Toronto AirBnB Pricing Assistant (TAPA)

# Project Completion Report

## Project Summary

The **Toronto AirBnB Pricing Assistant (TAPA)** began as a data-driven tool aimed at helping AirBnB hosts and property investors optimize their rental pricing strategy in Toronto. Our completed project delivers on this core objective, but with several strategic pivots to improve performance and usability.

The final implementation provides:

* Price prediction based on property details and amenities
* Revenue estimation showing monthly earnings potential
* Occupancy forecasting to guide booking optimization
* Web-based interface for easy access and intuitive use

Live Demo Available at: <https://oceanpro00.github.io/toronto_airbnb_pricing_assistant/>

## Original vs. Final Implementation

### Original Plan

* **Data Sources**: Four Inside AirBnB datasets (listings, calendar, reviews, neighborhoods)
* **Model Approach**: Linear Regression for price prediction
* **Focus Areas**: Pricing, amenity value, stay duration strategy
* **Output Format**: Analysis notebooks with recommendation functions

### Final Implementation

* **Data Sources**: Listings data only (other datasets proved less valuable)
* **Model Approach**: XGBoost models for price, revenue, and occupancy prediction
* **Focus Areas**: Optimized for prediction accuracy and user experience
* **Output Format**: Interactive web application with S3-hosted models

## Key Pivots and Adaptations

### 1. Data Source Refinement

* **Original Plan**: Use all four Inside AirBnB datasets
* **Final Implementation**: Used only the listings dataset
* **Reason for Change**:
  + Calendar data contained forecasted rather than factual information, making it unreliable for ML training
  + Reviews data would have required NLP techniques beyond project scope
  + Geojson data was less necessary with our neighborhood standardization approach

### 2. Model Selection Evolution

* **Original Plan**: Linear Regression for interpretability
* **Final Implementation**: XGBoost models for optimal performance
* **Reason for Change**:
  + Linear Regression achieved only 40-50% R² scores
  + Our data showed complex, non-linear relationships between features
  + XGBoost captured these relationships much better (73-74% R²)

### 3. Feature Engineering Approach

* **Original Plan**: Standardize property types to 5 categories
* **Final Implementation**: Maintained data variance while cleaning outliers
* **Reason for Change**:
  + XGBoost performs better with more granular data
  + Retaining variety in property types improved model performance

### 4. Output Delivery Method

* **Original Plan**: Python functions and notebooks
* **Final Implementation**: Interactive web application
* **Reason for Change**:
  + Web interface makes tool accessible to non-technical users
  + Provides more intuitive, visual experience
  + Allows deployment via GitHub Pages for wider access

### 5. Model Scope Expansion

* **Original Plan**: Primarily price prediction
* **Final Implementation**: Price, revenue, and occupancy predictions
* **Reason for Change**:
  + Stakeholder value greatly increases with comprehensive predictions
  + Multi-factor approach provides more actionable insights

## Model Development Journey

Our model development went through several distinct phases:

### Phase 1: Linear Regression (Limited Success)

* Implemented multiple linear regression models
* Achieved R² scores of only 40-50%
* Identified key limitations with linear approach for complex pricing dynamics

### Phase 2: Random Forest Exploration

* Shifted to Random Forest models to capture non-linear relationships
* Improved performance to around 60-65% R²
* Still fell short of desired performance benchmark

### Phase 3: XGBoost Implementation (Success)

* Implemented XGBoost with extensive hyperparameter tuning
* Focused on maintaining data variance rather than extreme feature engineering
* Achieved R² scores of 73-74% for price prediction

### Phase 4: Advanced Techniques (Mixed Results)

* Attempted PCA for dimensionality reduction
* Applied SHAP for feature importance analysis
* Surprisingly, these advanced techniques slightly reduced model accuracy

### Phase 5: Baseline Protection Enhancement

* Implemented baseline price protection using 10% below 10th percentile pricing
* Created fallback mechanism for underrepresented property types
* Enhanced prediction reliability for edge cases

## Technical Challenges and Solutions

### Challenge 1: Data Variance and Complexity

* **Problem**: Extreme variance in Toronto AirBnB pricing (ranging from $30 to $1000+ per night)
* **Solution**:
  + Implemented log transformation of price target
  + Limited outliers while preserving meaningful variance
  + Leveraged XGBoost's ability to handle non-linear relationships

### Challenge 2: Amenity Processing

* **Problem**: Amenities stored as JSON arrays with inconsistent naming
* **Solution**:
  + Extracted and standardized amenity names
  + Created binary feature matrix for model input
  + Grouped similar amenities to reduce dimensionality

### Challenge 3: Feature Selection

* **Problem**: Too many features causing overfitting
* **Solution**:
  + Used SHAP analysis to identify most important features
  + Created interaction terms for key variables (e.g., bed\_bath\_interaction)
  + Implemented log transformations for skewed features

### Challenge 4: Model Deployment

* **Problem**: Making models accessible to users without Python knowledge
* **Solution**:
  + Created web interface with HTML/CSS/JavaScript
  + Stored serialized models in Amazon S3 bucket
  + Implemented GitHub Pages deployment for easy access

## Results and Performance

### Final Model Performance

| **Model** | **R² Score** | **RMSE** | **Key Features** |
| --- | --- | --- | --- |
| Price Prediction | 73.25% | 0.35 | Property type, location, amenities |
| Revenue Prediction | 72.74% | 0.58 | Price, location, occupancy factors |
| Occupancy Prediction | 65.47% | 0.52 | Property characteristics, amenities |
|  |  |  |  |

### Performance Context

While we initially targeted R² scores of 80%, our achieved results of 73-74% represent strong performance given:

1. The inherent variability in short-term rental pricing
2. Seasonal and event-based pricing fluctuations not captured in the data
3. Subjective elements like decor quality and photography
4. Host-specific pricing strategies beyond property characteristics

The model demonstrates strong predictive power for practical use cases, with baseline pricing protection ensuring recommendations remain market-competitive.

## Frontend Implementation

To maximize accessibility and usability, we developed a web-based interface:

* **Technology Stack**: HTML, CSS, JavaScript
* **Hosting**: GitHub Pages
* **Model Access**: Amazon S3 bucket for serialized models
* **Key Features**:
  + Neighborhood and property type selection
  + Detailed amenity selection interface
  + Responsive design for desktop and mobile
  + Clear visualization of predictions

The frontend implements the baseline pricing protection system to ensure predictions are always reasonable, even for edge cases or unusual property combinations.

## Lessons Learned

### Data Insights

1. **Neighborhood dominance**: Location proved to be the strongest predictor of price
2. **Amenity impact**: Certain amenities (hot tub, pool, parking) have outsized price effects
3. **Size matters**: Bedroom and bathroom counts are more important than total guest capacity

### Technical Insights

1. **Model selection matters**: The right algorithm made a substantial difference in performance
2. **Data variance vs. standardization**: For XGBoost, preserving data variance was more valuable than extensive standardization
3. **Feature interactions**: Creating interaction terms (e.g., bed\_bath\_interaction) improved model performance

### Process Insights

1. **Staged modeling approach**: Progressive model refinement allowed us to learn what worked
2. **Data cleaning balance**: Removing true outliers while preserving natural variance was key
3. **Baseline protection**: Implementing minimum price thresholds improved recommendation quality

## Future Improvements

With additional time and resources, we would enhance TAPA with:

1. **Seasonal pricing models**: Develop separate models for high/low seasons
2. **Geographic microanalysis**: Create more granular neighborhood clusters
3. **NLP for reviews**: Incorporate sentiment analysis from reviews
4. **Time series forecasting**: Use calendar data to predict optimal pricing by date
5. **Competitive pricing**: Add competitor density and pricing analysis
6. **Image analysis**: Incorporate property image quality assessment

Report prepared by: Sean Schallberger, Bryan Carney, Jitesh Makan  
University of Toronto Data Analytics Bootcamp, 2025