CSE 343 : Machine Learning Project Final Presentation

Forecasting Hospitality Costs: A Data-Driven Journey into Hotel Room Price Prediction

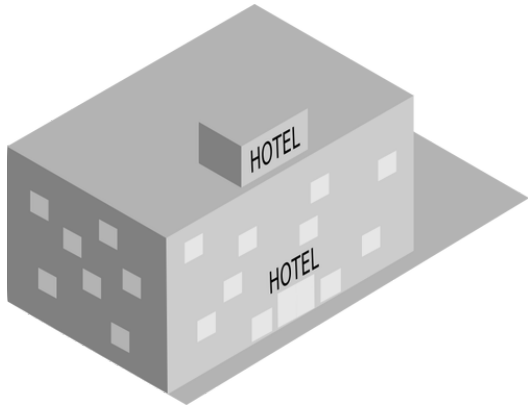
INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Motivation

- Accurate hotel price prediction has became very important for the convenience of both travelers and hotel management.

- Understanding property prices is essential for gauging an area's development.

- By forecasting prices effectively, hotels can optimize occupancy rates, ensuring that as many rooms as possible are booked at the right price point.



Motivation

- Since the prices of hotel depends on various factors, such as location, infrastructure, management etc.
- Machine learning algorithms can be used in order to predict the prices of the hotel/property.
- The price prediction is an regression problem hence regressive machine learning algorithms can be applied on the data of the prices of the hotels.
- Therefore this will help the hospitality industry to effectively manage their prices in order to be more competitive and profitable.

Literature Review

- Airbnb Price Prediction using Machine Learning and Sentiment Analysis

  The article covers how several machine learning approaches were used to create a pricing prediction model for Airbnb rental units.

  The researchers emphasized the significance of customer reviews in influencing the pricing of Airbnb listings, they employed sentiment analysis and assigned -1 (very negative sentiment) to +1 (very positive sentiment) to each review.

  ML Techniques used: Ridge Regression, K-means clustering with Ridge Regression, Support Vector Regression (SVR), Neural Network, Gradient Boosting Tree Ensemble.

  Performance metrics: Evaluated trained models using Mean Absolute Error (MAE), Mean Squared Error (MSE) and R^2 score.

  Key findings: Among the models tested, SVR performed the best.

  Note: For detailed insights,refer to full research paper by Rezazadeh Pouya, et al.

Literature Review

- Real Estate Price Prediction with Regression and Classification.

  This research paper predicts the prices of the real estates, by both regression, by directly predicting the prices and classification, by making dividing the price ranges in to intervals and then predicting.

  Researchers used PCA techniques in every model they tried to get better results, by reducing the dimensionality, of the model.

  ML Techniques used:

  - *For Regression problem*: Linear reg., Lasso, ridge, SVR(linear, gaussian kernel), Random Forests rig.
  - *For Classification problem* : Naive Bayes,Logistic reg, SVC(linear, gaussian kernel), random forest classification.

  Performance metrics: Accuracy of the model is used as an metric for classification problem and RMSE value is used as an metric for regression analysis.

  Key findings: SVC with linear kernel is best performing for classification, and SVR with gaussian kernel is best for regression analysis.

  living area square feet, material of the roof, and neighborhood have the greatest statistical significance in predicting a house's sale price.

  Note: For detailed insights,refer to full research paper by Hujia Yu et al.

Literature Review

- Warehouse Rental Price estimation using Machine Learning Techniques

  The research paper addresses the undergoing change the rental warehouse market is experiencing due to growing logistics industry and digital transformation.

  The research involves predicting warehouse rental price estimation using machine learning techniques using data from classified ads.

  ML Techniques used: Explored Linear Regression, Regression Tree, Random Forest Regression, Gradient Boosting Regression Trees.

  Performance metrics:Evaluated models using RMSE and r-squared

  Key findings: Random Forest outperformed single factor models

  Location ,distance from city center and local land prices crucial for price prediction.

  Note: For detailed insights,refer to full research paper by Yixuan et al.

Dataset Description

- The original dataset included 74,112 entries, each with 29 features.

- Target to predict was prices of the property, given various features about the property, such as no. of accommodates, no. of bedrooms and beds in it, bathrooms etc.

Dataset Description

	id	log_price	accommodates	bathrooms	latitude	longitude	number_of_reviews	review_scores_rating	bedrooms	beds
count	7.411100e+04	74111.000000	74111.000000	73911.000000	74111.000000	74111.000000	74111.000000	57389.000000	74020.000000	73980.000000
mean	1.126662e+07	4.782069	3.155146	1.235263	38.445958	-92.397525	20.900568	94.067365	1.265793	1.710868
std	6.081735e+06	0.717394	2.153589	0.582044	3.080167	21.705322	37.828641	7.836556	0.852143	1.254142
min	3.440000e+02	0.000000	1.000000	0.000000	33.338905	-122.511500	0.000000	20.000000	0.000000	0.000000
25%	6.261964e+06	4.317488	2.000000	1.000000	34.127908	-118.342374	1.000000	92.000000	1.000000	1.000000
50%	1.225415e+07	4.709530	2.000000	1.000000	40.662138	-76.996965	6.000000	96.000000	1.000000	1.000000
75%	1.640226e+07	5.220356	4.000000	1.000000	40.746096	-73.954660	23.000000	100.000000	1.000000	2.000000
max	2.123090e+07	7.600402	16.000000	8.000000	42.390437	-70.985047	605.000000	100.000000	10.000000	18.000000

- We observe that there are some missing values in bathrooms and review scores ratings and in other features.
- We can see that the minimum value for price is 0 which is not possible hence proper outlier removal is necessary.
- Also we can see that min, max and mean of all the features are highly varied therefore scaling is required before training the models.
- So data cleaning and preprocessing is required in the dataset.

Data Preprocessing

- All duplicate entries were removed from the dataset.

```
df.drop_duplicates(inplace=True)
```

- All null values where dropping can be done (without loss much data) were dropped.

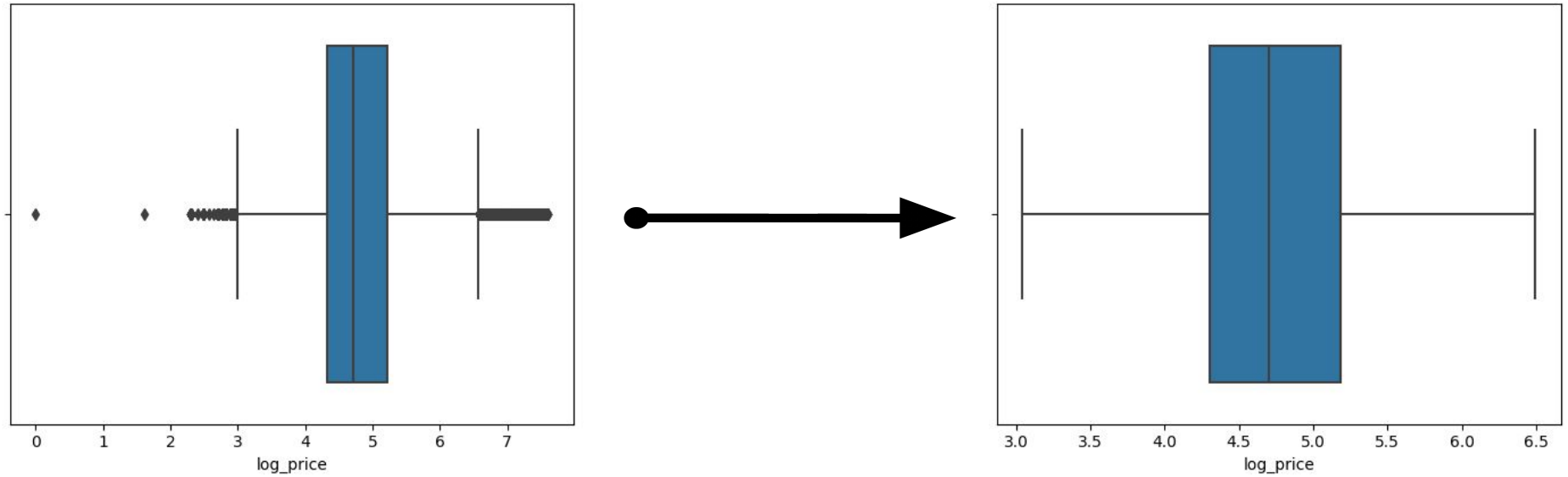
```
df = df.dropna(subset=['host_has_profile_pic', 'host_identity_verified'])
```

- Features which were irrelevant for price prediction such as thumbnail url, name, description of the host, etc. were removed so that most pertinent features were used for model prediction.

```
useless_data = ['description', 'first_review', 'host_since', 'last_review', 'name', 'thumbnail_url', 'zipcode',
'neighbourhood', 'amenities']
df = df.drop(useless_data, axis=1)
```

Data Preprocessing

- We observed that there were outliers in our dataset with the help of box plots.



- So we removed the outliers.

Data Preprocessing

- We filled the mean value in other places where simply record dropping might lead to huge data loss.

```
df['host_response_rate'].fillna(df['host_response_rate'].mean(),inplace=True)
```

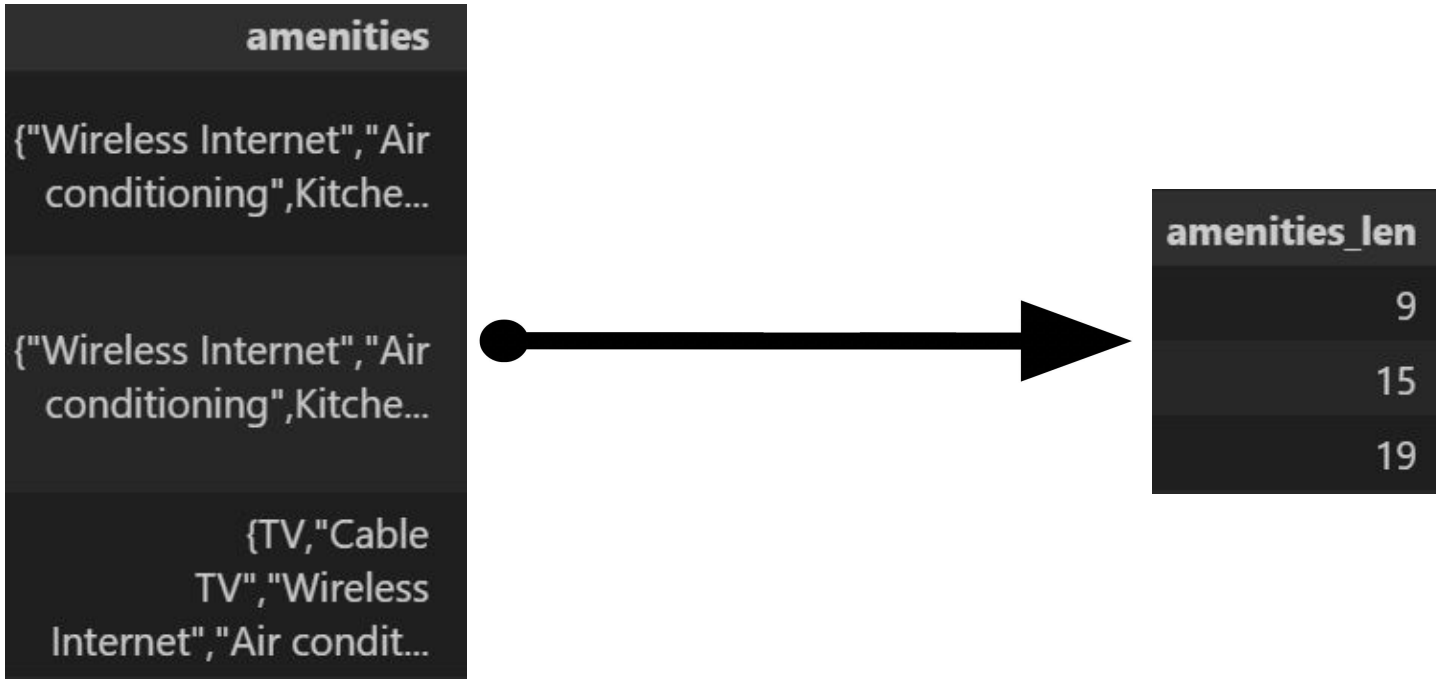
- Some features has data in string format, proper formatting of the data is done

```
df['host_response_rate'] = df['host_response_rate'].str.replace('%', '')
df['host_response_rate'] = df['host_response_rate'].astype(float)
```

- Irrelevant feature for price prediction such as url, name of the host etc. are dropped.

Data Preprocessing

- A new feature, 'Amenities,' is derived from an existing feature to enable its use in the model.



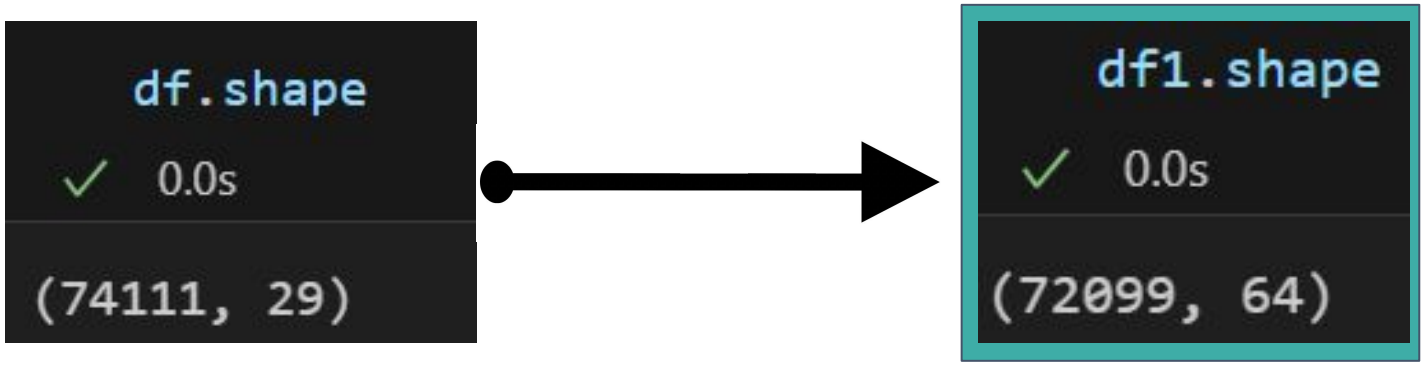
Data Preprocessing

- To deal with categorical data, we have employed the strategy of encoding the data using both label and one-hot methods wherever necessary.

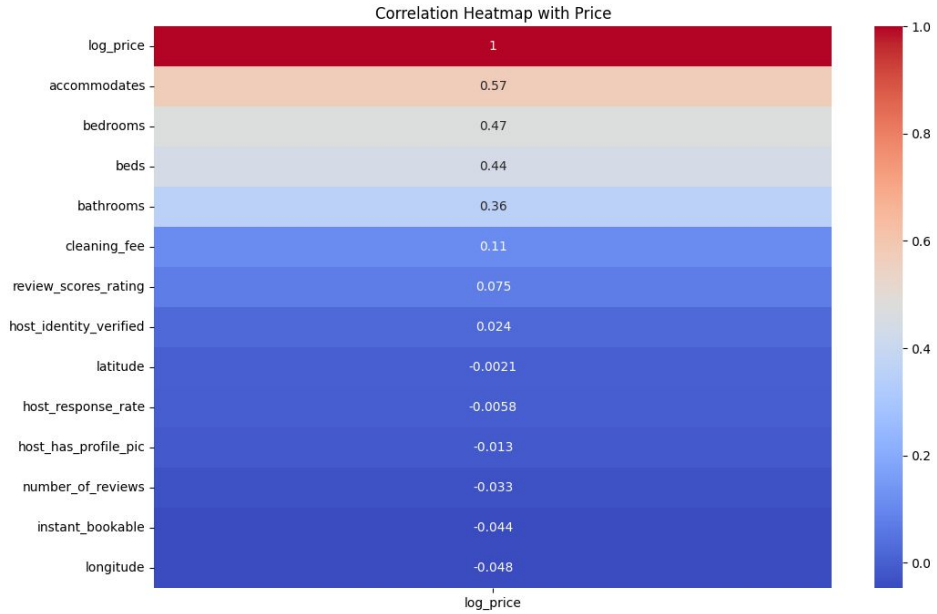
```
le = LabelEncoder()
df1['host_has_profile_pic'] = le.fit_transform(df1['host_has_profile_pic'])
```

```
encoder = OneHotEncoder(sparse_output=False, handle_unknown='ignore', drop='first')
X_encoded = encoder.fit_transform(df1[categorical_data])
```

- Now the final shape of the dataframe.

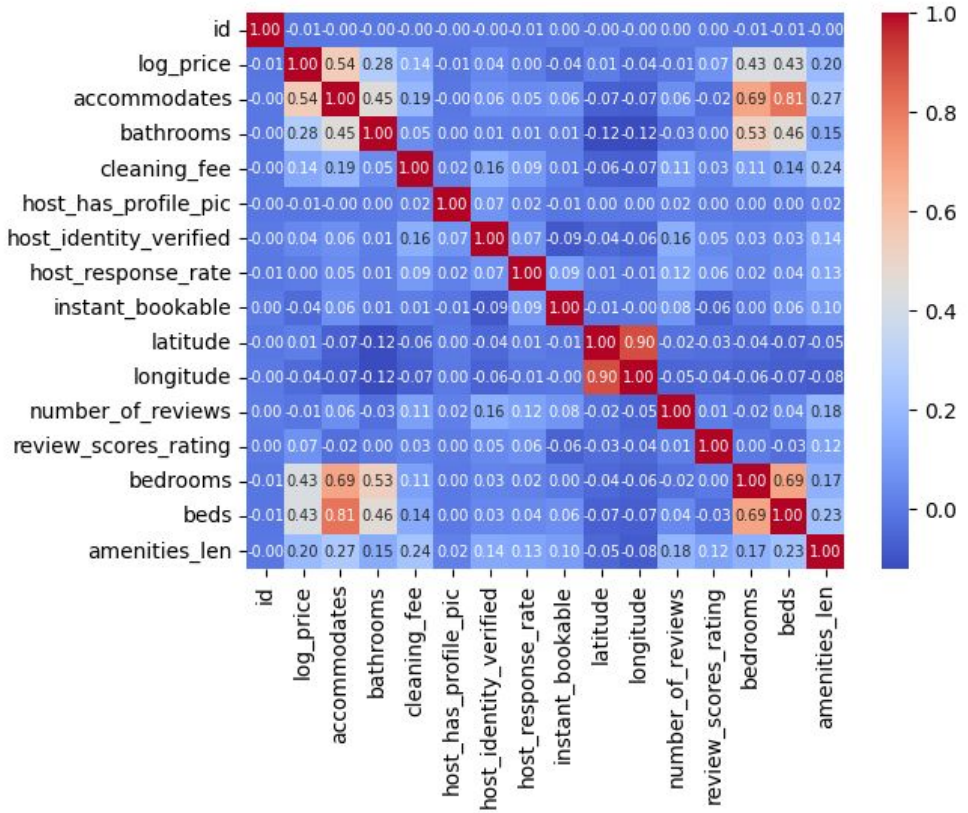


Data Analysis



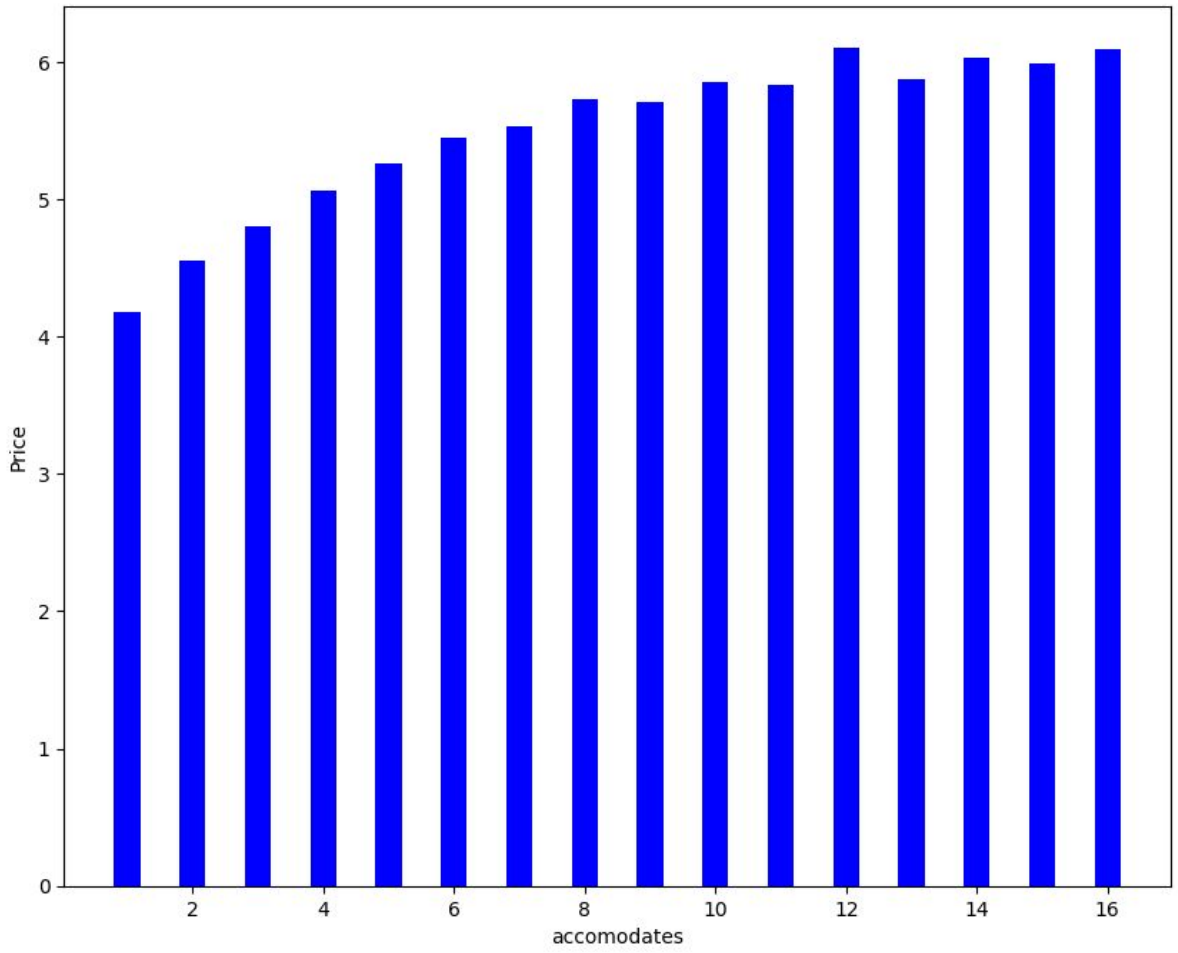
- **Correlation Heatmap** shows that the prices mainly depends on factors such as accommodates, bedrooms, bed, bathrooms, and cleaning fee.

- By doing this we get an idea about the relation of each feature with the target feature.



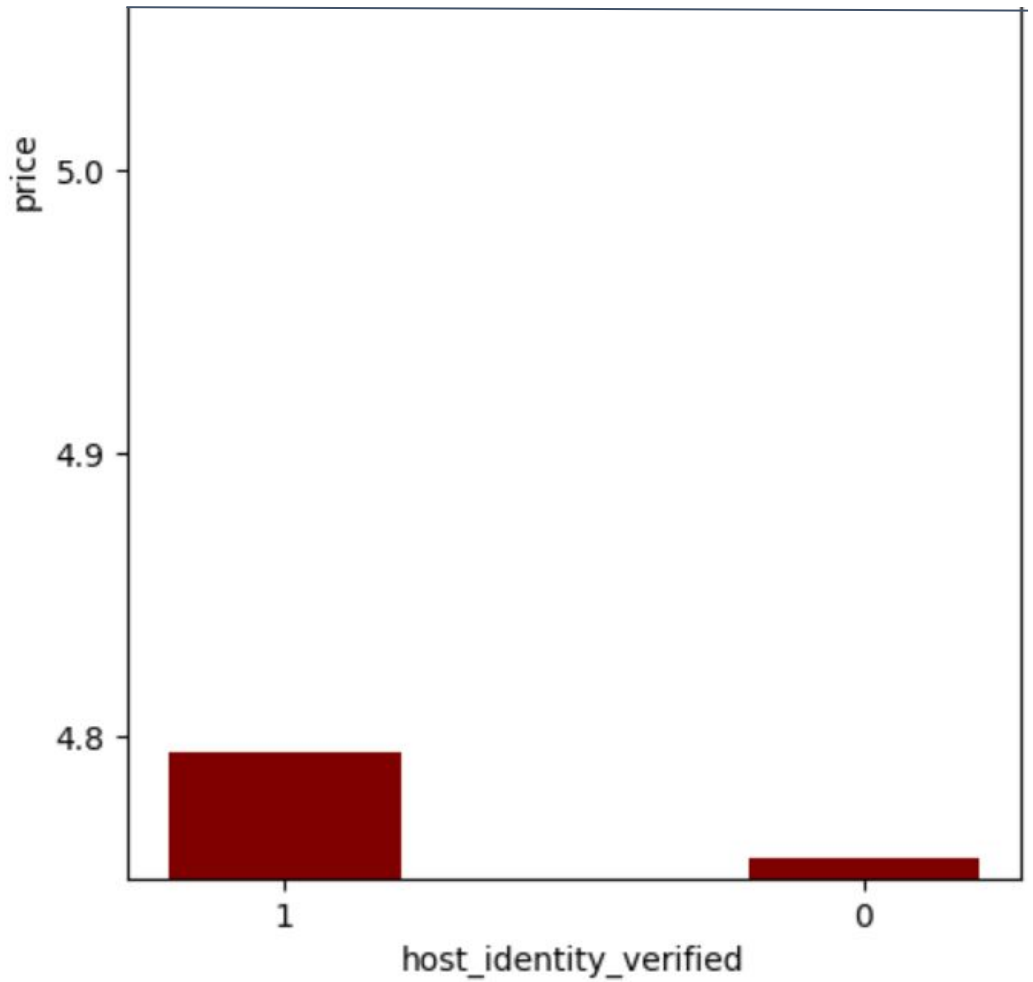
- Correlation heatmap of our dataset.

Data Analysis



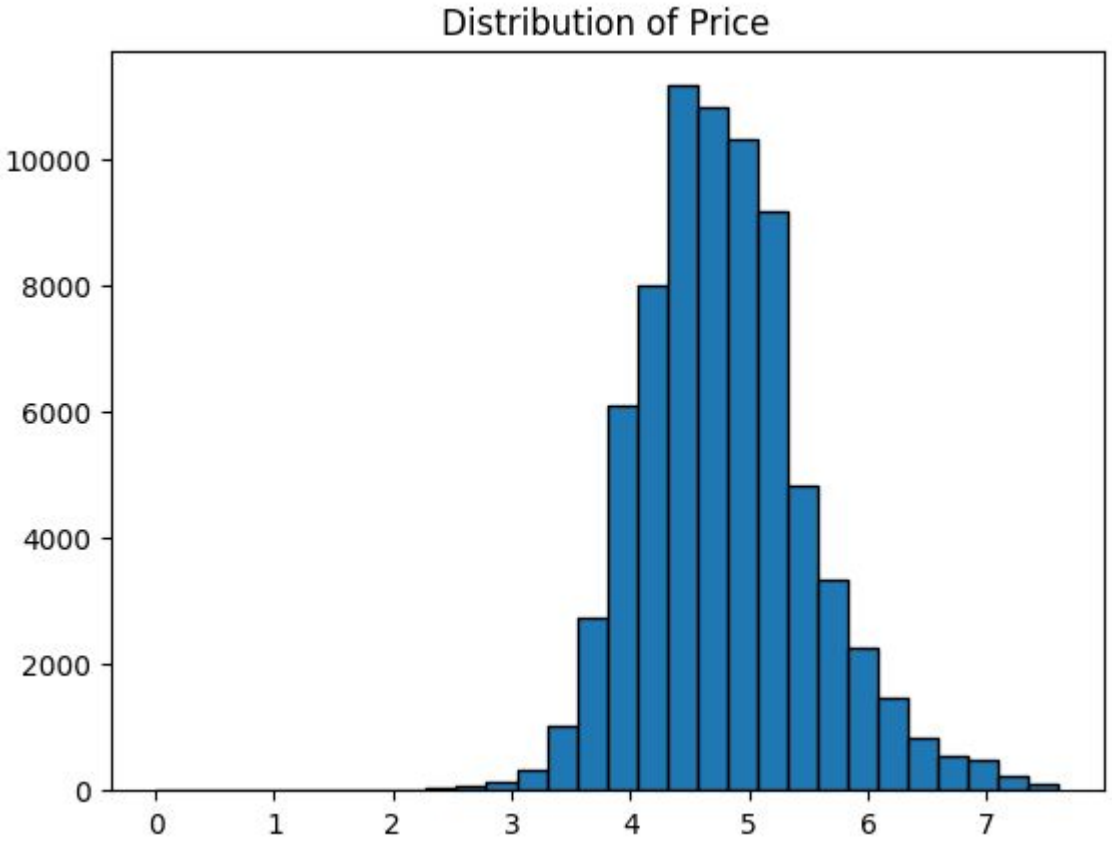
- As seen from the correlation heatmap, we plotted accommodation with their average prices to see the relation.
- It can be seen that, logarithmic relation exists between price and accommodates.

Data Analysis



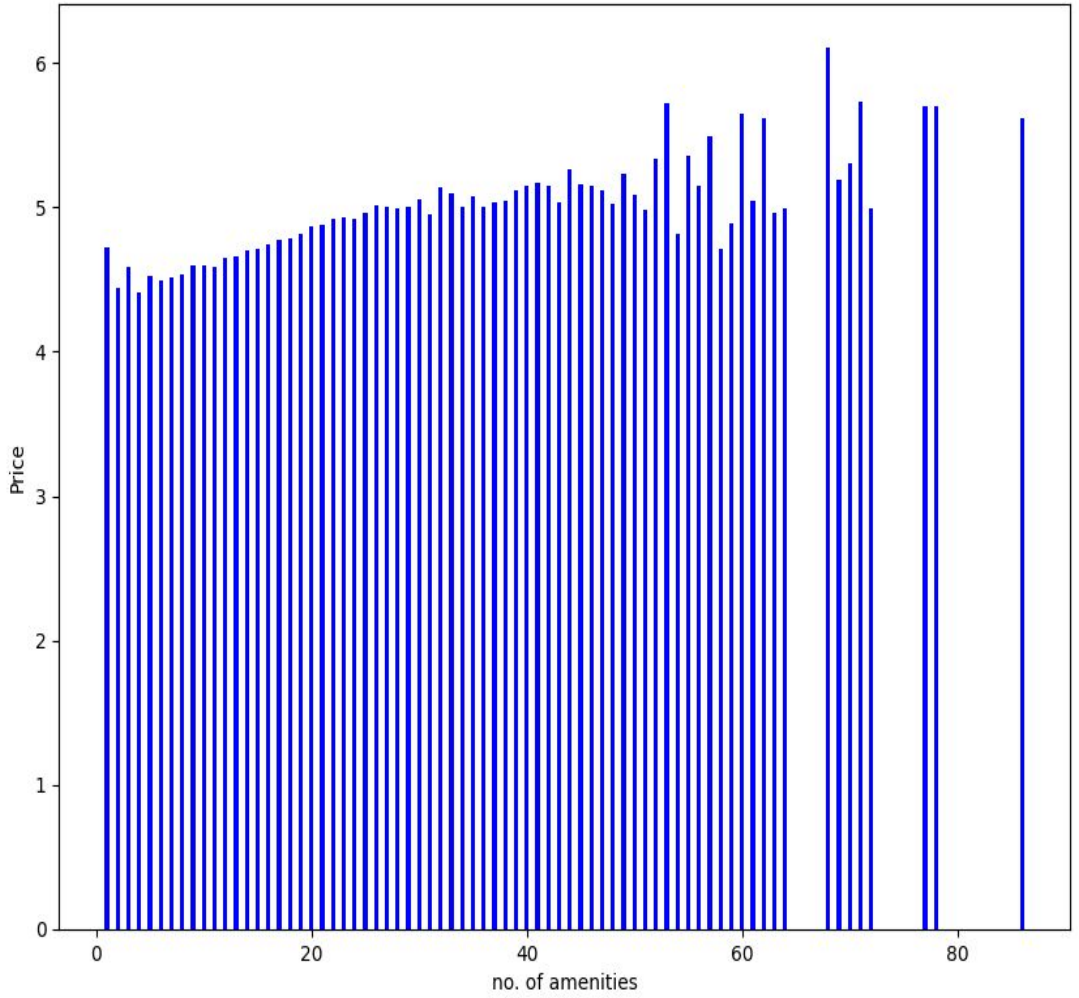
- We have plotted this graph to analyse the relation between host identity verified and average hotel price.
- As seen from the graph, the average price of the hotel increases if the host is verified.

Data Analysis



- The prices are distributed in a normal (Gaussian) manner, which is considered one of the best distributions for the model to learn effectively.

Data Analysis

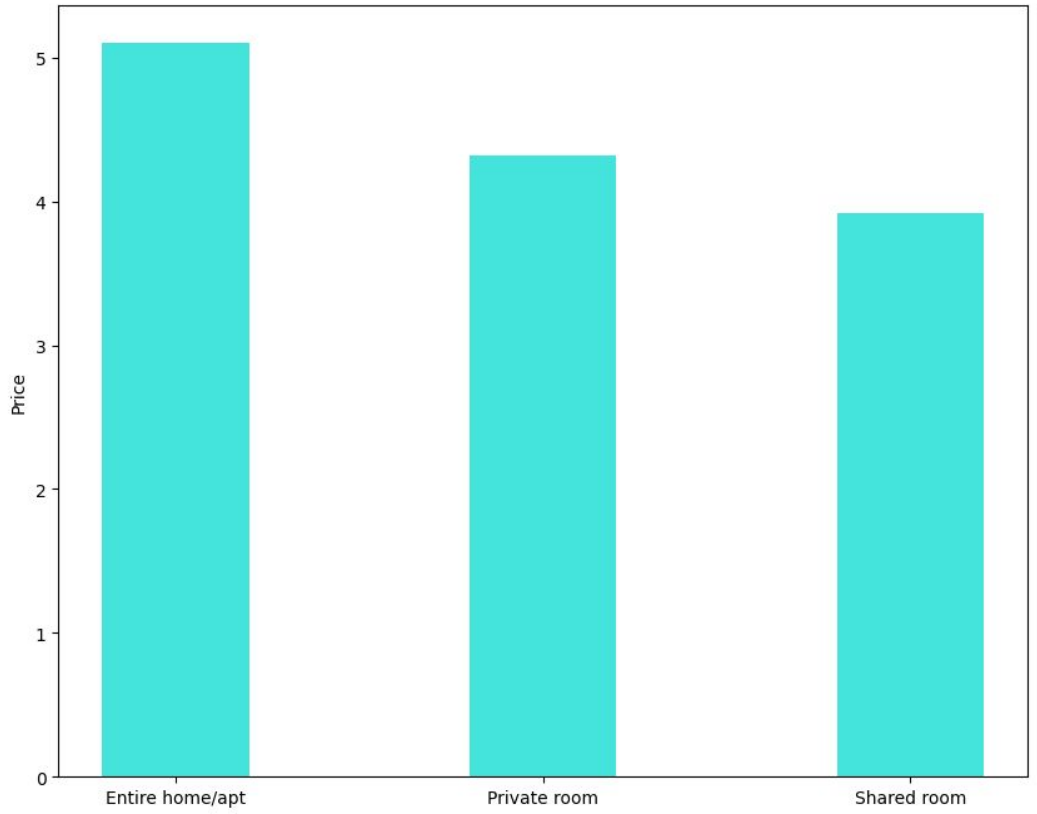


- This graph shows the relation between the newly constructed feature “no. of amenities” and price.

- It can be seen from the graph that as the no. of amenities increase, the average price also increases. Thus there is a linear relation between them.

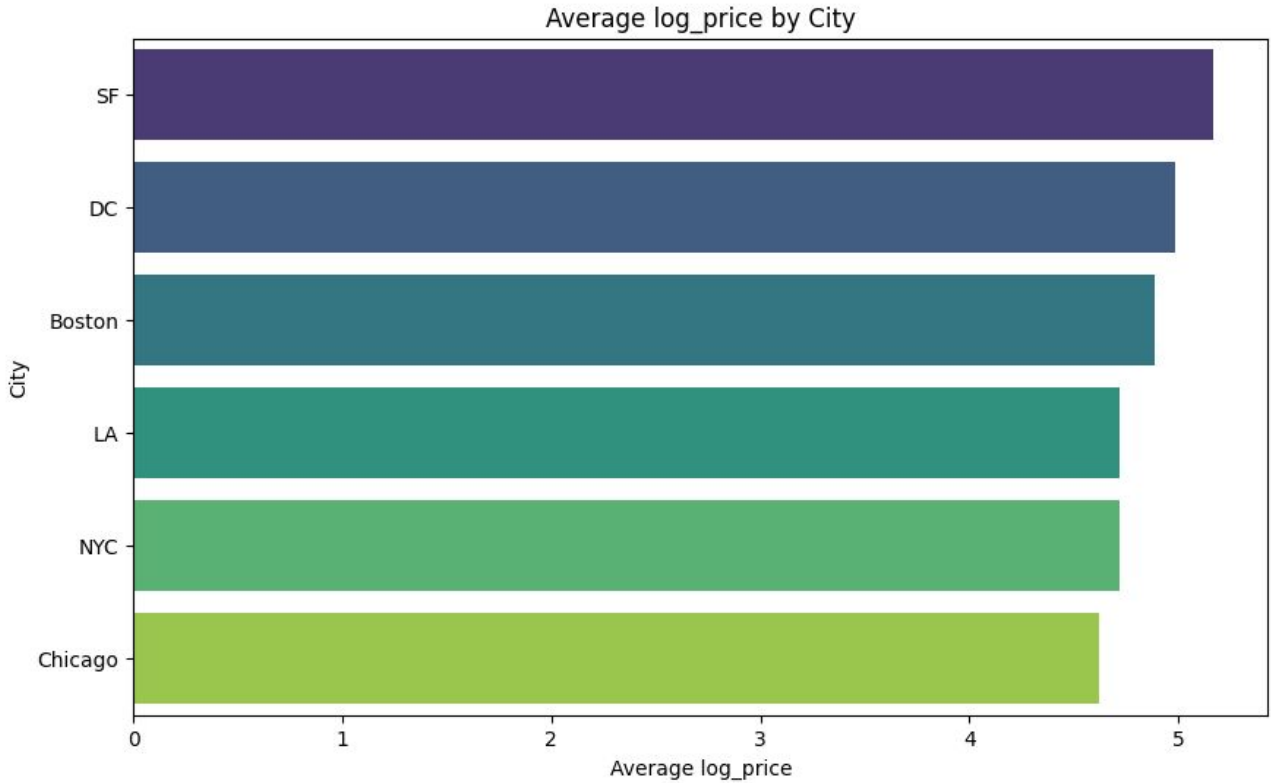
Data Analysis

- The shared room is the least expensive type of accommodation, while the entire home/apartment is the most costly property. This can be observed in the graph depicting the average price for each category.

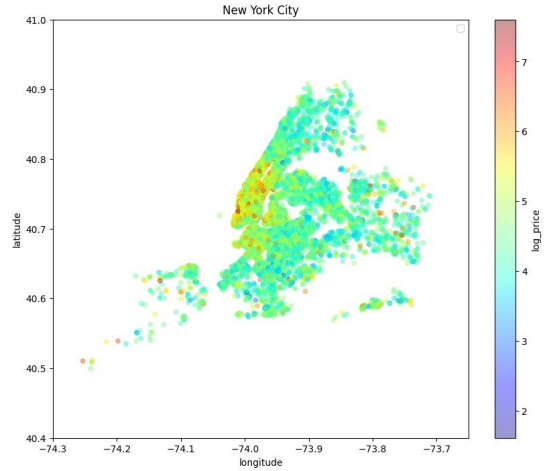


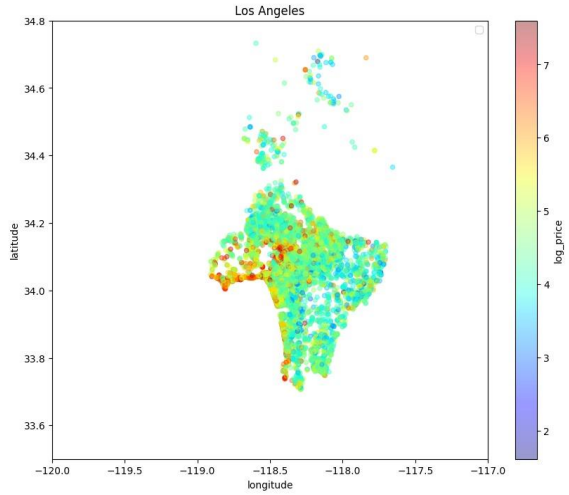
Data Analysis

- Average prices are different in different cities, with san francisco being the maximum and chicago being the lowest.



Data Analysis





- These are the scatter plots of New York City and Los angeles showing the distribution of hotels across the city using latitude and longitude for marking the position.
- The variation in colour of the distribution shows the variation in prices of hotel. Bluish color signifies low price while reddish signifies high price.

Methodology

Objective: To predict the log prices of rental properties using different machine learning regression techniques while incorporating feature engineering,feature scaling, and dimensionality reduction techniques.

Created a dataframe df1 for exploring distinct preprocessing strategies.

df1:

- Clustered data using K Means and added the cluster label as a feature in the dataset. This label captures the underlying patterns in the data.
- Applied StandardScaler to standardize numerical features before applying any type of encoding to improve the performance of certain algorithms
- Label encoded and one-hot encoded categorical features in the dataframe.

Methodology

Techniques applied:

Linear Regression:

→Evaluation on testing data: MSE, RMSE, R2 Score, MAE.

```
Mean Square Error : 0.18
Root Mean Square Error : 0.43
R2 Score : 0.56
Mean Absolute Error : 0.33
```

Regularized Regression: Lasso and Ridge Regression

→Evaluation on testing data: MSE, RMSE, R2 Score, MAE.

```
Mean Square Error : 0.46
Root Mean Square Error : 0.67
R2 Score : 0.54
```

Lasso

```
Mean Square Error : 0.44
Root Mean Square Error : 0.66
R2 Score : 0.56
```

Ridge

Methodology

Support Vector Machine regression:

→PCA for dimensionality reduction

→Trained SVM model(Kernels explored: linear kernel, rbf)

→Evaluation on testing data

```
Mean Square Error : 0.42
Root Mean Square Error : 0.65
R2 Score : 0.58
Mean Absolute Error : 0.50
```

MLP

→Used GridSearch on hidden_layer_sizes, activation function and solver.

```
{'activation': 'logistic', 'batch_size': 200, 'hidden_layer_sizes': (256, 128)}
```

→Evaluation on testing data : MSE, RMSE, R2 Score, MAE.

```
Mean Squared Error: 0.42
Mean Absolute Error: 0.50
Root Mean Squared Error: 0.65
R-squared (R²) Score: 0.57
```

Methodology

Decision Tree Regressor:

→Used GridSearch on hyperparameters such as max_depth, min_samples_leaf,min_samples_split as part of hyperparameter tuning.

```
Best Hyperparameters: {'max_depth': 10, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 10}
```

→ Evaluation on testing data: MSE, RMSE, R2 Score, MAE.

```
Mean Squared Error: 0.60
Mean Absolute Error: 0.58
Root Mean Squared Error: 0.78
R-squared (R²) Score: 0.39
```

```
Mean Squared Error: 0.37
Mean Absolute Error: 0.46
Root Mean Squared Error: 0.61
R-squared (R²) Score: 0.63
```

Improvement After hyper parameter tuning

Methodology

Random Forest Regressor:

→Evaluation on testing data: MSE, RMSE, R2 Score, MAE.

```
Mean Squared Error: 0.31
Mean Absolute Error: 0.41
Root Mean Squared Error: 0.56
R-squared (R²) Score: 0.69
```

XGBRegressor (Gradient Boosting Algorithm):

→Used GridSearch to tune the hyperparameters.

→Evaluation on testing data: MSE, RMSE, R2 Score, MAE.

```
Best parameters found:
{'learning_rate': 0.1, 'max_depth': 10, 'subsample': 0.9}
```

```
Mean Squared Error: 0.30
Mean Absolute Error: 0.41
Root Mean Squared Error: 0.55
R-squared (R²) Score: 0.70
```

Results

	Linear Regression	Lasso	Ridge	SVM (rbf kernel)
MSE	0.44	0.46	0.44	0.42
RMSE	0.66	0.67	0.66	0.65
R^2	0.56	0.54	0.56	0.58
MAE	0.51	0.52	0.51	0.50

Results

	Decision Tree Regressor	Random Forest Regressor	XGBRegressor	MLP
MSE	0.37	0.31	0.30	0.42
RMSE	0.46	0.41	0.41	0.65
R^2	0.63	0.69	0.70	0.57
MAE	0.61	0.56	0.55	0.50

Conclusion

Linear Regression, Lasso, Ridge: Traditional linear models exhibit moderate performance. Linear, Lasso and Ridge offer competitive performance with reduced errors.

SVM (RBF Kernel) with PCA: Utilizing PCA, SVM demonstrates good predictive ability with a balanced trade-off between accuracy and model complexity.

Tree-Based & Ensemble Methods: Decision Tree, Random Forest, and XGBoost models outperform linear approaches, with XGBoost displaying the **best performance.**

MLP: It shows moderate yet balanced predictive power in capturing Airbnb prices.

Team member contributions.*

Task	Team Member
Data Collection	All
Data Preprocessing and visualization	Aryesh and Aakash
Feature Extraction	L.Kumar and L.Agrawal
Feature Analysis, Selection, Correlation	Aryesh and L.Agrawal
Linear regression, SVMs	Aryesh,Aakash, L.K.
RF, Decision Trees, NNs, XGBoost	L.K., L.A., Aakash
Analysis of the outcomes	All
Hyperparameter tuning	All
Writing Report	All
Presentations	All

***Each member of group has contributed to each task, and no task has been done entirely by one member.**

Thank You