PROJECT

HOME STAYS

About Project

Goal

The project aims to explore the data through `comprehensive exploratory data analysis (EDA)` and construct a `robust machine learning model` for the task, followed by detailed extraction of the model's insights.

Building a robust predictive model to estimate the `log_price` of homestay listings based on comprehensive analysis of their characteristics, amenities, and host information.

About Dataset

The dataset `"Homestays_Data.csv"` is provided for a project aimed at building a predictive model to estimate homestay prices based on various characteristics. With over 70,000 rows, it contains columns like 'log_price', 'host_since', 'amenities', 'last_review', 'room_type', 'property_type', and geographic coordinates and others.

DataLink : Home stays Data

Contents:

Some Visualization are missing you can see them

Technologies Used

1. pandas: Data manipulation and analysis in tabular format.

2. pickle: Serialization and deserialization of Python objects, often used for saving and loading machine learning models.

3. xgboost, lightgbm, CatBoostRegressor: Gradient boosting algorithms for regression tasks. These are powerful algorithms for predictive modeling.

4. sklearn.metrics.mean_squared_error, mean_absolute_error, r2_score: Evaluation metrics for regression models.

5. sklearn.model_selection.train_test_split: Splitting data into training and testing sets for model evaluation.

6. sklearn.linear_model.LinearRegression, Ridge, Lasso, ElasticNet: Linear regression and its regularized versions for regression tasks.

7. sklearn.tree.DecisionTreeRegressor: Decision tree-based regression algorithm.

8. sklearn.ensemble.RandomForestRegressor, GradientBoostingRegressor: Ensemble learning algorithms for regression tasks.

9. sklearn.feature_selection.RFE, SelectKBest, mutual_info_regression: Feature selection techniques for selecting the most important features for modeling.

10. sklearn.preprocessing.StandardScaler: Standardization of features by removing the mean and scaling to unit variance.

11. sklearn.decomposition.PCA: Dimensionality reduction using Principal Component Analysis.

12. numpy: Numerical computing, especially for arrays and matrices.

13. seaborn, matplotlib.pyplot, plotly.express, scipy.stats.skew: Data visualization libraries for generating insights and outcomes from data.

14. geopy.geocoders.Nominatim, geopy.distance.geodesic: Geocoding and distance calculation between geographic locations.

15. transformers.AutoModelForSequenceClassification, TFAutoModelForSequenceClassification, AutoTokenizer, AutoConfig, pipeline: Components from Hugging Face's Transformers library for natural language processing tasks like sequence classification.

16. sklearn.impute.KNNImputer: Imputation of missing values using k-nearest neighbors.

17. warnings: Python's built-in module for issuing warnings.

Problems & their solutions in Dataset

Missing Data & Imputation :

The missing values in the columns ['bathrooms', 'first_review', 'host_has_profile_pic', 'host_identity_verified', 'host_response_rate', 'host_since', 'last_review', 'neighbourhood', 'review_scores_rating', 'thumbnail_url', 'zipcode', 'bedrooms', 'beds'] were handled using a combination of random imputation, mean/mode imputation, and KNN imputation.

Encoding Techniques:

I utilized two encoding techniques for model training: one-hot encoding and label encoding. Interestingly, the label encoding method yielded the best results, although I experimented with both encodings and attached the outcomes for your review.

Feature Selection:

1. Starting with filter methods: This method is chosen for its efficiency and ease of interpretation. It involves techniques like correlation analysis, information gain, and L1 regularization to identify potentially relevant features from the dataset.

2. Refining with wrapper or embedded methods: To further enhance feature selection, you've incorporated wrapper methods such as forward selection and recursive feature elimination (RFE), as well as embedded methods like LASSO regularization. These methods consider the interaction between features and can improve model performance, albeit at a higher computational cost. They're employed after filter methods have narrowed down the pool of candidate features.

3. Considering PCA: In cases of high-dimensional data, Principal Component Analysis (PCA) is being considered. PCA reduces the dimensionality of the dataset while retaining its most informative features. This step aims to streamline the dataset for more efficient modeling without sacrificing crucial information.

Ultimately, the project's focus on filter methods underscores a pragmatic approach, prioritizing simplicity and ease of implementation while ensuring efficient feature selection tailored to its objectives and constraints.

Model Selection:

For model selection, I experimented with a variety of algorithms, including simple baseline models such as Linear Regression, Decision Tree Regression, and Random Forest Regression, along with more advanced techniques like XGBoost, LightGBM, and CatBoost. After thorough evaluation, I ultimately selected XGBoost, LightGBM, and CatBoost as the final models for their superior performance and robustness in predicting the target variable.

Model Monitoring & Feature importance

For model monitoring, I relied on Evidently, a comprehensive toolkit for assessing model performance and monitoring model behavior over time. Additionally, for feature extraction insights, I employed SHAP (SHapley Additive exPlanations), a powerful technique for understanding the impact of features on model predictions and extracting meaningful insights from complex machine learning models.

Insights form the EDA

There are alot of findings i cant write it here you can refer to This Link

Model Insights

I H've evaluated three models: XGBoost, LightGBM, and CatBoost. Let's summarize their performance:

- **XGBoost:**
  - Parameters: Learning rate of 0.1, max depth of 3, and 100 estimators.
  - Evaluation Metrics: MSE of 0.172, MAE of 0.304, and R^2 of 0.664.

- **LightGBM:**
  - Parameters: Maximum depth of 3, subsample of 0.8, and colsample bytree of 0.8.
  - Evaluation Metrics: MSE of 0.173, MAE of 0.304, and R^2 of 0.664.

- **CatBoost:**
  - Parameters: Default parameters.
  - Evaluation Metrics: MSE of 0.171, MAE of 0.297, and R^2 of 0.667.

Among these models, CatBoost performed the best with the lowest MSE, MAE, and highest R^2 score, indicating superior accuracy in predicting the target variable.