Methodology Document

# Technical Specifications

## Physical

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **Machine Type** | **Model** | **Processor** | **RAM** | **GPU** |
| 1. | Server | *Asus A555L* | *Intel® Core™ i3-5010U CPU (2.1 GHz, 64-bits)* | *8 GB* |  |
| 2. | Google Cloud | None | None | None | None |
| 3. | [IntelMovidius Neural Compute Stick](https://software.intel.com/en-us/movidius-ncs) | None | None | None | None |
|  |  |  |  |  |  |

## Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **OS/Software** | **Version** | **Details (any specifics)** | **URL** |
| 1. | *Windows 8.1 Pro* | *Version 6.3 (Build 6300)* |  |  |
| 2. | *Anaconda Software Package* | 1.9.6 | JupyterLab (0.35.1)  JupyterNotebook (5.7.0) |  |
| 3. | *Python* | 3.5 |  |  |
|  |  |  |  |  |

# Feature Summary

|  |  |  |  |
| --- | --- | --- | --- |
| amount\_spent\_per\_room\_night\_scaled (**target)** | numeric | 0 missing values |  |
| reservation\_id | nominal | 0 missing values (identifier) |  |
| booking\_date | Date time | 0 missing values |  |
| checkin\_date | Date time | 0 missing values |  |
| checkout\_date | Date time | 0 missing values |  |
| channel\_code | categorical | 0 missing values |  |
| main\_product\_code | categorical | 0 missing values |  |
| numberofadults | numeric | 0 missing values |  |
| numberofchildren | numeric | 0 missing values |  |
| persontravellingid | categorical | 0 missing values |  |
| resort\_region\_code | categorical | 0 missing values |  |
| resort\_type\_code | categorical | 0 missing values |  |
| roomnights | numeric | 0 missing values |  |
| season\_holidayed\_code | categorical | 35 missing values |  |
| state\_code\_residence | categorical | 4764 missing values |  |
| total\_pax | discrete | 0 missing values |  |
| member\_age\_bucket | nominal | 0 missing values |  |
| booking\_type\_code | categorical | 0 missing values |  |
| memberid | nominal | 0 missing values |  |
| cluster\_code | nominal | 0 missing values |  |
| reservation\_statusid | nominal | 0 missing values |  |
| resort\_id | nominal | 0 missing values |  |
| room\_type\_booked\_code | categorical | 0 missing values |  |
|  |  |  |  |

# Data Cleaning

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Column Name** | **Treatment** | **Details** |
| 1. | season\_holidayed\_code | Missing value | 35 missing values, filled with -999 |
| 2. | state\_code\_residence | Missing value | Filled with -999 |
|  |  |  |  |
|  |  |  |  |

# Feature Engineering

## Transformation

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Column Name** | **Transformation** | **Details** |
| 1. | Booking\_date | Convert to datetime | Initially, object type |
| 2. | Checkin\_date | Convert to datetime | Initially, object type |
| 3. | Checkout\_date | Convert to datetime | Initially, object type |
| 4. | Member\_age\_bucket | Encode to numerical form | Initially, object type |
| 5. | Member\_id | Encode to numerical form | Initially, object type |
| 6. | Cluster\_code | Encode to numerical form | Initially, object type |
| 7. | Reservationstatusid\_code | Encode to numerical form | Initially, object type |
| 8. | Resort\_id | Encode to numerical form | Initially, object type |

## Derived Variable

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **New Column Name** | **Treatment** | **Details** |
| 1. | Booking\_booking,  Booking\_month,  Booking\_day,  Booking\_dayofweek,  Booking\_weekofyear | Extract the year, month, day, week from the date given | Used booking\_date (basically, extraction make easy to handle the features as well as we come to know about the yearly, monthly and daily details) |
| 2 | Checkin\_year,  Checkin\_month,  Checkin\_day,  Checkin\_dayofweek,  Checkin\_weekofyear | Extract the year, month, day, week from the date given | Used checkin\_date |
| 3. | Checkout\_year,  Checkout\_month,  Checkout\_day,  Checkout\_dayofweek,  Checkout\_weekofyear | Extract the year, month, day, week from the date given | Used checkout\_date |
| 4. | fm | Add numberof adults and numberofchildren | Here, I just figure out the total family member who visited the resort |
| 5. | Count\_memid | Count unique reservation\_id | Here, I figure out the unique number of reservation\_id on groupby with memberid, |
| 6. | Diff\_booking\_prev | Difference between the current booking date and previous booking date in days | I found the day difference between the two consecutive booking of particular member, from that it become clear when the member visited resort as well as how frequently it was. |
| 7. | Diff\_checkin\_prev | Difference between the current booking date and previous booking date in days | I found the day difference between the two consecutive checkin of particular member, from that it become clear when the member visited resort as well as how frequently it was. |
| 8. | Diff\_checkout\_prev | Difference between the current booking date and previous booking date in days | I found the day difference between the two consecutive checkout of particular member, from that it become clear when the member visited resort as well as how frequently it was. |

# Model Run

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Run No.** | **Model** | **Metric** | **Value** | **Hyperparameter values** | |
| 1 | XGBoost | RMSE | RMSE: 97.53 | | Eda= 0.03  Metric= ‘rmse’  Max\_leaves= 51  Nthread= 7  Max\_depth= 8  Seed= 2019  Subsample= 0.5  Colsample\_bytree= 0.65  Feature\_selector= ‘greedy’  (Used 5 fold GroupFold method, groupby= ‘memberid’) | |
| 2 | LightGBM | RMSE | RMSE: 97.03 | | Learning\_rate= 0.01  Objective= ‘regression’  Metric= ‘rmse’  Num\_leaves= 51  Num\_threads= 7  Bagging\_fraction= 0.65  Feature\_fraction= 0.5  Seed= 2019  (Used 5 fold GroupFold method, groupby= ‘memberid’) | |

# Coding Details

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Programming Language** | **Package Used** | **Details** |
| 1. | Python | Scikit learn | sklearn.preprocessing  sklearn.metrics  sklearn.model\_selection  (used for the label encoding, cross validation and finding metrics) |
| 2. | Python | matplotlib | Used for the plotting purpose |
| 3. | Python | pandas | Used for the data processing |
| 4. | Python | Lightgbm | Used for the LightGBM modeling |
| 5. | Python | XGBoost | Used for the XGBoost modeling |

For more visit my github repo: - <https://github.com/sngupta07/AV_Challenges-/tree/master/Club_Mahindra>