Visitor prediction

Library selection:

- **pandas** for data processing tasks
- **numpy** for high-level mathematical functions
- **matplotlib** and seaborn of data visualization
- **pickle** for fast and efficient data storing
- **sklearn** packages for fast,build-in model preparation,optimization and evaluations functions plus the models itself
- **xgboost** for the XGBRegressor model

How to run the code - very brief - :

1. Make sure the required packages are available in your environment.
2. Open the jupyter notebook
3. Load the required packages
4. Load the support functions
    a. Go and train the model by going through:
        i. Collect the train data into a dataframe
        ii. EDA
        iii. Data Preparation
        iv. Feature Engineering
        v. Modeling
    b. Go and test a model
        i. Collect the test data into a dataframe
        ii. Load the saved model
        iii. Prepare the test dataset by using the data preparation and data engineering functions (so the test dataset will get as similar to the training dataset as it can and it is necessary for accurate predictions)
        iv. Predict the output using the loaded model
        v. Save the data and output columns into a .csv file

How and why transformed the data:

I have transformed some of the features during the feature engineering part of the exercise. The reason is that those features were categorical, therefore there was no logical order between them, however the algorithms we were going to use would misunderstand that. Their numerical values would make the model think that ‘2 = school holiday only in county #2’ have a higher value than ‘1 = school holiday only in county #1’, which is wrong. Therefore I renamed the numerical values for easier readability and transformed those columns into separate binary columns so the algorithm will not be misguided.

The approaches used:

It was clear from the beginning that we need a regression algorithm to predict the daily visit counts. Regression is a supervised machine learning technique for investigating the relationship between independent variables or features and a dependent variable or outcome.
The plan was to start with something simple such as a linear model to get more information about the ideal algorithm selection.
After the initial linear model - the baseline model - I tried out several more complex models as well. Out of all the regression models I have tried, the RandomForestRegressor and the XGBRegressor performed the best. I have chosen the RandomForestRegressor as my final algorithm because the RMSE score was similar to the XGBoost model's but the model overfit a little bit less than the XGBoost which indicates that it shall perform better on unseen data. (Unfortunately, the overfitting was significant and it is my main concern toward this model)
Random Forest Regression uses ensemble learning methods for regression. The ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. Therefore this algorithm shall be quite stable even though the model is overfitting.

The evaluation as well as a comparison with a base-line approach:

In summary, the final Random Forest Regression model seems good, it is overfitting therefore adding more relevant and unique features plus possibly using a less complex algorithm shall increase the reliability of the prediction.
However the final model's RMSE on the validation dataset was just under 300 so comparing it to the baseline model's 500 RMSE score, it is significantly more accurate.

The time you spend for solving the task : 5 hours