Hotel Price Prediction Project Report

Eser Inan Arslan
March 2024

CONTENTS

Hotel Price Prediction Project Report

This report includes prediction models of house pricing for Mews.

Project Overview

The primary objective of this project was to develop a predictive model for determining the appropriate factor to apply to hotel pricing based on demand. The goal was to optimize revenue by dynamically adjusting hotel prices in response to fluctuating demand. The project involved exploring datasets containing information necessary for analysis and modeling.

Understanding the Problem

- Familiarized with the project goal of predicting the factor for hotel pricing.
- Considered the variables available in the dataset and their relevance to pricing decisions.

Data Exploration

- Conducted exploratory data analysis (EDA) to gain insights into the distribution, trends, and patterns within the data.
- Identified potential correlations or key features that may impact hotel pricing.

Model Development

- Choose suitable predictive modeling approaches such as regression and machine learning algorithms to create the model.
- Trained the model using the provided dataset to predict the factor for hotel pricing.

Evaluation

- Assessed the performance of the model using appropriate metrics such as Mean Absolute Error (MAE) and R-squared.
- Documented the evaluation process and results comprehensively.

Documentation

- Prepared a comprehensive report detailing the methodology, key findings, challenges faced, and recommendations.
- Include any code, visualizations, or graphs that support your analysis.

Summary of Process

EDA

I merged two separate datasets representing different hotel types, resulting in a combined dataset comprising 119,390 observations and 32 features. Following the data merge, I delved into exploratory data analysis (EDA) and visualization techniques to glean insights into the dataset's characteristics and patterns. The comprehensive analysis and visualizations are meticulously documented in the "01_Mews_eda.ipynb" file, providing a detailed understanding of the data distribution, trends, and relationships between variables.

During the preprocessing stage, I addressed NULL values in the 'agent' and 'company' columns by replacing them with zero values, considering the significance of these features in the context of the problem solution. Additionally, I identified 'ADR' (Average Daily Rate) as the target feature for modeling, as it serves as a pivotal metric in hotel pricing decisions.

Despite conducting feature correlation analysis, I observed relatively low correlation between input and output features. Therefore, I adopted a cautious approach and selected features with an absolute correlation value exceeding 0.05 to ensure the inclusion of potentially influential variables in the modeling process. This meticulous feature selection process aimed to enhance the model's predictive performance and generalizability.

Preprocess

I employed a series of preprocessing steps to prepare the dataset for modeling. This included applying label encoding and one-hot encoding techniques to handle categorical variables effectively, ensuring compatibility with machine learning algorithms. Additionally, I grouped numeric values into bins and normalized all numeric features using MinMaxScaler to standardize their scales and mitigate the impact of varying magnitudes on model performance.

Furthermore, I amalgamated date-related columns and adjusted them based on the last day of the dataset, facilitating a unified temporal perspective across the dataset. This adjustment ensured consistency and accuracy in temporal analysis and modeling. Moreover, I partitioned the dataset, designating the last month's data as the test dataset, while using the remaining data for model training and validation purposes.

The test dataset, encompassing the final month's observations, exhibited a shape of (4897, 41), providing a suitable sample for assessing the trained model's predictive performance. By meticulously preprocessing the dataset and partitioning it into appropriate subsets, I aimed to optimize the modeling process and enhance the reliability of the predictive model's outcomes.

Model

Faced memory issues with the Random Forest algorithm due to resource constraints, resulting in downsizing of the dataset. Naive Bayes algorithms did not encounter memory issues. Utilized the following algorithms are listed below:

1. Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem with the "naive" assumption of independence between features. It calculates the probability of each class given a set of input features and selects the class with the highest probability as the prediction. Despite its simplicity and computational efficiency, Naive Bayes may oversimplify complex relationships in the data, leading to suboptimal performance in certain scenarios.

2. Calibrated Naive Bayes with Sigmoid Function

Calibrated Naive Bayes with Sigmoid is an extension of the Naive Bayes algorithm that incorporates probabilistic calibration using the sigmoid function. This calibration technique adjusts the raw output probabilities of the Naive Bayes classifier to better reflect the true likelihood of class membership. By mapping the raw scores to calibrated probabilities, this approach aims to improve the reliability of the model's probability estimates, particularly for imbalanced datasets or when probabilistic outputs are required.

3. Calibrated Naive Bayes with Isotonic Function

Similar to the sigmoid calibration method, Calibrated Naive Bayes with Isotonic Regression enhances the probabilistic calibration of Naive Bayes by employing isotonic regression. Isotonic regression is a non-parametric technique that fits a monotonically increasing function to the raw output scores, thereby transforming them into calibrated probabilities. This method is particularly useful for addressing issues of overconfidence or underconfidence in the raw predictions of Naive Bayes, leading to more accurate and well-calibrated probability estimates.

4. Random Forest Regressor

Random Forest Regressor is an ensemble learning algorithm that constructs multiple decision trees during training and outputs the average prediction of the individual trees. Each tree in the forest is trained on a random subset of the training data and a random subset of features, promoting model diversity and reducing overfitting. Random Forest is known for its robustness, scalability, and ability to capture complex nonlinear relationships in the data. However, it may suffer from computational inefficiency and lack of interpretability, especially with large datasets.

5. Calibrated Naive Bayes with Sigmoid Function

The Sigmoid Calibrated Random Forest algorithm enhances the traditional Random Forest model by applying calibration using the sigmoid function. This calibration adjusts the raw outputs of the model to produce well-calibrated probabilities, ensuring more accurate probability estimates for classification tasks. By mapping raw scores to probabilities within the [0, 1] range, the algorithm improves the reliability and accuracy of predictions, particularly in applications requiring calibrated confidence scores or probability estimates.

As a result for algorithm perspectives, each algorithm has its strengths and weaknesses, and the choice of algorithm depends on various factors such as the nature of the data, the complexity of the problem, and computational resources. By leveraging a combination of these algorithms and calibration techniques, we aim to develop a predictive model that optimizes hotel pricing and enhances revenue generation effectively.

To optimize model performance and fine-tune algorithm parameters, I implemented GridSearch pipeline for hyperparameter tuning across selected algorithms. This systematic approach allowed for comprehensive exploration of hyperparameter space, aiding in identifying the most optimal configuration for each algorithm. Despite the computational demands, GridSearch enabled efficient parameter tuning, enhancing the models' predictive capabilities and generalization performance.

Furthermore, I observed varying execution times across different algorithms. While Random Forest, known for its complexity and computational intensity, typically required 30-40 minutes to complete, the Naive Bayes solutions exhibited significantly shorter execution times, completing within minutes. This discrepancy in execution times underscores the trade-off between model complexity and computational efficiency, highlighting the importance of selecting appropriate algorithms based on computational resources and modeling objectives. By

leveraging GridSearch for hyperparameter tuning and considering algorithm-specific computational characteristics, I aimed to strike a balance between model performance and computational efficiency, ultimately enhancing the overall effectiveness of the predictive modeling process.

Model Evaluation

Model accuracy and evaluation performance extensively on different thresholds.

- *Summary of MAPE (Mean Absolute Percentage Error)*
    - Naive Bias: 76.10%
    - Isotonic Naive Bias: 26.64%
    - Sigmoid Naive Bias: 25.62%
    - Random Forest: 17.61%
- *Percentage of rows with MAPE below 10%*
    - Naive Bias: 18.45%
    - Isotonic Naive Bias: 31.96%
    - Sigmoid Naive Bias: 26.75%
    - Random Forest: 55.54%
- *Percentage of rows with MAPE below 50%*
    - Naive Bias: 27.82%
    - Isotonic Naive Bias: 79.69%
    - Sigmoid Naive Bias: 84.17%
    - Random Forest: 91.78%

Challenges Faced

- Memory limitations with Random Forest algorithms due to large dataset size.
- Difficulty in ensuring model interpretability, especially with complex algorithms like Random Forest.
- Challenges in feature engineering and selection amid a large number of variables.
- Time-intensive model training, particularly for complex algorithms.
- Selection of appropriate evaluation metrics for model performance assessment.
- Dealing with data quality issues and handling missing values, especially in categorical features.

Conclusion & Recommendations

Based on the evaluation of the data, the conclusion can be made that the project achieved its objectives. Overall, the developed models showed promising performance in predicting the factor for hotel pricing based on demand. The Random Forest algorithm outperformed the Naive Bayes models, providing the lowest MAPE and the highest percentage of accurate predictions. Further optimization and fine-tuning of models may enhance performance and contribute to better revenue optimization for accommodations.

Based on these observations, the following suggestions can be made:

- Consider exploring additional features or data sources that may further improve model performance.
- Continuously monitor and update the predictive models to adapt to changing demand patterns and market conditions.
- Collaborate with domain experts to incorporate domain knowledge and insights into the modeling process for better predictions and revenue optimization.