STEVENS
INSTITUTE OF TECHNOLOGY
1870

World Real Estate Market Prediction with Machine Learning Algorithms

AML Project - AAI 559

Group Members :
Rajanee Shenkar
Samradnyee Kudalkar
Karina Fayn

Table of Contents

Introduction

- Data-driven predictive modeling is crucial in today's world
- Real estate price prediction is a complex challenge
- **Goal:** Develop robust machine learning frameworks for accurate predictions

Key Challenges

- Multi-dimensional nature of real estate data
- Balancing computational efficiency and predictive accuracy
- Capturing complex, non-linear relationships

Problem Statement

Objective
Develop a predictive model for real estate prices using multiple machine learning algorithms

Approach
- Utilize diverse algorithms to balance:
  - Interpretability
  - Predictive power
  - Computational efficiency

Algorithms Explored
- Linear Regression (Baseline)
- Random Forest
- Neural Network Regression
- Stacking Regression

Dataset Overview

Source: Kaggle real estate dataset (global listings).

- **Key Features:**
  - apartment_total_area (size of property).
  - price_in_USD (target variable).
  - country, location, number_of_rooms, and amenities.
  - Derived Feature: price_per_m2 for normalized comparison.
- **Dataset Size:** 147,000 rows.
- **Importance:** Provides diversity in property attributes and pricing across regions.

Data Preprocessing

- **Handling Missing Data:**
  Rows with missing values were dropped to ensure clean data.
- **Categorical Encoding:**
  - One-hot encoding for `country` and `location` columns.
- **Unit Conversion:**
  - Converted `apartment_total_area` from string to numeric by stripping units ("m²").
- **Outlier Handling:**
  - Capped extreme values using the 5th and 95th percentiles.
- **Feature Scaling:**
  - StandardScaler was applied to numerical features to normalize data.
- **Feature Engineering:**
  - Created `price_per_m2` as a derived feature to analyze price relative to area.

Model Selection and Rationale

1. **Linear Regression:**
   - Baseline model for capturing linear relationships.
   - **Limitation:** Cannot model complex non-linear patterns.
2. **Random Forest Regressor:**
   - Ensemble model that handles non-linear relationships.
   - Reduces overfitting through aggregation of decision trees.
3. **Neural Network (MLP):**
   - Captures intricate patterns using multi-layer perceptron architecture.
   - Activation Function: ReLU; Optimizer: Adam.
   - **Limitation:** Requires extensive tuning and less interpretable.
4. **Stacking Regressor:**
   - Combines Random Forest and Neural Network predictions using a meta-model (Random Forest).
   - Balances bias, variance, and predictive power.

Feature Importance Analysis

- **Random Forest Feature Importance:**
  - `price_per_m2` emerged as the most influential predictor (importance score: 0.658).
  - `apartment_total_area` was the second most significant feature (score: 0.328).
  - `apartment_living_area` contributed minimally (score: 0.005).
- **Insight:** Price per square meter is a consistent predictor across regions.

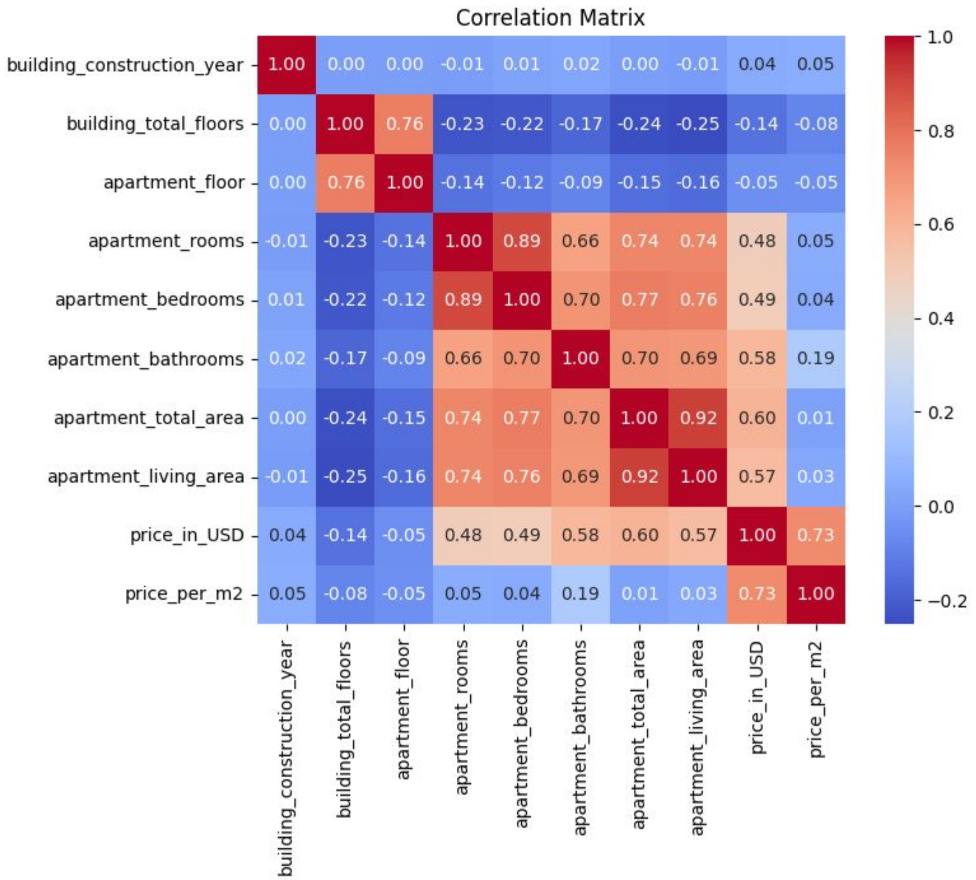
Exploratory Data Analysis (EDA)

Correlation Analysis:

Heatmap revealed high positive correlation between:

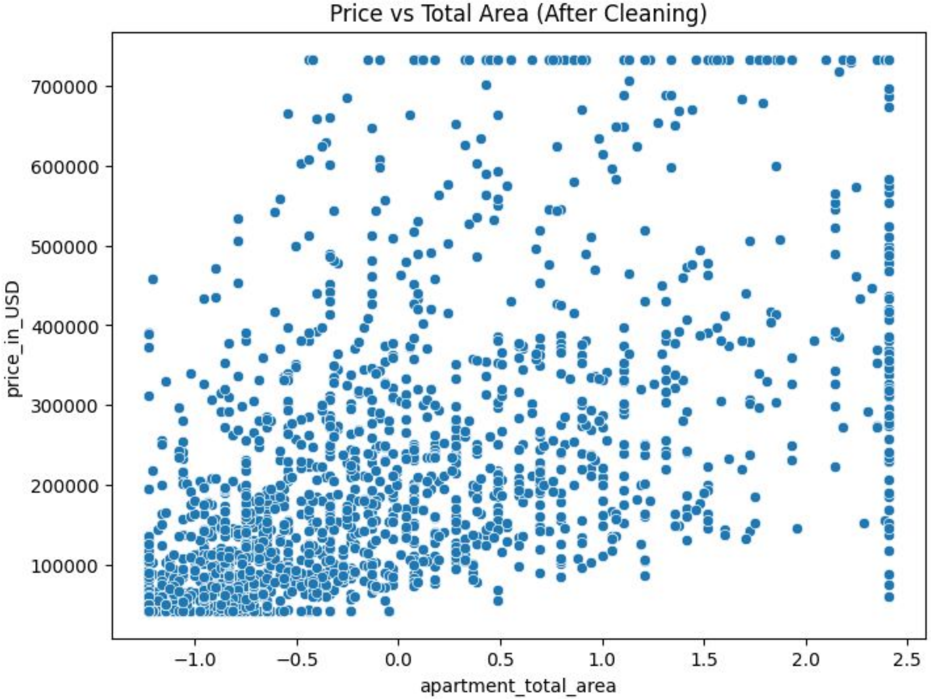
-price_in_USD and price_per_m2.

-price_in_USD and apartment_total_area.



Exploratory Data Analysis (EDA)

Scatterplot: `apartment_total_area` vs. `price_in_USD`.



Models Implemented
Linear Regression

- **Theory:** Models the relationship between input features and target using:
- **Assumption:** Linear relationship between features and target.

The linear regression equation is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

- **Result**
  - High MSE and negative R^2 score indicate poor performance.
  - **Limitation:** Unable to handle complex, non-linear relationships.

Random Forest Regression

- **Theory:** Ensemble of decision trees:
  - Combines predictions of multiple trees to improve accuracy and reduce overfitting.

  Mathematical Formula

  $$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} T_i(x)$$

- **Advantages:**
  - Handles non-linear relationships.
  - Provides feature importance.
- **Performance:**
  - MSE: 1.26e+08; R^2: 0.9963.
  - Close alignment between predicted and actual values
- **Tuned Hyperparameters to get an optimised Random forest model**

Neural Network

- **Theory:** Multi-Layer Perceptron (MLP) with ReLU activation:

  The input-output relationship in a Neural Network is described by:

$$h_j = \sigma\left(\sum_i w_{ij} x_i + b_j\right)$$

- **Architecture:**
  - Input Layer → Hidden Layers → Output Layer.
  - Optimizer: Adam; Activation: ReLU.
- **Performance:**
  - Moderate fit with $R^2 = 0.8602$.
  - Deviations observed for extreme values.

Stacked Model

- **Theory:** Combines multiple base models (Random Forest + Neural Network):
- **Meta-Model:** Random Forest synthesizes predictions from base models.
  Overall Prediction is given by:

$$\hat{y}_{stacked} = \mathrm{MetaModel}\left(\hat{y}_1, \hat{y}_2\right)$$

- **Advantages:**
  - Reduces bias and variance.
  - Combines strengths of individual models.
- **Performance:**
  - MSE: 1.46e+08; R^2: 0.9957.

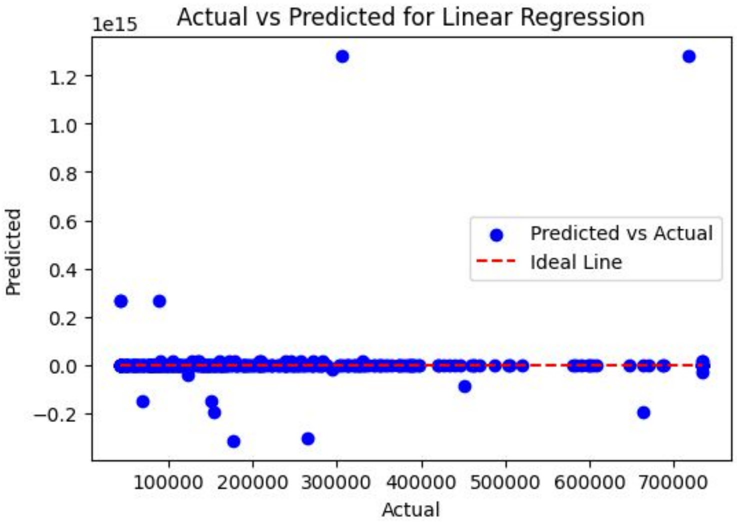
Model Evaluation Comparison

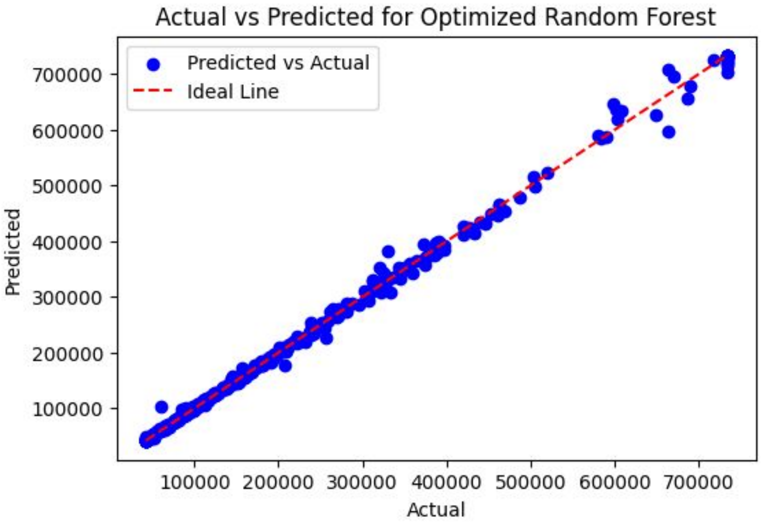
- **Performance Metrics:**
    - MSE, MAE, R^2 scores for all models.
- **Results Table:**

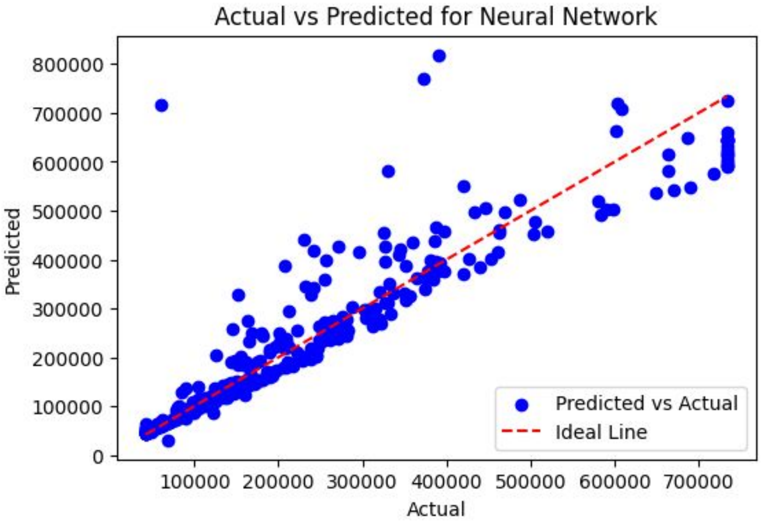
Model	MSE	MAE	R^2
Linear Regression	High	High	Negative
Random Forest	1.26e+08	5950.38	0.9963
Optimized Random Forest	8.92e+07	4898.25	0.9974
Neural Network	4.76e+09	34726.4	0.8602
Stacking Regressor	1.46e+08	5921.3	0.9957

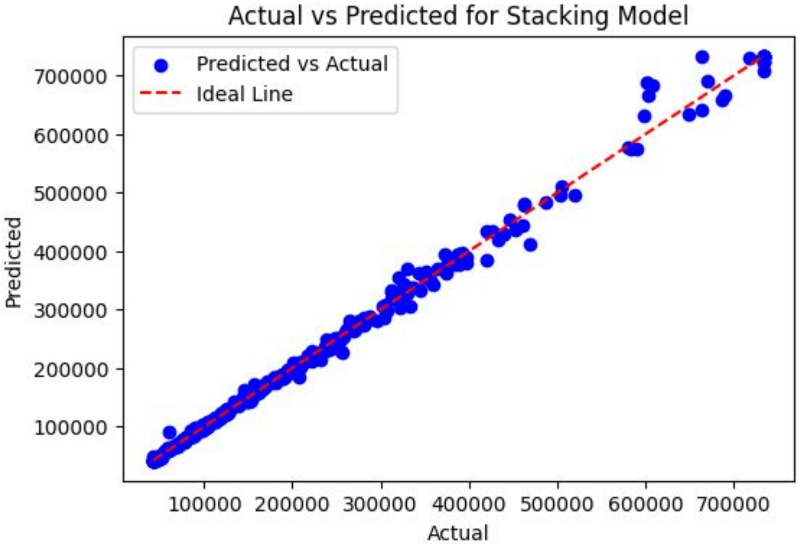
Results

1. **Prediction vs Actual Values:**
    - Linear Regression: Large deviations.
    - Random Forest: Close alignment.
    - Neural Network: Moderate deviations.
    - Stacking Model: Minimal errors.


Actual vs Predicted for Linear Regression
1e15
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Predicted
Predicted vs Actual
Ideal Line
100000 200000 300000 400000 500000 600000 700000
Actual


Actual vs Predicted for Optimized Random Forest
Predicted vs Actual
Ideal Line
700000
600000
500000
400000
300000
200000
100000
Predicted
100000 200000 300000 400000 500000 600000 700000
Actual


Actual vs Predicted for Neural Network
800000
700000
600000
500000
400000
300000
200000
100000
0
Predicted
Predicted vs Actual
Ideal Line
100000 200000 300000 400000 500000 600000 700000
Actual


Actual vs Predicted for Stacking Model
Predicted vs Actual
Ideal Line
700000
600000
500000
400000
300000
200000
100000
Predicted
100000 200000 300000 400000 500000 600000 700000
Actual

Insights

- **Best Model:** Optimized Random Forest achieved the lowest error and highest R^2.
- **Stacking Model:** Provided competitive performance, balancing model strengths.
- **Linear Regression:** Failed to capture non-linear relationships.
- **Neural Networks:** Required extensive tuning but underperformed.

Conclusion and Future Scope

- Developed robust machine learning frameworks
- Demonstrated effectiveness of ensemble methods
- Achieved high prediction accuracy for real estate prices

Future Scope

- Explore more complex ensemble techniques
- Incorporate additional features
- Expand dataset diversity