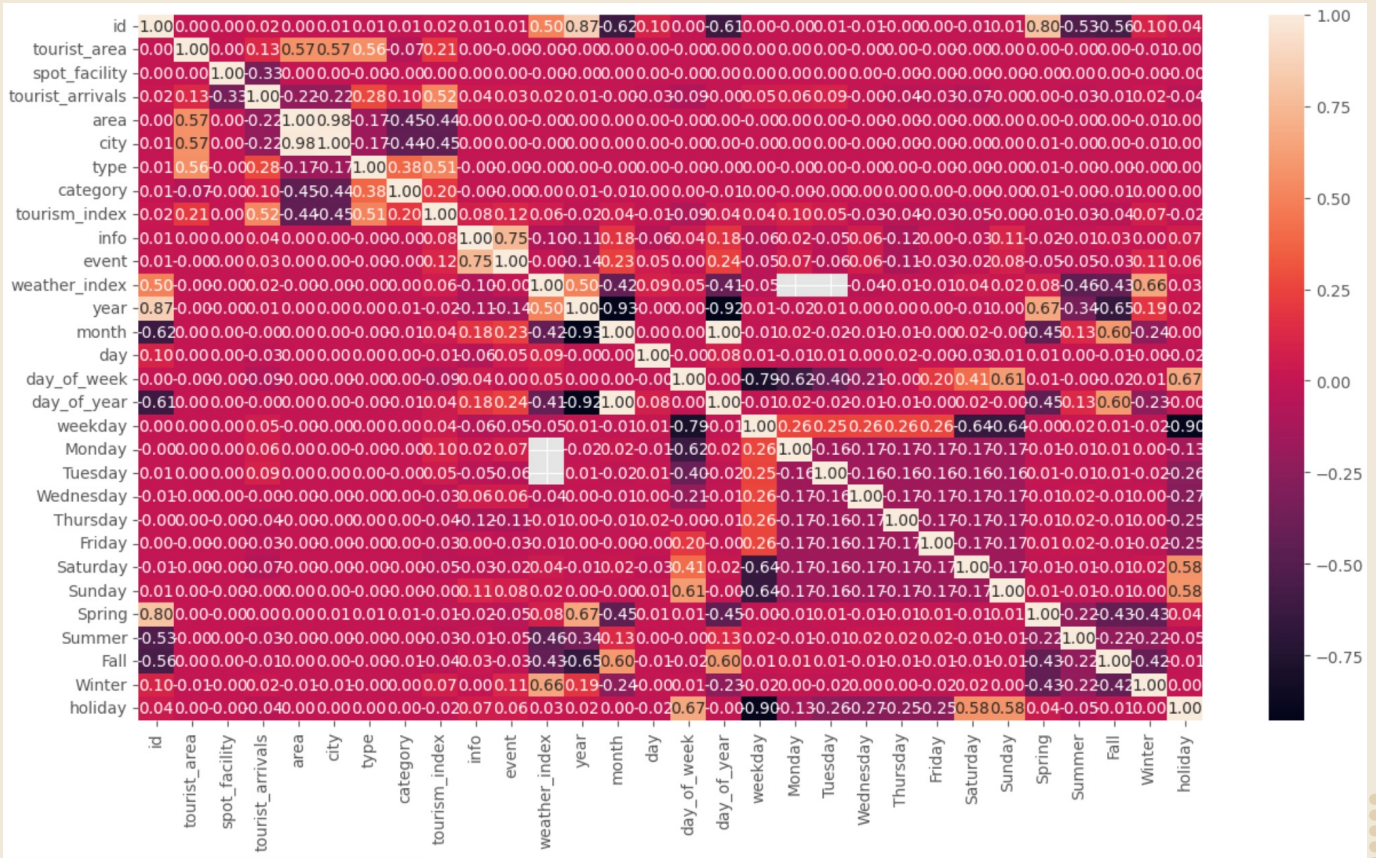
- Construct new features from existing ones to enhance the interpretability of the target variable:

STT	Feature	Definition
1	day_of_week	Day of the week
2	day_of_year	Day of the year
3	Weekday	Is it a weekday?
4	Monday	Is it Monday?
5	Tuesday	Is it Tuesday?
6	Wednesday	Is it Wednesday?
7	Thursday	Is it Thursday?
8	Friday	Is it Friday?
9	Saturday	Is it Saturday?
10	Sunday	Is it Sunday?
11	Spring	Is it spring? (March, April, May)
12	Summer	Is it summer? (June, July, August)
13	Fall	Is it fall? (September, October, November)
14	Winter	Is it winter? (December, January, February)
15	Holiday	Is it a holiday? (Japanese holidays, Saturday, Sunday)

Feature Engineering

Feature selection

- 15 new features were selected based on their strong correlation with the original features
- The final feature set consists of the 12 original features and the 15 newly selected features



Feature Engineering

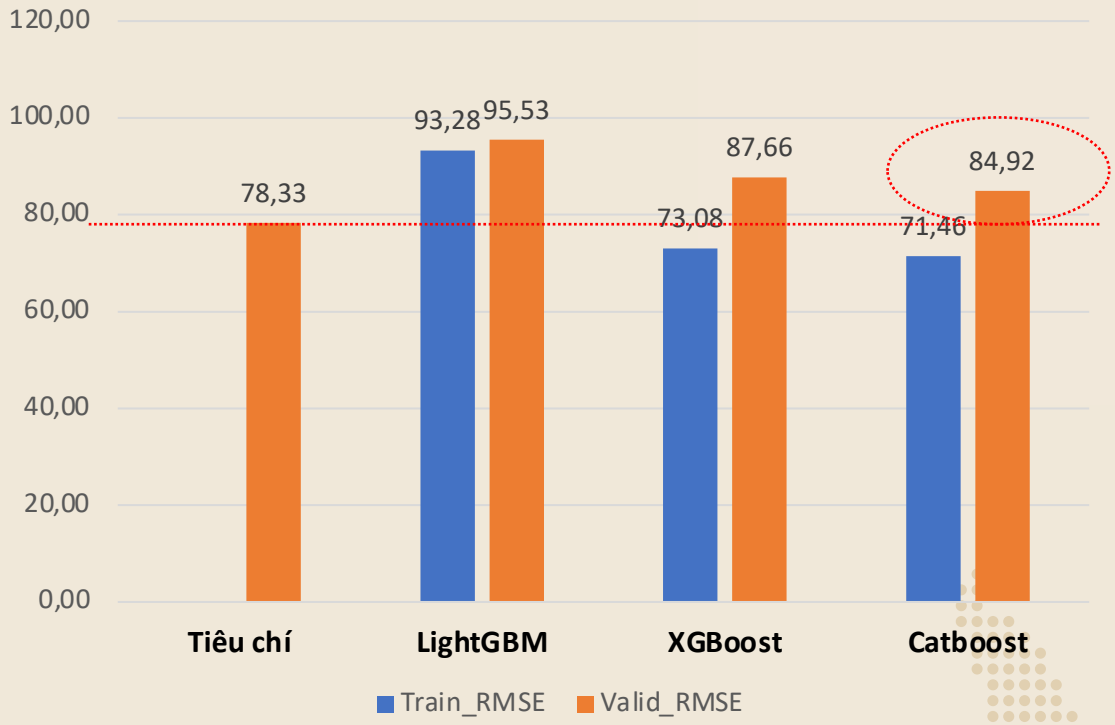
Training models:

- Retrain the LightGBM, XGBoost và CatBoost models using their default parameter values:
  - subsample
  - learning_rate
  - num_iterations
  - max_depth
- early_stopping_rounds = 100

Feature Engineering

Evaluation results

- None of the retrained models met the evaluation criterion of RMSE <= 78.33
- However, all models exhibited performance improvements compared to the initial models:
  - CatBoost: Achieved the best performance on the validation set with an RMSE of 84.92 (decrease 17.74)
  - LightGBM: Achieved the worst performance on the validation set with an RMSE of 95.53 (decrease 13.85)



Parameter Tuning

LightGBM	tuning_1	tuning_2	tuning_optuna
subsample	0,5	0,9	0,894
learning_rate	0,05	0,2	0,085
num_iterations	300	200	4778
max_depth	10	5	45

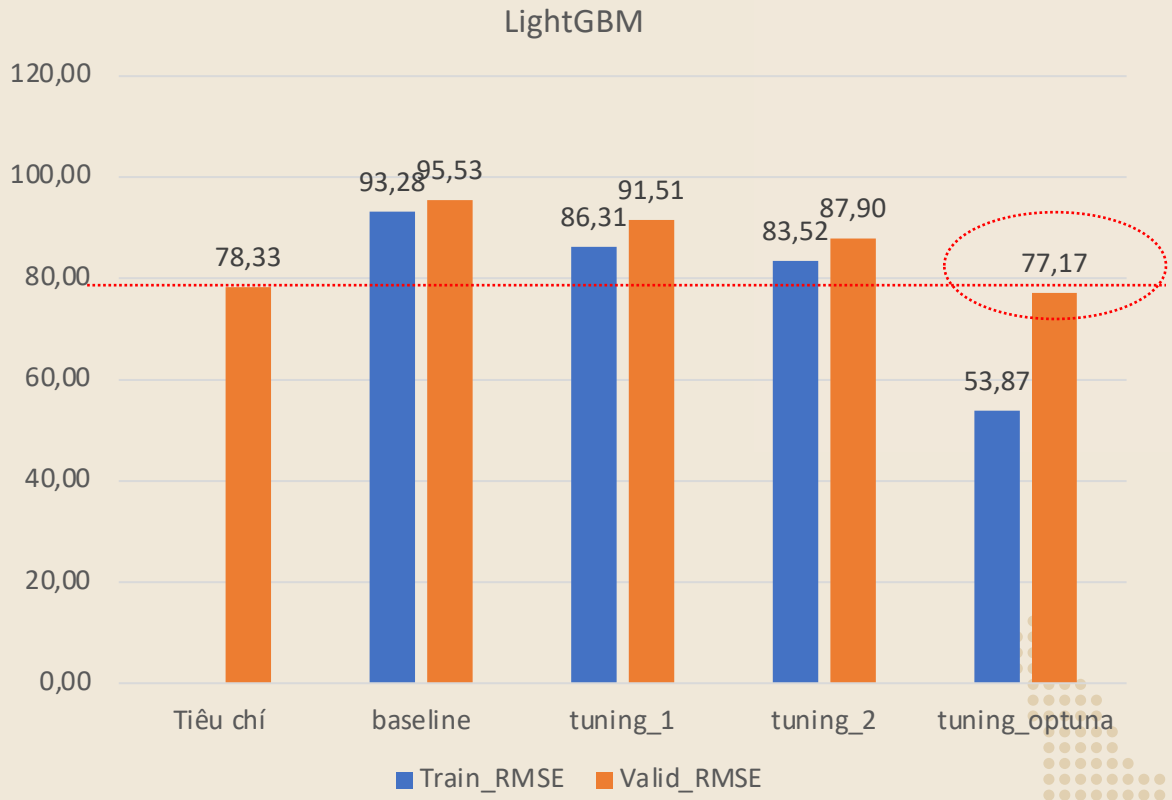
XGBoost	tuning_1	tuning_2	tuning_optuna
subsample	0,8	0,95	0,984
learning_rate	0,05	0,05	0,053
n_estimators	3000	3500	3255
max_depth	5	5	6

CatBoost	tuning_1	tuning_2	tuning_optuna
subsample	0,5	0,8	0,658
learning_rate	0,1	0,05	0,099
iterations	1000	3000	2808
depth	7	7	6

Result Comparison

LightGBM results

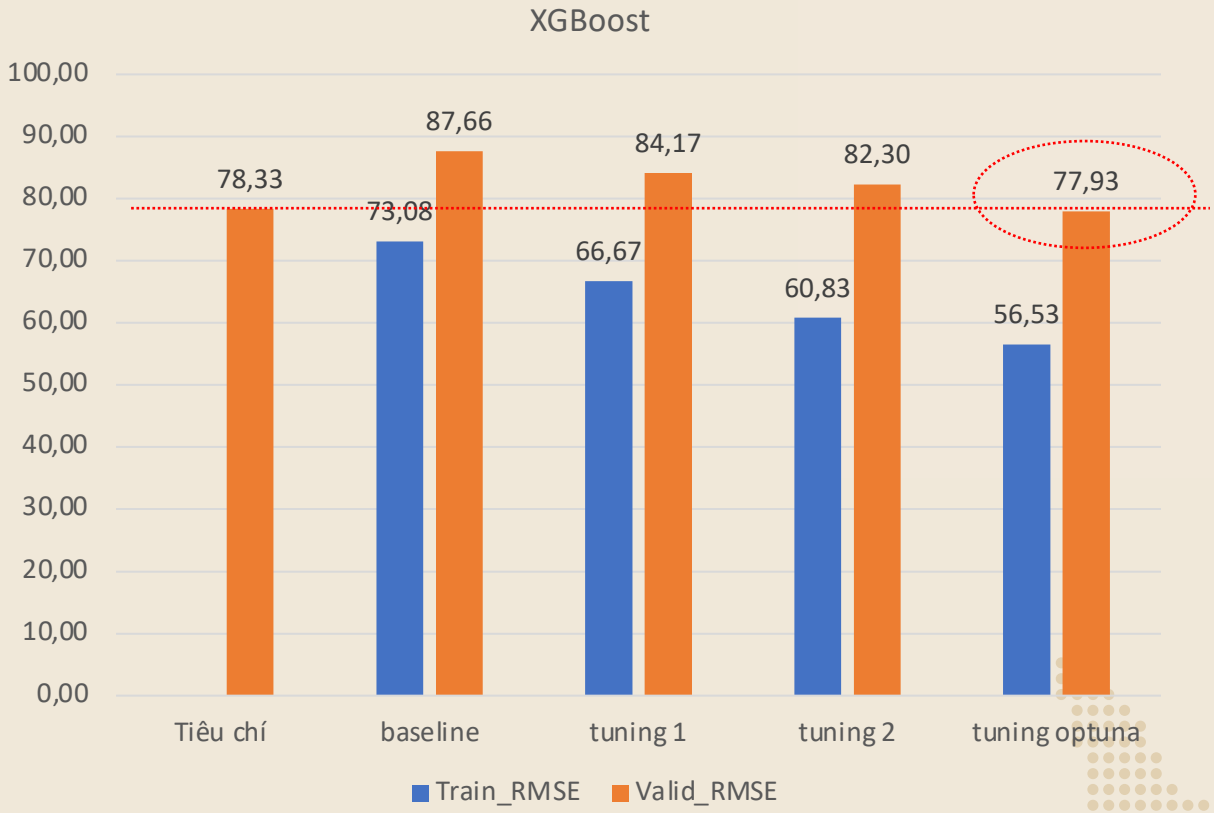
- The tuned LightGBM model successfully achieved the target RMSE <= 78.83
- The best performance on the validation set was achieved with an RMSE = 77,17



Result Comparison

XGBoost results

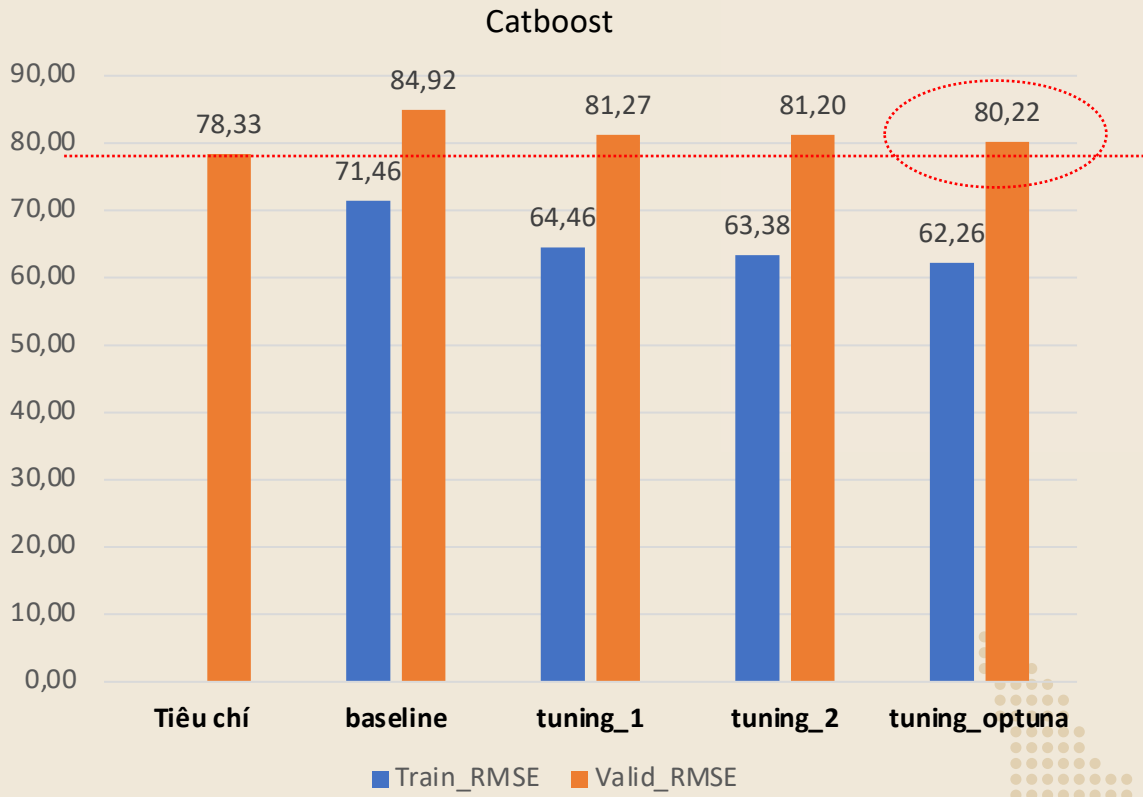
- The tuned XGBoost model successfully achieved the target RMSE <= 78.83
- The best performance on the validation set was achieved with an RMSE = 77,93



Result Comparison

CatBoost results

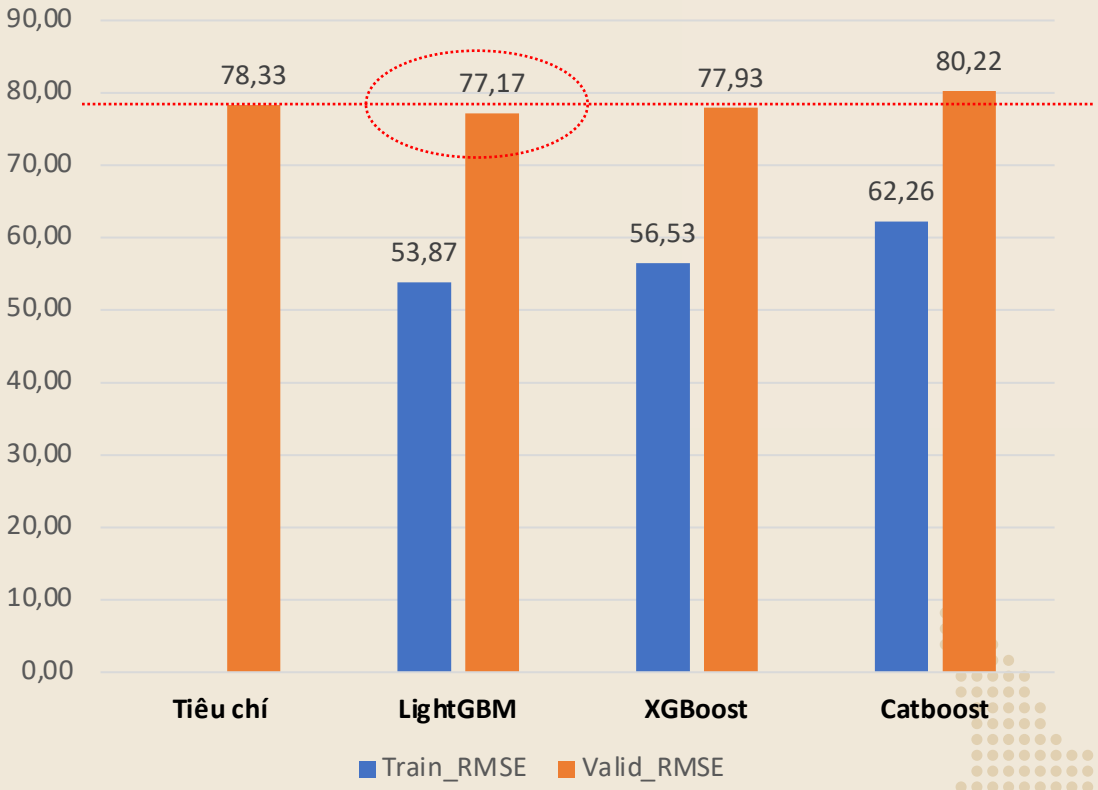
- The tuned CatBoost model did not achieve the target RMSE <= 78.83
- The best performance on the validation set was achieved with an RMSE = 80.22



Result Comparison

Summary of results

- To compare the performance of the best-tuned machine learning models
- Đạt tiêu chí RMSE <= 78.33
- The tuned LightGBM model achieved the best performance among the evaluated models with an RMSE = 77.17 on validation set



IV

Conclusion

Conclusion

Evaluation: Achieving Evaluation Criteria (RMSE <= 78.33) with Best Result of RMSE = 77.17

Selected Features:

- Original 12 features: tourist_area, spot_facility, area, city, type, category, tourism_index, info, event, year, month, day
- 15 created features: day_of_week, day_of_year, Weekday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, Spring, Summer, Fall, Winter, Holiday

Selected final model:

- Gradient Boosting Machine Learning Model using the LightGBM framework

Tuned Parameter Values (Other parameters set to default values):

- subsample: 0,894
- learning_rate: 0,085
- num_iterations: 4778
- max_depth: 45

Source code: https://github.com/supham95/Projects/tree/main/Final%20Project

Limitations and Extensions

Feedback from mentors and limitations:

- The analysis of errors and biases in the model's predictions is missing
- The explanations for data quantity and feature evaluation for categorical variables are lacking

Future Directions for Improving Tourist Volume Prediction:

- Employ Advanced Machine Learning Models
- Integrate External Data Sources (web search history, weather data, Social media data, etc.)
- Implement Real-time Data Analysis

References

- LightGBM Website. *Welcome to LightGBM's documentation!* <https://lightgbm.readthedocs.io/en/stable/>
- XGBoost Website. *XGBoost Documentation* <https://xgboost.readthedocs.io/en/stable/>
- CatBoost Website. *CatBoost Documentation* <https://catboost.ai/en/docs/>
- KENTAK0928 (2023). *Prediction of Tourist Arrivals* <https://www.kaggle.com/competitions/prediction-of-tourist-arrivals/overview>
- Aarshay Jain (2023). *Mastering XGBoost Parameter Tuning: A Complete Guide with Python Codes*
- <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/#XGBoost_Parameters>
- MJ Bahmani (2023). *Understanding LightGBM Parameters* <https://neptune.ai/blog/lightgbm-parameters-guide>
- Mario Filho (2023). *CatBoost Hyperparameter Tuning Guide with Optuna* <https://forecastegy.com/posts/catboost-hyperparameter-tuning-guide-with-optuna/>
- Pham Minh Hoang (2020). Ensemble learning và các biến thể (P1) <https://viblo.asia/p/ensemble-learning-va-cac-bien-the-p1-WAyK80AkKxX>
- Brain John (2023). *When to Choose CatBoost Over XGBoost or LightGBM [Practical Guide]* <https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm>

Thank you