Major Northeast Cities Airbnb Listings Price Prediction

Group Details

Name	Net ID
Pranavnath Bathula	pbb50
Anish Jha	aj992
Yash Gudimalla	yng9

Project Definition

The idea for the project originated from a problem encountered while planning a trip. During the attempt to reserve an Airbnb, it was problematic to assess the reasonableness of the pricing. For example, two apartments could appear very similar in the photos, but one could cost significantly more with no apparent reason. It seemed that there was a need for a better, more organized approach for using data to ask questions like “Is that a good price for that area?” or “What is the approximate price range for that level of apartment?” This is when the idea for the project was conceptualized, which would take on the idea of using the Inside Airbnb data set to predict the nightly price of Airbnb listings for apartments in the Major Northeast Cities area. Instead of simply loading the CSV file and using it for the model, it was recognized that the data should instead be considered material that should be actively managed, similar to how it is used in a real system.

The Inside Airbnb data is very comprehensive but also very flawed. It is provided in the form of several large CSVs, listing data, calendars showing daily availability, review data, and neighborhood polygons. However, these are not organized in a very accessible, tidy manner. There are columns that are stored as text when they actually have specific types, such as numbers, money, and dates, and there are also nulls. Our project cleans up the very raw form of this data, creating a much cleaner, easier-to-work-with representation. It involves the creation of a relational database, the import of the data into the database, and the construction of a data pipe that cleans the listings, assigns them to neighborhoods, and creates a features data structure for inclusion in machine learning models of pricing. One of the important aspects of this project is that it has a dual, interconnected approach. The data science approach focuses strictly on data cleaning, investigation of patterns, and features of possible influence on the price, like the number of bedrooms, borough, host super status, and number of reviews. The data management approach also strictly focuses on data storage and transfer, like the definition of tables for neighborhoods and listings, the use of foreign keys, and the design of the logic for the data extraction transformation, transforming raw CSV data into properly typed data stored in the database. Machine learning sits on top of the other two. The model can only learn features extracted by the pipe, and the quality of the features extracted is directly proportional to the data cleaning and organization. Our goal is not simply to calculate the error metric, RMSE, for example, but also to provide a complete reproducible path from the raw data of the Inside Airbnb CSVs to a well-functioning price prediction.

Introduction

The project's goal is to examine the pricing differences among Airbnb rentals in New York City, Boston, and Washington, D.C., and to understand what these prices reveal about the markets in these cities. To achieve this, we link the listings themselves to data about their neighbourhoods using a relational database. From here, we will compare the listing's attributes with the publicly available information at this time to determine an estimate for what a 'fair' nightly rental would be. This question is extremely relevant for both hosts and guests. The project focuses heavily on cleaning, taking data from a CSV format, organizing it, and adding it to a SQLite schema that includes neighbourhood data. It then establishes a complete feature set that can be used for predictive analysis. While the project follows many of the principles of this course through topics related to data cleaning and organization, relational databases, and analysis for predicting pricing, we will approach the problem by consulting with each city regarding potential issues with leakage from features that may not have existed prior to the listing going live.

The work we have done provides significant and novel outcomes in three key areas. First, by combining three disparate city data sets into a single, common schema and an engineered features table, we establish an analytical and engineering workflow that spans from raw data files to a full ML model. Secondly, we have engineered features to encode the characteristics about our listings such as capacity, level of hosting, review patterns, patterns of availability, and neighbourhood context (borough, neighbourhood name, and city level), while eliminating any leakage features associated with prices (such as price per bed or estimated revenue). Finally, we rigorously benchmark our models against simple strategies (global and neighbourhood mean) using metrics such as RMSE, MAE, R^2, etc. Our final cleaned multi-city data set (approximately 21k listings) produces our tuned Random Forest model with an RMSE of $75, MAE of approximately $47 and explains about ~72% of the variance in nightly pricing, which demonstrates that our engineered features and our approach to integrating databases yield higher-quality predictions over naive baselines.

Methodology

1) Data Science Component

Data Collection and Cleaning

- Processed raw Airbnb listings from NYC, Boston, and Washington DC
- Cleaned price fields (removed currency symbols, converted to numeric
- Removed invalid records and outliers
- Applied domain constraints: price ($10-$1000), accommodates (1-10), bedrooms (0-8), beds (0-10), bathrooms (0-5), review scores (1-5), availability (0-365 days)

Feature Engineering

- Price features: price per accommodate/bed/bedroom, log-transformed price
- Availability features: available days/rate, blocked/booked days/rate, availability ratio
- Review features: log-transformed review counts and reviews per month, high-rating indicator (≥4.8), active host indicator

- Temporal features: host years, host listings buckets, rating buckets, capacity buckets Geographic aggregations: neighborhood-level (avg/median price, listing count), city-level (listing count, superhost rate, avg rating, reviews per month, entire home share)
- Categorical encodings: room type flags, property type grouping

Data Quality Assurance

- Validated feature ranges and data types
- Handled missing values appropriately
- Ensured consistency across cities
- Final dataset: 20,631 listings with 59 engineered features

2) Database Component

Database Design

- SQLite database with normalized schema
- Two main tables: neighbourhood (borough, neighbourhood_name) and listing (property details, host info, pricing, reviews, availability)
- Foreign key constraints enabled for referential integrity
- Appropriate data types (INTEGER, REAL, TEXT, DATE)

ETL Pipeline

- Extracted cleaned CSV data from processed files
- Transformed data: boolean conversion (t/f → 0/1), date parsing, neighbourhood matching via borough and name, handling missing boroughs for Boston and Washington DC
- Loaded data: neighbourhoods first, then listings with foreign key resolution
- Integrated multi-city data with city identifiers

Data Integration

- Unified schema across cities
- Handled city-specific differences (e.g., borough availability)
- Maintained data relationships and constraints
- Final database: 20,631 listings across 3 cities

3) Machine Learning Component

Feature Selection and Preprocessing

- Excluded data leakage features: price-derived ratios, neighborhood price aggregations, estimated revenue
- Feature groups: 26 numeric (accommodates, bedrooms, coordinates, review metrics, availability metrics, host years, city aggregations), 8 categorical (city, borough, neighbourhood_name, room type, property_type_grouped, capacity_bucket, host_listings_bucket, rating_bucket), 6 binary (host_is_superhost, instant_bookable, room type indicators)

Preprocessing Pipeline

- Numeric features: median imputation, StandardScaler normalization
- Categorical features: most frequent imputation,

- OneHotEncoder with unknown category handling
- Combined via ColumnTransformer

Model Development

- Baseline models: global mean (RMSE: 143.09), neighborhood mean (RMSE: 124.57)
- Linear Regression: RMSE 94.61, MAE 63.27, R2 0.563
- Random Forest Regressor: 200 estimators, no max depth limit; RMSE 75.69, MAE 46.61, R2 0.720

Model Evaluation

- Train/test split: 80/20 (random_state=42)
- Metrics: RMSE, MAE, R-squared
- Random Forest outperformed baselines and linear regression, explaining 72% of price variance with a 47% reduction in RMSE compared to the neighborhood baseline

Output

- Final feature set saved to data/processed/listing_features.csv
- Trained models ready for production inference
- Preprocessing pipeline for consistent feature transformation

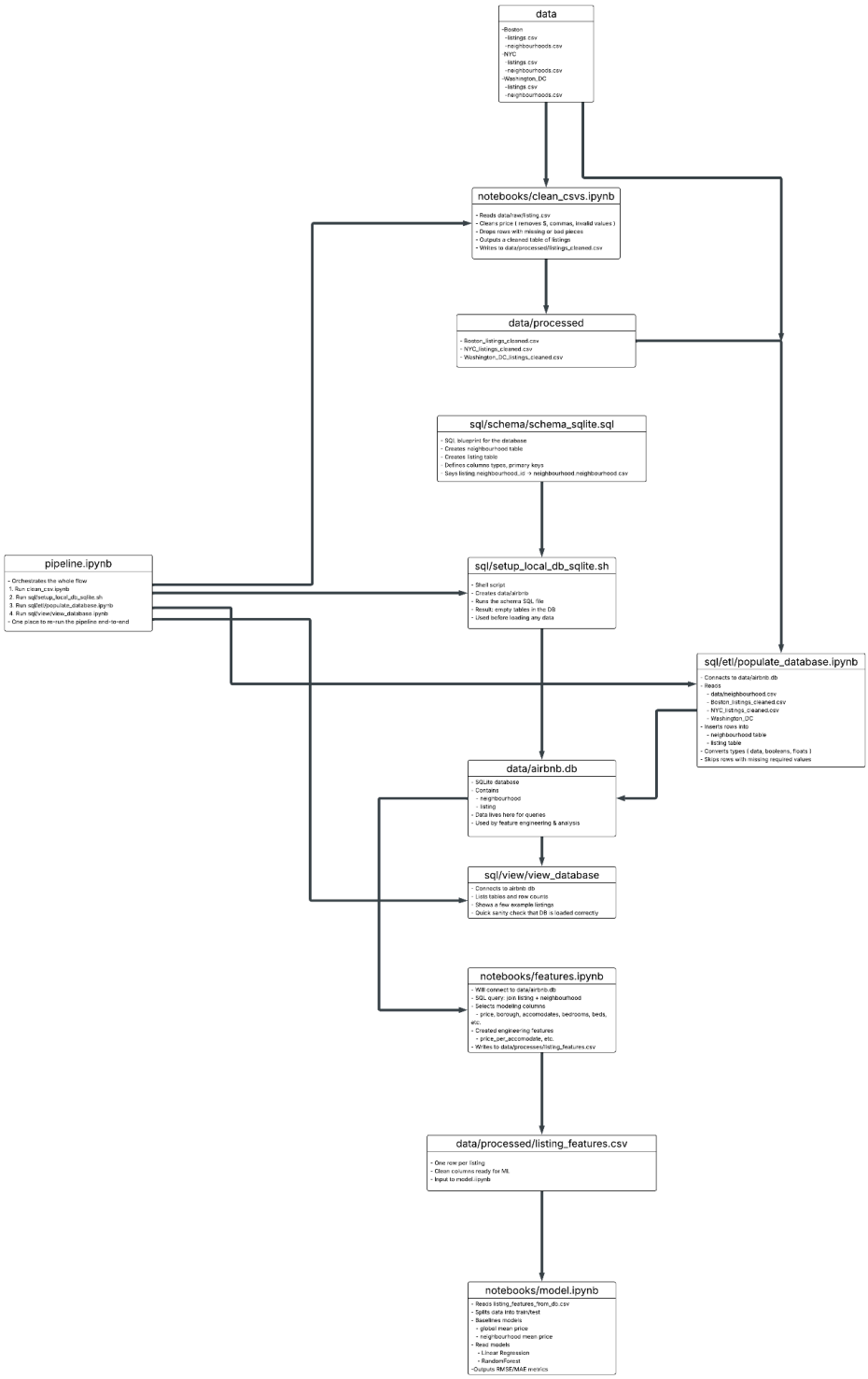
SQL Table

It is the main record keeper for our entire project. It's where we keep the detailed profile for every single Airbnb listing. We store everything from who the host is (and if they're a "Superhost"), to the specific layout of the apartment (like bedrooms and bed count), right down to exactly which neighborhood it's from. This setup is important because it organizes all the messy, raw info into one clean place, giving our model exactly what it needs to look at when it's trying to figure out a fair price.

```sql
-- SQLite Schema for Airbnb Price Predictor
-- Enable foreign key constraints
PRAGMA foreign_keys = ON;

CREATE TABLE neighbourhood (
    neighbourhood_id INTEGER PRIMARY KEY AUTOINCREMENT,
    borough          TEXT,
    neighbourhood_name TEXT NOT NULL
);

CREATE TABLE listing (
    listing_id   INTEGER PRIMARY KEY,

    neighbourhood_id INTEGER REFERENCES neighbourhood(neighbourhood_id),
    city         TEXT,  -- NYC, Boston, or Washington DC

    --Host Information
    host_id      INTEGER,
    host_name    TEXT,
    host_since   DATE,
    host_is_superhost INTEGER,  -- 0 = false, 1 = true

    --Property Information
    room_type    TEXT,
    property_type TEXT,
    accommodates INTEGER,
    bedrooms     INTEGER,
    beds         INTEGER,
    bathrooms    REAL,
    bathrooms_text TEXT,

    latitude     REAL,
    longitude    REAL,

    --Price and Availability
    price        REAL,
    number_of_reviews INTEGER,
    availability_365 INTEGER,
    estimated_revenue REAL,

    --Review Information
    first_review DATE,
    last_review  DATE,
    review_scores_rating REAL,

    instant_bookable INTEGER,  -- 0 = false, 1 = true
    calculated_host_listings_count INTEGER,
    reviews_per_month REAL
);
```

Project Pipeline Flowchart

This flowchart shows the entire Data Engineering and Machine Learning pipeline for the project. It illustrates the data lifecycle from Raw Ingestion, which involves processing different CSVs from NYC, Boston, and DC, to Storage, where cleaned data is loaded into the normalized airbnb.db SQLite database.Finally, it presents the Modeling phase, where data is extracted using the features.ipynb notebook to create the final training set, listing_features.csv, which is used to train the Linear Regression and Random Forest models.

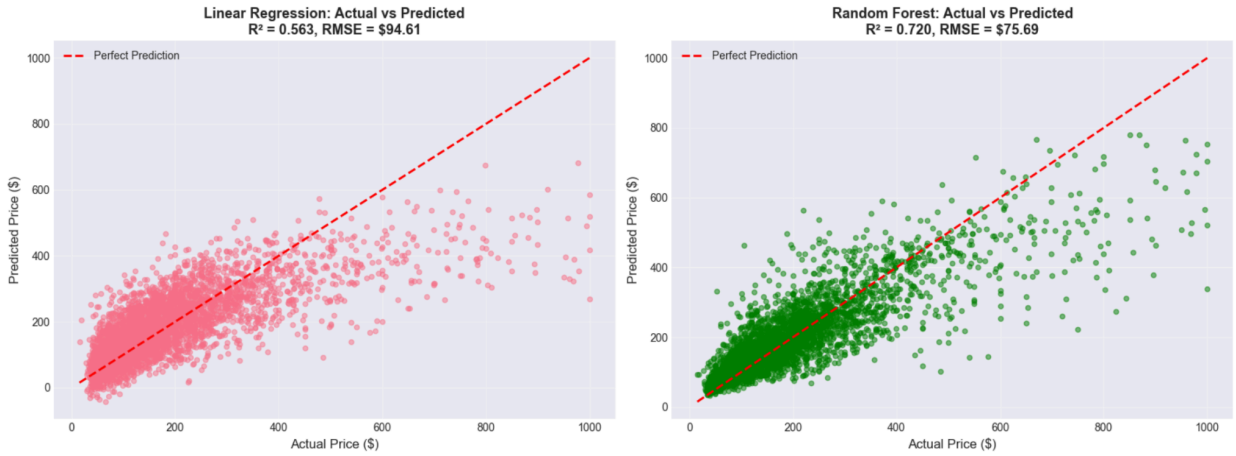


Results

```
=== Model comparison on test set ===
Global mean baseline RMSE:        143.09
Neighbourhood mean baseline RMSE: 124.57
Linear Regression RMSE:           94.61
Random Forest RMSE:               75.69

MAE scores:
Linear Regression MAE:            63.27
Random Forest MAE:                46.61

R^2 scores:
Linear Regression R^2:            0.563
Random Forest R^2:                0.720
```

To evaluate the effectiveness of our pricing pipeline, we compared our machine learning models against a Global Mean baseline (RMSE: 143.09) and a Neighborhood Mean baseline (RMSE: 124.57) to set a performance standard. The neighborhood baseline showed that location is a main driver of price, but it did not consider specific property features. We then trained Linear Regression and Random Forest models and evaluated them using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2. The Linear Regression model significantly surpassed the baselines with an RMSE of 94.61 and an R^2 of 0.563. This indicates it successfully captured basic linear trends but had difficulty modeling complex feature relationships.

The Random Forest Regressor stood out as the best model, achieving the lowest RMSE of 75.69 and an MAE of 46.61. This means our predictions were, on average, within about $47 of the actual price. This model explained around 72% of the variance in listing prices (R^2: 0.720), a significant improvement over the 56% variance explained by Linear Regression. By using an ensemble of decision trees, the Random Forest model effectively captured non-linear relationships, such as the varying premium of specific amenities across different boroughs. This ultimately led to a 47% reduction in error compared to the neighborhood baseline.



These scatter plots show Actual vs. Predicted Prices. The red dashed line represents a perfect prediction. The Linear Regression model (left) displays a scattered group of data points, indicating a higher error ($R^2 = 0.563$, RMSE = $94.61). In contrast, the Random Forest (right) demonstrates a much tighter grouping along the diagonal, which visually confirms its better accuracy ($R^2 = 0.720$, RMSE = $75.69).

Contributions

Pranavnath Bathula

Model Development & Evaluation:

- Baseline Model Framework: Established global and neighborhood mean baselines (RMSE: 143.09, 124.57), demonstrating location as a strong price predictor.
- Linear Regression: Built a preprocessing pipeline achieving RMSE 94.61, R^2 0.563 (24% improvement over baseline).
- Random Forest: Implemented 200-estimator ensemble achieving RMSE 75.69, R^2 0.720 (39% improvement over baseline, 20% over linear regression).

Feature Engineering:

- Feature Transformation: Engineered 26+ numeric features, including log transforms, availability ratios, and binary indicators.
- Engineered categorical features, including capacity and binary room types, and computed neighborhood-level statistics to capture geographic pricing patterns to get maximum model performance

Database Engineering:

- SQL Schema Design: Designed an initial normalized relational database schema for NYC (later extended by a teammate for multi-city).

Anish Jha

- **Database Engineering** - Designed and implemented a normalized relational database to replace static CSV files. This change enabled efficient storage and querying of listing data without memory overload.
- **ETL Pipeline Development** - Built the data extraction and cleaning logic. This transformed raw, untyped text, such as currency strings and dates, into usable numerical features for the model.
- **Model Optimization Research** - Led the investigation into regression techniques beyond simple Linear Regression such as Random Forest. The goal was to better capture complex pricing factors and improve model accuracy.
- **Data Integrity & Architecture** - Enforced strict typing within the SQL schema to ensure high-quality inputs. Also engineered a unified multi-city architecture to integrate data from NYC, Boston, and DC.

Yash Gudimalla

- **Feature Engineering** - Built out a full-featured multi-city feature set from joining together listing & neighbourhood tables, which includes capacity, availability, reviews, host experience, stats about the environment of the city, and property types.
- **Leakage Detection & Fix** - Used feature importances to identify features with price leakage (price per unit of capacity, neighbourhood price differences), deleted features with leakage, and created a new feature schema safe from leakage using only information available at the time of making the prediction.
- **Clean Pipeline & Dataset** - Removed outliers (price restrictions, reasonable maximums on capacity/reviews) and checked that each database row matches the corresponding CSV row, grouped features into categories, and created an engineered features CSV that can be reused.
- **Model Comparison & Tuning** - Compared to global averages and averages for each neighbourhood, trained model using the full pipeline to preprocess data, optimized hyperparameters for Random Forest, and improved RMSE/MAE/R2 on a cleaned dataset.

References

1. https://www.kaggle.com/datasets/stevezhenghp/airbnb-price-prediction
    a. Saw some examples of some features that we could expect from Inside Airbnb Data
2. https://arxiv.org/abs/2308.06929
    a. Compared our methods of Data exploration with “4.2 - Data Exploration.”
3. https://www.projectpro.io/article/how-data-science-increased-airbnbs-valuation-to-25-5-bn/199#mcetoc_1faeqq6p5f
    a. Some project idea inspiration on the real-world impacts our project can have