Real Estate Alpha Calculator:
A Tool for Assessing Risk-Adjusted Returns in Residential Real Estate Investing

David Latortue

Abstract

This paper presents the Real Estate Alpha Calculator, a tool designed to aid real estate investors by providing a quantitative assessment of investment opportunities. By calculating an "alpha" value for properties in Montreal, the tool helps investors evaluate risk-adjusted returns. The calculator leverages machine learning models for price prediction and adapts the Capital Asset Pricing Model (CAPM) to real estate, offering a data-driven approach to streamline investment decision-making. Our results demonstrate its effectiveness in comparing properties and identifying high-potential investments while minimizing systemic risks.

1 Introduction

Real estate investment involves navigating a competitive landscape where identifying profitable opportunities is challenging. The idiosyncratic nature of properties, combined with complex, obscure or incomplete market information, makes screening potential investments both costly and time-consuming. Currently, investors lack an efficient solution for evaluating relative returns while properly accounting for associated risks, hindering their ability to effectively filter and prioritize investment opportunities.

This paper introduces the *Real Estate Alpha Calculator*, a screening tool designed to help investors assess the expected returns and systemic risks of potential real estate investments in Montreal. The calculator allows investors to systematically evaluate and compare prospective properties, identifying those offering the best risk-adjusted potential returns. This enables investors to quickly focus on a targeted set of high-potential properties, saving them from the burden of time-consuming extensive market research.
To achieve this, we implemented the following steps:

1. Developed comprehensive neighborhood profiles to capture factors influencing property values.
2. Compared properties within similar categories to ensure fair evaluation.
3. Assigned expected return and risk ratings, enabling a quantitative assessment of investment potential.

Our tool computes an ***alpha*** (α) value, to quantify the excess returns of a given property, bridging financial modeling and practical real estate investment decisions. ***Alpha*** measures an asset's potential performance relative to its peers, offering a data-driven approach to initial screening, focusing on systemic risk factors.

Paper Structure

The remainder of this paper is organized as follows:

- **Background:** Theoretical foundations in finance and machine learning
- **Methodology:** Data processing, risk calculation, price prediction, and alpha computation
- **Results:** Model outcomes, tool demonstration, and statistical analysis

2 Background: A little bit of theory

2.1 Alpha (α)

Alpha (which we will refer to as α) is a measure used in finance to assess the performance of an investment relative to the market it is part of. α tells you if the actual return of the target investment is above or below where it should be, relative to the level of systematic risk involved (estimated by its beta) and the overall market conditions[11]. Here is how alpha is calculated :

$$\alpha = R_i - [R_f + \beta_i(R_m - R_f)] \tag{1}$$

where:

- R_i is the actual return of the investment.
- R_f is the risk-free rate (more on this later).
- β_i is the beta of the investment, which measures its volatility relative to the market (also more on that later).
- R_m is the expected return of the market.

2.1.1 Interpretation of α

Once α is calculated, you will obtain a real value that can be either positive or negative. Here's how to interpret it:

- **Positive Alpha** ($\alpha > 0$): A positive alpha indicates that the investment has outperformed the market on a risk-adjusted basis. It suggests that the investment generated returns higher than expected given its level of risk.
- **Zero Alpha** ($\alpha = 0$): A zero alpha means that the investment has performed exactly as expected based on its risk level. In other words, it neither outperformed nor underperformed the market. In practice, to determine if an investment aligns with market expectations, we set bounds around zero.
- **Negative Alpha** ($\alpha < 0$): A negative alpha suggests that the investment has underperformed the market on a risk-adjusted basis. The investment's returns were lower than expected given its risk, indicating poor performance relative to the benchmark.

2.2 Risk (β)

Beta is a measure of risk of an asset relative to the market. The measure encapsulates the relative volatility of the asset compared to the market. Volatility is simply the variability in the pricing of an asset [12]. The higher the gaps in the high and lows of the pricing of an asset, the more volatile it is. To measure the risk, we can first compute the standard deviation (often referred to as σ) in the price history of the asset and the one its market as such :

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(R_i - \bar{R})^2} \tag{2}$$

where:

- σ is the volatility (standard deviation of returns).
- N is the number of periods (e.g., days, months, years).
- R_i is the return of the asset in period i.
- $\bar{R}$ is the average return of the asset over N periods.

Then, we need to know if the asset and the market are correlated in any way and if the move in the same direction or opposites. The **correlation** between two variables X and Y is calculated as:

$$Corr(X, Y) = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2} \cdot \sqrt{\sum(Y_i - \bar{Y})^2}}$$

Where:

- X_i and Y_i are the individual data points of X and Y.
- $\bar{X}$ and $\bar{Y}$ are the means (averages) of X and Y, respectively.
- $\sum$ denotes summation over all data points.

This output of *Corr* is a value between -1 and 1, its interpretation can be done as follow :

- A value of 1 indicates a perfect positive linear relationship.
- A value of -1 indicates a perfect negative linear relationship.
- A value of 0 indicates no linear relationship between the variables.

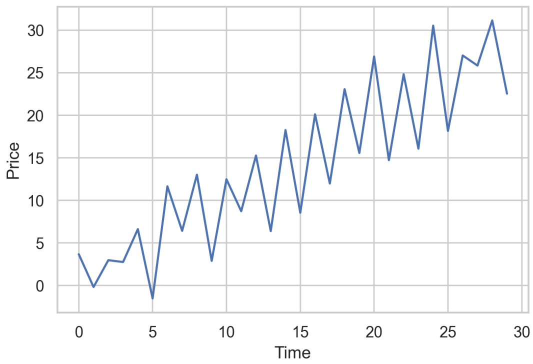
Finally, to calculate β given the price history of the asset A and the market M we simply do the following :

$$\beta(A, M) = Corr(A, M) \cdot \frac{\sigma(P)}{\sigma(M)}$$

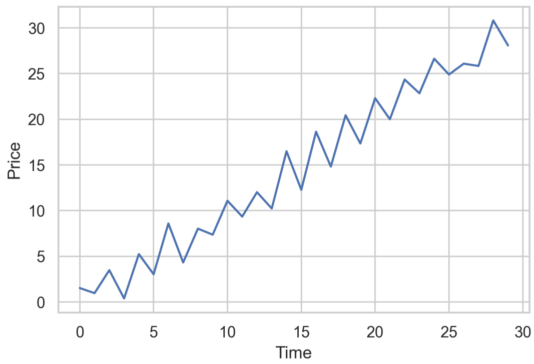
Ideally, you would want the lowest risk possible on an investment. However, in finance, there is a fundamental trade-off between risk and reward: higher risks means higher returns.

Interpretation of β

- $\beta > 1$: More volatile than the market (higher risk, potentially higher returns)
- $\beta = 1$: Same risk level as the market
- $0 < \beta < 1$: Lower risk than the market (steady returns)
- $\beta = 0$: Uncorrelated with the market
- $\beta < 0$: Negatively correlated with the market



(a) High volatility in Pricing example



(b) Low volatility in Pricing example

Figure 1: Comparison of volatility in pricing examples.

2.2.1 Risk-Free rate

The risk-free rate is the theoretical return an investor can expect from an investment with no risk of financial loss. It represents the minimum return required for an investment that carries no risk of default, credit, or inflation impact. In practice, the risk-free rate is typically represented by the yield on short-term government-issued securities, such as Canadian Treasury bills (T-bills), because these are backed by the Canadian government's full faith and credit, making them highly secure and virtually free from default risk. If an investment opportunity has any risk but offers returns lower than the risk-free rate, it's generally wise to avoid it, as you're not being adequately compensated for taking on additional risk. In our case, we will refer to the policy interest rate from Bank Of Canada : Bank of Canada Key Interest Rate.

2.3 Alpha (α) for Real Estate (RE)

Real estate offers various investment opportunities, including Real Estate Investment Trusts (REITs), house flipping, and commercial properties. This discussion will focus on buying and selling residential properties. In this domain, investors use several metrics to evaluate different aspects of a real estate investment. Here are a few key metrics:

- **Net Operating Income (NOI)**: A measure of profitability of an investment, calculated as Gross Operating Income (revenues) − Operating Expenses.
- **Cash on Cash Return**: The ratio of annual pre-tax cash flow to total cash invested, calculated as $\frac{\text{Annual Pre-Tax Cash Flow}}{\text{Total Cash Invested}}$.
- **Cap Rate**: The capitalization rate, calculated as $\frac{\text{NOI}}{\text{Property Value}}$.
- **Purchase Price**: The price paid for the property.
- **Property Value**: The market value of the property.
- **Internal Rate of Return (IRR)**: An important metric not specific to real estate, representing the annualized rate of return.

Traditionally, alpha is used in public financial markets, but it can also be applied to real estate. To adapt alpha for real estate, we must address several considerations. First, individual properties often lack significant price histories because homeowners and residential real estate investors typically hold assets for long periods, making buying and selling slow and costly (legal fees, inspections, Insurance, taxes, Appraisal, etc). To overcome this challenge, we propose defining criteria to identify properties similar to a given one, thereby establishing a comparative price history. We refer to this definition as the **Property Class (P_C)**. The definition is shown in the following formula :

$$P_C = \{p_i \mid p_i \in \text{Prices}, \text{Market}_{\text{Columns}}[i] = \text{Criteria}\} \quad (3)$$

Where:

- P_C: Set of property prices from the market that match the specified criteria.
- Prices: Vector of all property prices available in the market.
- Market: Matrix where each row represents a property and each column represents a feature (e.g., 'neighborhood', 'propertTtype', etc.).
- Columns: List of columns in Market used to filter properties, corresponding to ['neighborhood', 'propertyType', 'totalUnits', 'residentialUnits', 'businessUnits'].
- Criteria: Vector of specific values for the columns, representing the desired attributes of the property.
- i: Index of properties in Market where the columns match the criteria.

For a given property P, we find the matches of its property class P_C and use them as price history to estimate the risk of our class β_C. This allows us to circumvent the statistical limitations that come from the low observed volatility and limited price history of each individual residential real estate asset. We provide more details on this methodology in section 3.
The formula β of property P_C against the market M_C can be expressed as follow:

$$\beta_C = \beta(P_C, M_C)$$

We opt to use a machine learning approach to estimate the expected returns from a property. More on this will be detailed in section 2.4.

2.3.1 Final formula α

To apply α to real estate, we need to define the actual return on investment R_i and the expected market return R_m. For R_i, we can use metrics such as the capitalization rate (cap rate), internal rate of return (IRR), or cash-on-cash return. We choose the cap rate due to its simplicity and its independence from an individual's financing strategy. For R_m, we utilize machine learning models trained on our *Property Sales Listing* and *Renting* datasets. These models, denoted as $f_p(P)$ and $f_r(P)$ respectively, where P represents the property, predict the property prices and rental incomes to estimate the expected return. Thus, the final formula we will use is:

$$\alpha_{RE} = \frac{NOI}{PurchasePrice} - [R_f + \beta_C(\frac{f_r(P)}{f_p(P)} - R_f)] \quad (4)$$

The methodology section will show details of the algorithm used in practice to find α.

2.4 Machine Learning Concepts

As mentioned in the previous section, to estimate the expected rent and property value of a property in the market (V_p and V_r), we propose training a machine learning model for regression on data from various listing from the web. The estimates of the values of rent and property sales of the property P are expressed as following for $f_p(P)$ and $f_r(P)$:

$$V = f(P) \tag{5}$$

In this subsection, we explore various machine learning concepts and models that are widely used for regression tasks, particularly when dealing with tabular data. The models range from traditional non-deep learning estimators to modern deep learning approaches, including transformers specifically tailored for tabular data.

2.4.1 Non-Deep Learning Estimators

Gradient Boosting Models
These models build trees sequentially, with each new tree correcting errors made by the previous ones. They combine weak learners (shallow trees) to create a strong predictor, optimizing a loss function at each step [6]. We trained the following models from Scikit-Learn : Gradient Boosting Regressor, CatBoost[15], LightGBM[13], XGBoost[3] and NGBoost[5].

Random Forest
Random Forest creates multiple decision trees using random subsets of features and data samples [1]. It then aggregates predictions from all trees, typically using majority voting for classification or averaging for regression. We trained RandomForest from Scikit-Learn.

Linear Models
These models assume a linear relationship between features and the target variable. They fit a linear function to the data, with different regularization techniques to prevent overfitting:

- **Ridge [9]:** Uses L2 regularization
- **Lasso [18]:** Uses L1 regularization
- **ElasticNet [21]:** Combines L1 and L2 regularization

Support Vector Machines
Support Vector Regressor (SVR) works by finding the hyperplane that maximizes the margin between data points[4]. It can handle non-linear relationships by using kernel functions to transform the input space.

2.4.2 Deep Learning Models for Tabular Data

Transformers: Originally developed for Natural Language Processing (NLP) tasks, transformers have been adapted for tabular data[19]. They are adept at learning complex dependencies and feature interactions but typically require substantial amounts of data and careful tuning.

Tab Transformer: The Tab Transformer is a specialized variant of the transformer model designed for tabular data[10]. It efficiently captures feature interactions without requiring extensive preprocessing, making it particularly effective for datasets with categorical variables.

FT Transformer: The FT Transformer builds upon the Tab Transformer by extending its capability to handle both categorical and numerical features [7]. Instead of encoding only the categorical variables, the FT Transformer incorporates numerical features directly into the transformer architecture.

ResNet: ResNet is a powerful Convolutional Neural Network that has proven to be particularly effective in computer vision tasks[8]. However, researchers have shown that it could also perform well for tabular tasks[20].

2.4.3 Ensembling Techniques

Ensembling techniques are commonly used to improve model performance by combining predictions from multiple models. Key ensembling methods include:

- **Stacking and Blending:** Stacking involves combining different types of models, such as linear models with tree-based models, to leverage their complementary strengths.
- **Voting:** Voting methods aggregate the predictions from multiple models to make a final prediction. Common strategies include majority voting for classification tasks and averaging for regression tasks.
- **Averaging/ Weighted Averaging:** Averaging involves combining predictions from multiple models by calculating their average. This technique is often used in ensemble methods to improve the robustness of predictions. Additionally, the contribution of each models can be adjusted to increase performance.

While traditional machine learning models often outperform deep learning methods on tabular data, exploring deep learning approaches, particularly transformer-based models, can be valuable. This exploration is especially pertinent given the increasing complexity of tabular datasets and the potential of transformers to capture intricate feature interactions.

3 Methodology

3.1 Data Collection

The foundation of our Real Estate Alpha Calculator is a comprehensive dataset of property sales and apartment rentals across Quebec. This section details our data collection process, sources, and the scope of our initial dataset.

3.1.1 Data Sources

We employed web scraping techniques to gather data from multiple prominent real estate platforms in Quebec. The primary sources for our data collection were:

- LesPacs
- Duproprio
- Centris
- Kijiji

These platforms were chosen for their extensive coverage of the Quebec real estate market, ensuring a diverse and representative dataset.

3.1.2 Web Scraping Methodology

To efficiently collect data from these sources, we utilized Scrapy, a powerful and flexible web scraping framework. Our scraping process began in 2020, allowing us to accumulate a substantial amount of historical data.

3.1.3 Data Fields

Our data collection focused on two main categories: property sales and apartment rentals. For each category, we collected a comprehensive set of fields to provide a detailed picture of each listing.

Property Sales Data Fields
For property sales, we collected the following 23 fields:

```
'property_id', 'source', 'address', 'neighborhood', 'price', 'property_type',
'usage', 'construction_date', 'building_configuration', 'potential_gross_revenues',
'building_style', 'no_of_units', 'lots_area_in_sqr_ft', 'no_of_parkings',
'parking_type', 'description', 'title', 'url', 'creation_date', 'has_pool',
'pool_type', 'fireplace_or_stove_type', 'adapted_for_reduced_mobility',
'elevator', 'close_to_body_of_water', 'price_in_dollars'
```

Apartment Rentals Data Fields
For apartment rentals, we collected the following 26 fields:

```
neighborhood, no_of_rooms, no_of_bathrooms, category, electricity_included,
heating_included, wifi_included, parking_included, animal_friendly,
laundry_in_unit, air_conditioner_included, private_exterior_spaces_included,
smoking_allowed_included, furnished, lease_duration_in_months,
details_from_description, water_included, tv_included, laundry_in_building,
dishwasher, fridge, gym, pool, concierge, security24hrs, bicycle_parking,
storage_space, elevator, area
```

3.1.4 Dataset Size

Our initial dataset, prior to preprocessing, comprised:

- Property Sales: 159,346 listings
- Apartment Rentals: 59,242 listings

This substantial dataset provides a robust foundation for our analysis, offering a comprehensive view of the Quebec real estate market over time.

3.1.5 Data Collection Challenges and Limitations

While our data collection process was extensive, it's important to note some potential limitations:

- Data availability is limited to listings posted from 2020 onward.
- The accuracy of the data depends on the information provided by sellers and landlords on the source websites.
- There may be some inconsistencies or missing data across different platforms due to varying listing formats and requirements.

These limitations were addressed during our data preprocessing and cleaning stages, which will be discussed in the subsequent section.

3.2 Data Preprocessing

The raw data collected from various real estate platforms underwent a rigorous preprocessing pipeline to ensure its quality, consistency, and analytical value. This process was crucial for both the property sales dataset (159,346 listings) and the rental dataset (59,242 listings).
Our preprocessing pipeline consisted of the following key steps:

1. **Geographical Filtering:** We filtered listings to include only properties within Montreal, utilizing address and neighborhood information. This step ensured our analysis remained focused on our target market.

2. **Neighborhood Standardization:** We developed and applied a custom Montreal neighborhood classification system. This standardized location data across listings, allowing for consistent geographical analysis.

3. **Property Type Categorization:** Properties were classified into standardized categories (e.g., Apartment, Condo, Duplex/Triplex). This categorization facilitates market segment analysis and ensures comparability across listings.

4. **Feature Engineering:** We transformed raw data fields into analytically useful features. This process included:
    - Extracting numerical data from text descriptions
    - Standardizing area measurements to square feet
    - Creating binary indicators for amenities (e.g., parking, pool, elevator)
    - Deriving additional features such as property age and unit counts

5. **Data Cleaning:** We identified and removed outliers and erroneous entries using domain-specific heuristics. This step ensured data integrity and removed potentially misleading datapoints.

6. **Missing Value Handling:** Where appropriate, we estimated missing values using data from similar properties. This approach enhanced dataset completeness while maintaining data integrity.

7. **Data Transformation:** Certain numerical fields underwent Box-Cox transformation to address skewness, improving their suitability for subsequent statistical analyses.

8. **Encoding Categorical Variables:** We encoded categorical variables numerically to prepare them for machine learning algorithms.

This comprehensive preprocessing approach transformed our raw data into a robust, clean, and analytically rich dataset. It addresses the complexities inherent in real estate data, such as diverse property types, inconsistent neighborhood information, and the need to derive meaningful features from raw listings. The resulting dataset provides a solid foundation for our subsequent analyses and the development of the Real Estate Alpha Calculator.

3.3 Overview of Montreal's Real Estate Market

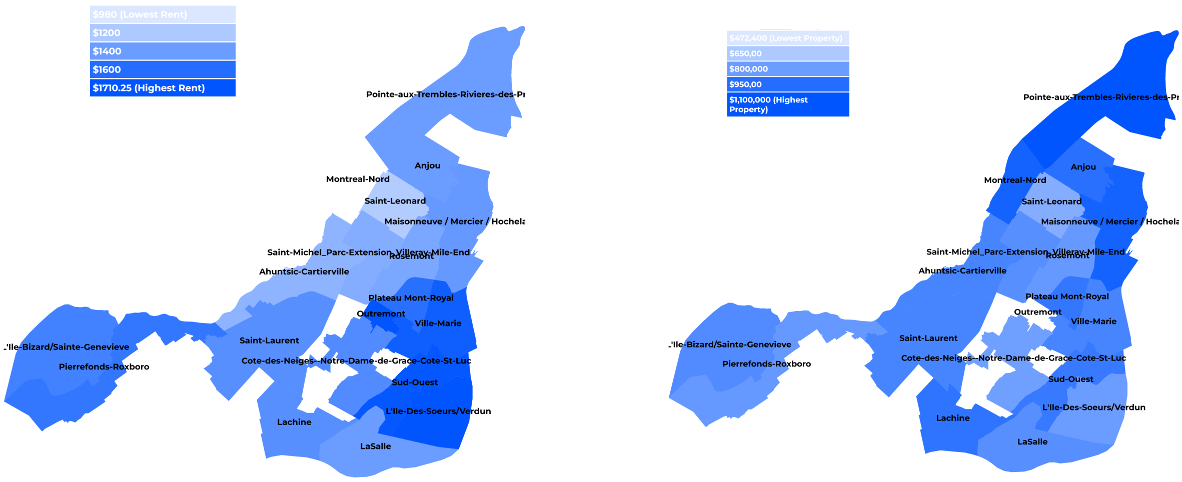
Montreal, a vibrant multicultural city of 1.7 million residents, has emerged as a significant player in Canada's real estate market. Known for its rich culture, delicious cuisine, and growing reputation as an AI hub, Montreal attracts a diverse population of residents and investors alike. As of September 2024, Montreal is considered the most affordable metropolis in Canada, according to CanadIm [2]. The city currently experiences a seller's market, with a sales-to-new listings ratio of 62% in August 2024 [14]. Recent data shows median price increases across various property types compared to the previous year:

- Single-family homes: 5.2% increase
- Condominiums: 3.6% increase
- Plexes: 6% increase

However, with interest rates at 6.45% as of September 2024, significantly higher than the near-zero rates seen during the pandemic (2020-2021) [16], many prospective buyers express concerns about affordability.
In the following subsections, we will analyze the Montreal real estate market using our data, providing insights to enhance the reader's understanding of various factors that may impact the performance of our Alpha Calculator.

3.3.1 Neighborhood Analysis

Price Distribution Across Neighborhoods
Our analysis reveals distinct patterns in both rental and property sales prices across Montreal's neighborhoods:



(a) Average Renting

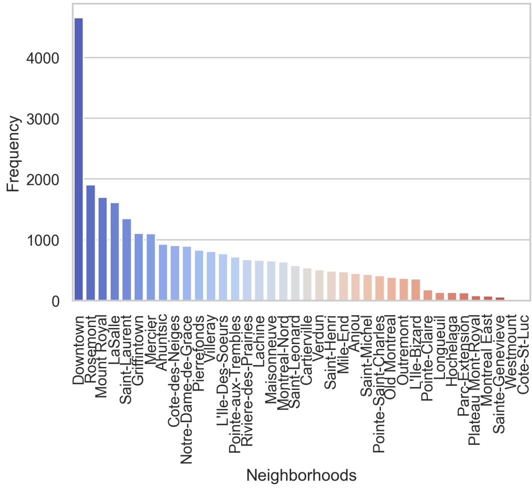
(b) Average Property Sales Price

Figure 2: Average Rent (a) and Property Sales Price (b) in each neighborhoods.

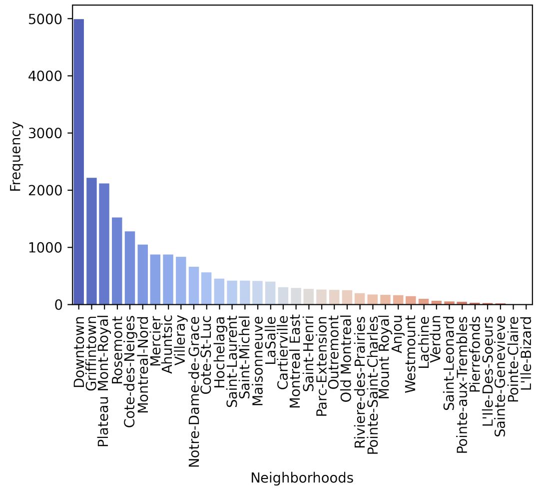
- Saint-Leonard emerges as the most affordable area for both renting and buying.
- Central areas (e.g., Ville Marie, Griffintown, Outremont) show higher rental prices relative to purchase prices, possibly due to the scarcity of single-family homes and plexes in these locations; Consequently favoring the sale of condos, which tend to be cheaper.
- Northeastern areas, with limited metro access[17], tend to have lower rental prices but higher property purchase prices.

It's important to note that, in this representation, some of our data is aggregated by larger districts in this map, which may obscure nuances in smaller neighborhoods, particularly in areas like Mercier-Hochelaga-Maisonneuve, L'Île-des-Sœurs-Verdun and Saint-Michel-Parc-Extension-Villeray-Mile-End.

Data Frequency by Neighborhood
The dataset shows varying levels of representation across neighborhoods:
Rental Market:



(a) Property Sales



(b) Rental

Figure 3: Data amounts per neighborhoods.

- Low representation (< 300 data points): Rivière-des-Prairies, Pointe-Saint-Charles, Anjou, Lachine, Saint-Leonard, Verdun, Pierrefonds, L'Île-des-Sœurs, L'Île-Bizard-Sainte-Geneviève
- High representation: Griffintown, Côte-des-Neiges, downtown area

Property Sales:

- Higher representation (> 500 data points): Saint-Léonard, Anjou
- Lower representation (< 300 data points): Sainte-Geneviève
- Highest representation: Downtown area, Rosemont, Griffintown, Plateau-Mont-Royal

This distribution suggests that areas with better public transportation and proximity to downtown have more active real estate markets, potentially indicating higher population density or more frequent tenant turnover.

Property Type Distributions

Properties from our dataset belong to either ones of theses property types : Condo', 'House', 'Multiplex', 'Lot', 'Commercial'. The following chart4 gives a the proportion of the how each types share the market. As suggested earlier, our intuition was that Condos were more common in Central Areas. To verify this assumption, we have plotted property types across our data and across certain neighborhoods. At first glance at 5 we can safely confirm our hypothesis.

To better see the disitrbitions in different areas, in the next graph we has grouped the distributions by area.

An another insight that we notice from this representation is that Condos seems to be less popular in areas the East and West extremes of the Island (Pointe-aux-Trembles-Riviere-des-Prairies and Pierrefonds-LÎle-Bizard-Pointe-Claire).

3.3.2 Price Distributions

Analysis of price distributions reveals that:

Both distributions are normal, but property sales show a wider range compared to rentals. The rental market appears more accommodating with a tighter price range, suggesting greater affordability and flexibility in Montreal's rental sector compared to property purchases.

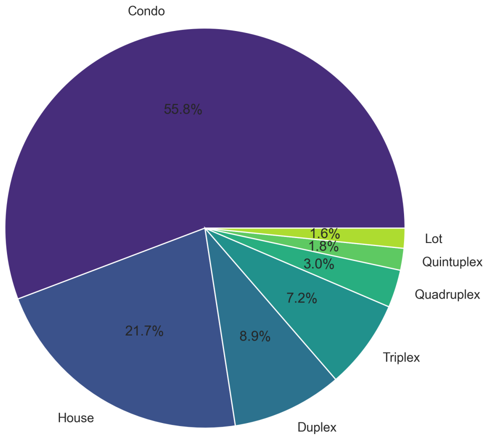


Figure 4: Property types chart.

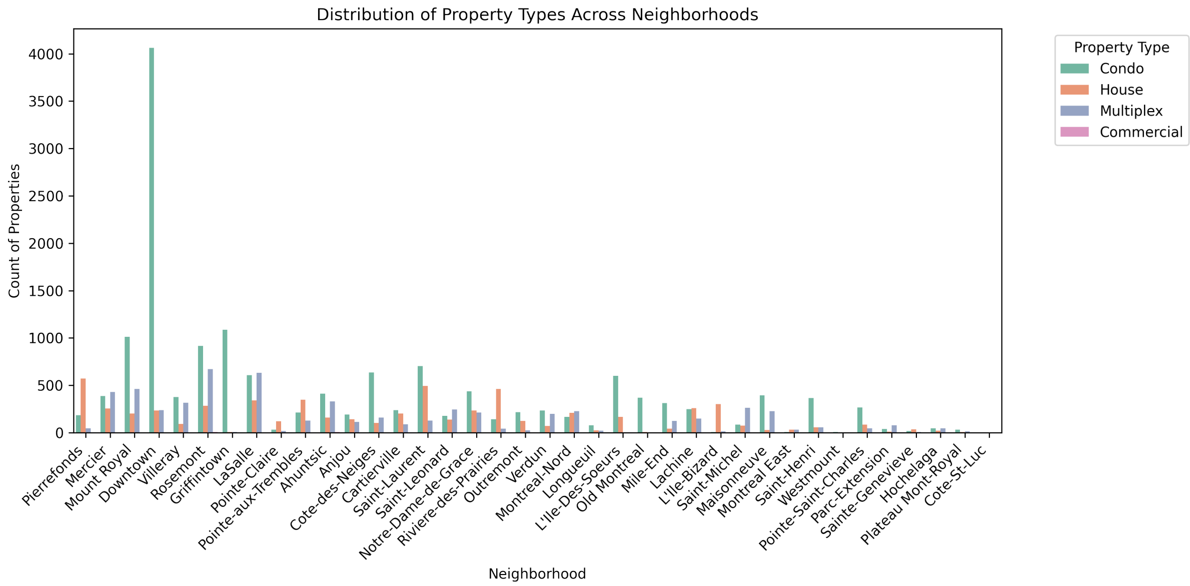


Figure 5: Property type distribution per neighborhood.

3.3.3 Correlation Analysis

Property Sales
Most features show low correlation, suggesting independence between each other. Exceptions include pool-related features and elevator presence with adapted mobility features.
Rental Market
Rental data shows stronger correlations between features.
Notable correlations:

- Gym and elevator (0.7-0.8): Indicative of multi-story condominium complexes, possibly indicating larger, amenity-rich developments with higher-priced units
- Air conditioning and dishwasher (0.6-0.7): Suggests bundling of amenities by property owners

These correlations provide insights into property characteristics and potential pricing factors

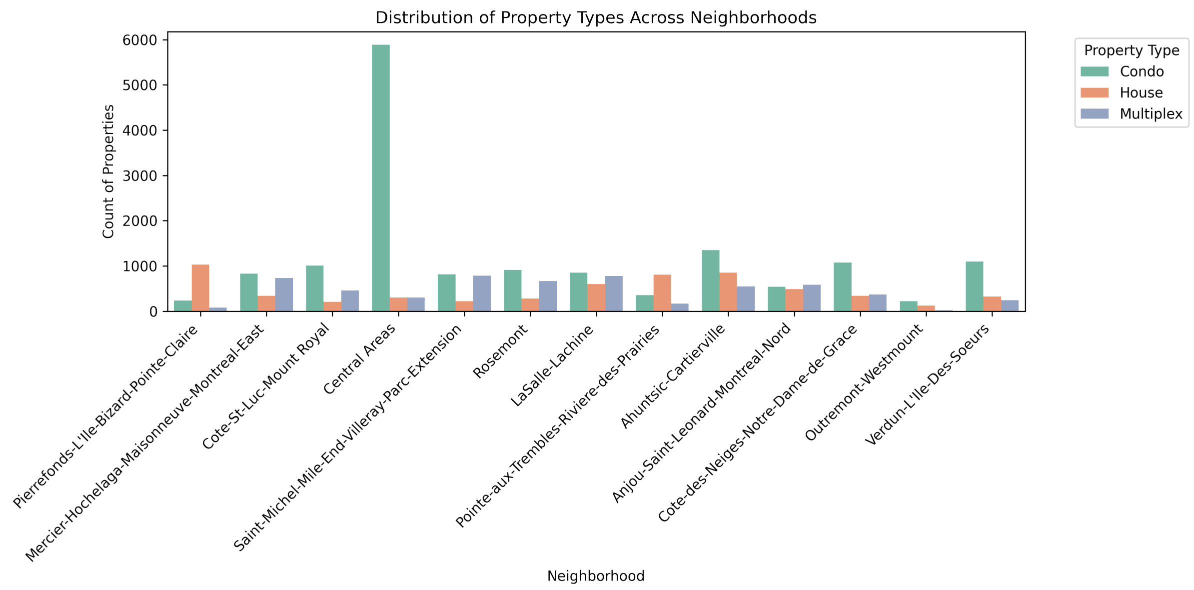
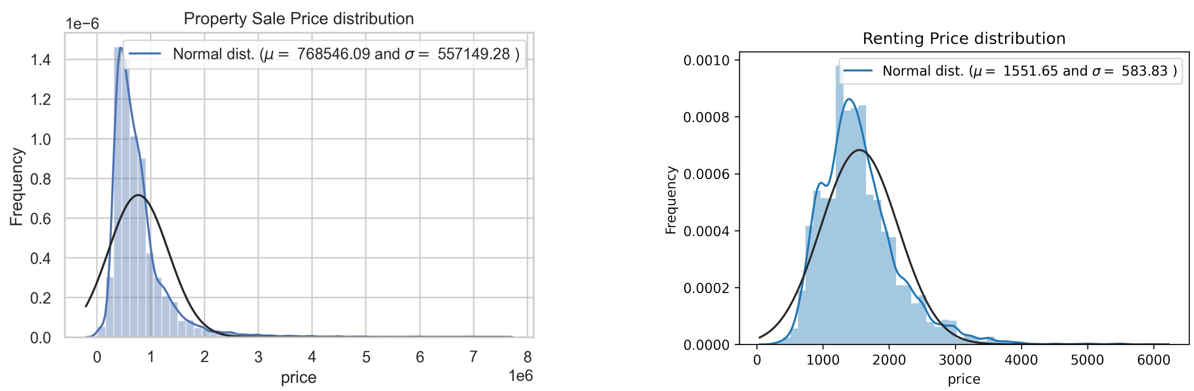


Figure 6: Property type distribution per areas.



(a) Property Sales

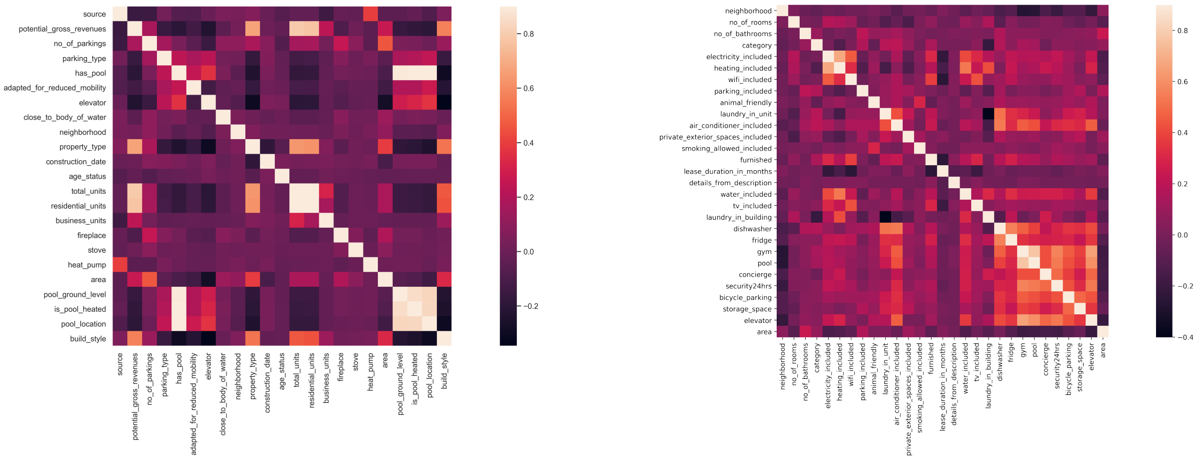
(b) Rental

Figure 7: Price distributions for the rentals and property sales data.

in the rental market. Additionally, the correlated features in rental data may enhance model robustness and performance through information redundancy.

3.3.4 Market Insights and Trends

- Montreal remains relatively affordable compared to other Canadian metropolises, despite recent price increases.
- Central areas show higher rental premiums, while some outlying areas offer better value for property purchases.
- Amenities significantly impact rental prices, with gym access and modern appliances commanding a premium.
- The market appears to favor sellers, but high interest rates may impact affordability and demand.
- Areas with good public transit connectivity show more active markets for both rentals and sales.



(a) Property Sales (b) Rental

Figure 8: Heatmaps of correlations between feature for Property Sales and Rental data.

3.3.5 Conclusion

Montreal's real estate market presents a complex landscape with opportunities and challenges for both investors and residents. While the city remains relatively affordable, recent trends suggest a gradual increase in prices across all property types. The disparity between central and peripheral areas, coupled with the impact of amenities on pricing, offers diverse investment opportunities. However, potential buyers and investors should carefully consider the effects of higher interest rates on long-term affordability and market dynamics.

3.4 Risk

In the context of real estate investment, calculating risk presents unique challenges due to the nature of property transactions and available data. Our approach to computing risk takes into account several key factors:

1. The specificity of the real estate market
2. Limited data availability (from 2020 onward)
3. Infrequent sales of individual properties within a short timeframe

To address these challenges, we've developed a novel method for estimating risk that considers similar properties within a neighborhood and across the entire city of Montreal.

3.4.1 Risk Calculation Methodology

Let A be the property of interest, β_A be a property within the same neighborhood as A, and M be a property within the entire city of Montreal. We define:

- P_A: The set of all properties P with features similar to A
- M_A: The set of all properties M with features similar to A, excluding neighborhood

Since the sample sizes of M_A and P_A differ, directly computing the correlation between them is not feasible. To resolve this, we use aligned samples from both distributions, denoted as M_A^a and P_A^a, where the a index refers to the aligned data. The method for aligning these samples is detailed in subsection 3.4.2. As data alignment does not affect the calculation of relative volatility, the standard deviations for both distributions are computed using the full datasets.

The risk β is calculated using the following formula:

$$\beta(A, M) = Corr(P_A^a, M_A^a) \cdot \frac{standardDeviation(P_A)}{standardDeviation(M_A)} \tag{6}$$

3.4.2 Aligning mismatching data counts

Naturally, as we are using similar properties to the asset P_A, the amount of data fitting those criterias is unlikely to align with the entire market dataset. To address the discrepancy in the number of price points between P_A and M_A, we developed a temporal grouping technique. Below is a simplified explanation; for more details, refer to Algorithm 3 in the appendix:

1. Group prices for both P_A and M_A by the week (Since Real Estate Prices appear much more stable than publicly traded assets, we consider a week the be a good enough time frame for a measure of "instant price")
2. Pair matching P_A and M_A groups by the week, discard the remaining groups
3. For each group, we calculate the median(instead of the mean to avoid sensitivity of the extremes)

After this, we are left with aligned samples of P_A and M_A we are ready to calculate our correlation.

3.4.3 Risk Normalization

As previously mentioned, the nature of RE data results in a lower risk investment setting compared to the stock market. To make better use of this metric, we propose emphasizing the risk by computing a β relative to the Montreal real estate market. Specifically, we calculate β for each unit in our investment landscape, which is divided into {neighborhood; property type} pairs.
We then normalize the β_A of interest using the maximum β_{max} and minimum β_{min} values found in our dataset. The normalization is done as follows:

$$\beta_{norm} = \frac{\beta_A - \beta_{min}}{\beta_{max} - \beta_{min}}$$

To provide insight into the riskiest markets in the City of Montreal based on our data, we have computed a risk table for each neighborhood, normalized across property types. The results are shown in Figure 9.
Finally, after normalization, we scale β to the range of -1 to 2 to ensure that it captures and conveys the complete spectrum of information to α.

3.5 Revenue and Price Prediction

As mentioned earlier, we trained regression models on our scraped data to predict property sales prices and rental rates, which we use to estimate the expected return on a property. Once our price prediction function, f_p, and rental prediction function, f_r, are trained, we can generate these forecasts.
For the property sales price, the process is straightforward: we simply predict the price based on the features of the property we are evaluating. However, estimating revenues and profits is more complex. The user first provides an estimate of the property's operating costs, and we then calculate yearly revenues by estimating monthly rental income and multiplying it by 12.

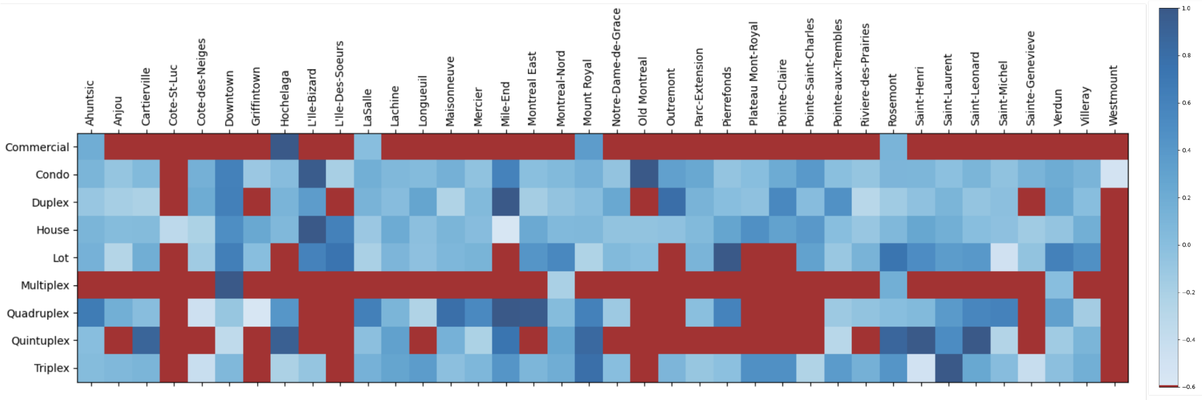


Figure 9: Relative market volatility by property type and neighborhood. The red values are vacant from our data.

Since our rental model operates on a per-unit basis, we require unit-specific information to make accurate predictions. This is relatively simple for single-family homes or condos but becomes more challenging for multi-unit properties, where each unit may vary in terms of the number of rooms.
To address this, we offer two options:

1. The user enters individual details for each unit (number of rooms, bathrooms, and area in square feet).

2. The user provides no details, and we estimate the units based on our data.

To make the **realistic unit** estimate, we proceed as follows:

- We start by using rental data from the same neighborhood as the property to ensure relevant comparisons.

- The most common room configurations are identified, and the average room size is calculated from nearby properties.

- We estimate rental income for each room configuration, adjusting for the unit's size and weighting more common configurations accordingly.

- The total revenue for the property is calculated by multiplying the estimated rental income per unit by the total number of units.

Finally, once we have our revenues, operating costs and asset price, we are ready to calculate alpha.

3.6 Alpha Calculation

To calculate α, we use the formula α_{RE} from (1) in the following algorithm:

4 Results and Discussion

4.1 Training setup

Non-Deep Estimators
Non-deep estimators were implemented using Scikit-Learn, with StandardScaler preprocessing. We evaluated Random Forest, SVR, XGBoost, CatBoost, Gradient Boosting, LightGBM, NGBoost,

Algorithm 1 Alpha Calculation for Real Estate Investment

Require: $purchase\_price$, $revenues$, $operating\_costs$, $property$, $market_{data}$, $risk\_free\_rate$

1: $beta \leftarrow$ ComputeBeta($property$, $market_{data}$, $prices$) 2
2: $NOI \leftarrow revenues - operating\_costs$
3: $actual\_returns \leftarrow NOI/purchase\_price$
4: $expected\_returns \leftarrow$ EstimateReturns($property$, $operating\_costs$)
5: $alpha \leftarrow actual\_returns - risk\_free\_rate - beta \times (expected\_returns - risk\_free\_rate)$
6: **return** α

Ridge, Lasso, and ElasticNet models. Hyperparameter tuning employed 5-fold cross-validation GridSearch (see Appendix 6). Experiments were conducted on an Intel i7-8700 CPU (3.20GHz, 24GB RAM). Feature selection attempts yielded no improvements. Despite exploring various ensembling techniques (stacking, voting, averaging), individual models—RandomForest for renting data and Gradient Boosting for property data—consistently outperformed ensemble methods.

Deep Learning Models

All DL models were implemented using PyTorch and PyTorch Lightning. We explored ResNet, FT Transformers, and Tab Transformer architectures. In addition, we explored pretraining our Transformer using Masked Autoencoder approach. The models were trained with an AdamW optimizer, cosine learning rate scheduling, and trained for 100 epochs with a batch size of 32 and initial learning rate of 10^{-4}. Experiments were conducted on an NVIDIA GeForce GTX1060 (3GB) GPU. After training each model, the three models were ensembled using a MLP as a meta-learner for stacking. The stacking models were trained for an additional 100 epochs. For each dataset, stacking was evaluated in two configurations: one using the output logits from the final layer of each model (referred to as $DeepStack_{Logits}$) and another using the predicted price (referred to as $DeepStack_{Preds}$). For detailed hyperparameters, refer to Appendix **??**.

4.2 Evaluation

The Evaluation was conducted on two datasets: PropertySalesPrice ($27,712$ samples, price range $\$9,500$–$\$7,500,000$) and Renting ($21,833$ samples, price range $\$280$–$\$5,990$). Both datasets were restricted to Montreal and underwent log1p transformation for training. We employed an 80:20 train-test split for non-deep estimators and a 70:10:20 train-validation-test split for deep models. Performance was assessed using Root Mean Squared Logarithmic Error (RMSLE), defined in (7), where p_i and a_i are predicted and actual values, respectively. This metric was chosen for its sensitivity to relative errors. The same training regimen was applied consistently across both PropertySalesPrice and Renting datasets.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2} \tag{7}$$

4.3 Experimental Results

4.3.1 Model Performance

Table 1 displays performance results, measured using RMSLE, for both *Property* and *Renting* datasets. The estimators are listed at the top of the table, while the deep learning models are shown at the bottom.

Table 1: RMSLE Performance of Various Models on Renting and Property Data

Model	Renting ↓	Property ↓
RandomForest	**0.1121**	0.1862
GBR	0.1391	**0.1851**
CatBoost	0.1381	0.2060
LightGBM	0.1412	0.2124
XGBoost	0.2822	0.2061
NGBoost	0.2668	0.4487
Ridge	0.2629	0.4507
SVR	0.1758	0.3420
ElasticNet	0.2723	0.4642
Lasso	0.3233	0.5155
FT-Transformer	0.1884	0.2594
Tab-Transformer	0.2367	0.3101
ResNet	0.2059	0.3048
DeepStack$_{\mathbf{Preds}}$	0.1195	0.2263
DeepStack$_{\mathbf{Logits}}$	0.1441	0.2345

Note: Bold values indicate the best-performing models (lowest RMSLE) for each dataset.

4.3.2 Discussion

The results across different models for the *Renting* and *Property* datasets highlight several notable patterns in the performance of both deep learning and non-deep learning models for tabular data regression tasks.

Renting Dataset The *RandomForest* model outperformed all other models on the renting dataset, achieving an RMSLE of 0.1121, significantly better than the second-best estimator model, *CatBoost* (0.1381), and *Gradient Boosting Regressor* (0.1391). This indicates RandomForest's strong generalization ability in this context, particularly for tabular data with simpler categorical features, such as the predominantly boolean categories in the renting dataset. $DeepStack_{Preds}$ achieved second place across all models, with an RMSLE of 0.1195.
Among boosting methods, a noticeable gap in performance was observed. While *CatBoost*, *LightGBM*, and *GBR* performed well, models like *XGBoost* (0.2822) and *NGBoost* significantly underperformed. This could be attributed to the different tree growth strategies employed by these models. The former set of models use a leaf-wise (best-first) tree growth strategy, while *XGBoost* and *NGBoost* rely on a level-wise (depth-first) strategy. Additionally, models like *CatBoost* and *LightGBM* have better native support for handling categorical data, which may explain their stronger performance.
Another observation is that the overall performance on the renting dataset was better compared to the property dataset. This may be due to the simpler nature of the categorical features in the renting dataset, where most categorical variables are Boolean, unlike the property dataset, which contains fields with multiple categories. For instance, the property type feature in the property dataset has 9 categories, making optimization more complex.

Property Dataset On the property dataset, GBR was the best-performing model, with an RMSLE of 0.1851, followed closely by *RandomForest* (0.1862). Most boosting methods (*CatBoost*, *LightGBM*, *GBR*) performed similarly, achieving RMSLEs within a narrow range (0.2 ± 0.015). However, *NGBoost* performed poorly, with an RMSLE of 0.4487, indicating its ineffectiveness in this context.

Across Datasets A consistent observation across both datasets is the underperformance of Lasso regression, which recorded the worst results, with an RMSLE of 0.3233 on the renting dataset and 0.5155 on the property dataset. This highlights Lasso's limitations for regression tasks involving complex tabular data, where interactions between features are essential for accurate predictions.

Deep learning models demonstrated competitive performance compared to traditional estimators but did not outperform the best non-deep models. Despite extensive hyperparameter tuning and significantly higher computational resource requirements, deep learning models could not close the gap with the best-performing classical models such as *RandomForest* and *GBR*. This suggests that, for these types of tabular datasets, deep learning may not yet be ready to replace classical models, especially when considering the disproportionate effort required for training.
However, stacking methods did show promise when applied to the deep learning models. Unlike the ensembling attempts with classical models, which did not lead to improvements, stacking consistently enhanced the performance of deep learning models. For both the renting and property datasets, stacking using the model predictions (as opposed to logits) yielded the best results.
Among the deep learning models tested, *FT-Transformer* can be considered state-of-the-art (SOTA) for tabular data on these datasets, achieving an RMSLE of 0.1884 on the renting dataset and 0.2594 on the property dataset. While it still lags behind the best classical models, *FT-Transformer* stands out as the most promising deep learning model in this domain.

Final Remarks Overall, the findings underscore the effectiveness of traditional models like *RandomForest* and *GBR* for tabular data regression, particularly for datasets with complex categorical features, such as the property dataset. While deep learning models show potential, their practical value remains limited by the significant effort required for training and tuning. Further advancements in deep learning architectures and optimization techniques may be necessary before these models can consistently outperform classical methods on tabular data.

4.4 Real Estate Alpha Calculator Demonstration

4.4.1 Context

Imagine you are an investor looking for a quadruplex in the city. You are considering two properties located in Downtown and Saint-Michel (see Table 2).

Table 2: Potential Quadruplexes Comparison Table

Price	**Neighborhood**	**Property Type**	**Revenue**	**Operating Costs**
\$860,000	Saint-Michel	Quadruplex	\$38,000	\$3,926
\$875,000	Downtown	Quadruplex	\$67,200	\$5,951

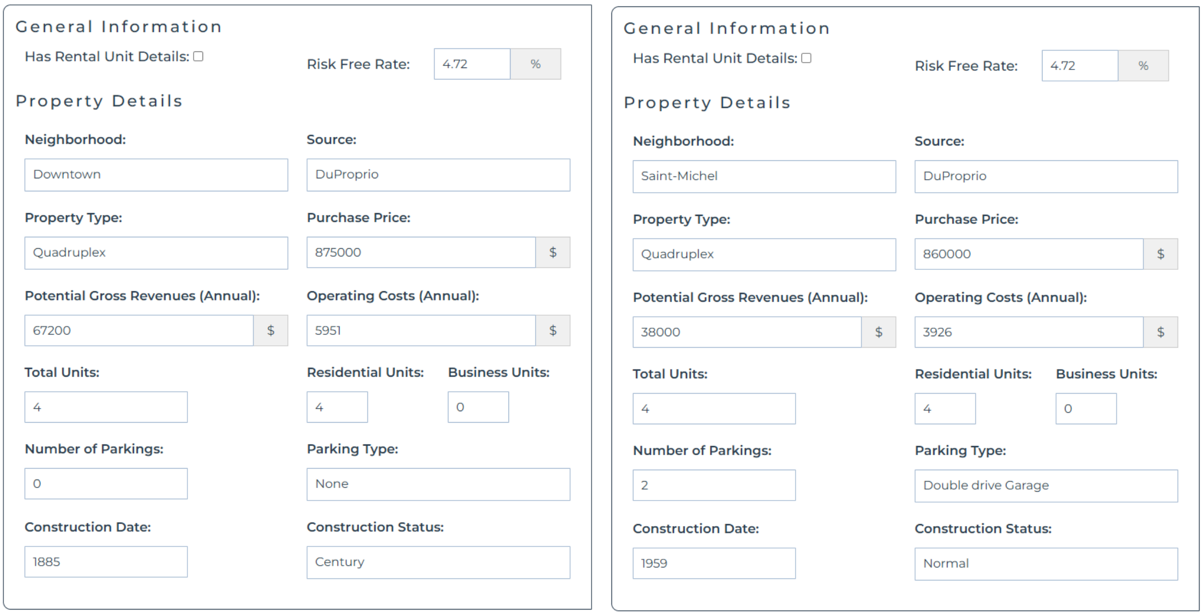
Note: Examples taken from real data available on DuProprio.

After entering the details of each deal into our calculator, we obtained the following outputs (see Table 3).

Table 3: Alpha Calculation Output

Neighborhood	**Downtown**	**Saint-Michel**
Expected Returns	3.769%	4.438%
Actual Returns	7%	3.962%
Risk-Free Rate	4.72%	
Beta	0.3684	1.464
Alpha	2.63%	-0.3452%

Now let's break down the α_{RE} formula (see Equation 4) to understand why these two properties have the alpha values they do.



(a) Quadruplex in Downtown (b) Quadruplex in Saint-Michel

Figure 10: Alpha calculator input examples for both quadruplexes.

$$
\begin{aligned}
\alpha_{RE} &= \text{Actual Returns} - [R_f + \beta_C \times (\text{Expected Returns} - R_f)] \\
\alpha_{\text{Saint-Michel}} &= 3.962 - 4.72 - 1.464 \times (4.438 - 4.72) \\
&= -0.3452 \\
\alpha_{\text{Downtown}} &= 7 - 4.72 - 0.3684 \times (3.769 - 4.72) \\
&= 2.63
\end{aligned}
$$

Even though both properties are of the same type, the one in Downtown has a much higher α than its counterpart, despite a \$15,000 higher price. There are two main reasons for this:

- The Downtown property yields significantly higher revenues for what you pay upfront and for its maintenance, with a 7% return compared to 3.962%.
- The market for quadruplexes in Saint-Michel is riskier than that in Downtown (with a β of 1.464 vs. 0.3684) (see Figure 9).

To improve the alpha for the Saint-Michel property, the investor should look into reducing operating costs and increasing rental income.

5 Conclusion

This paper introduced the Real Estate Alpha Calculator, a novel tool designed to streamline property investment screening in Montreal's real estate market. By adapting the Capital Asset Pricing Model (CAPM) to real estate investments and leveraging machine learning for price predictions, we developed a systematic approach to quantifying potential returns while accounting for systemic risks.

Our methodology encompassed the development of a real estate beta calculation, comprehensive neighborhood profiling, and the implementation of various machine learning models for property sales and renting price prediction. The empirical results demonstrated that traditional machine learning approaches outperformed more complex deep learning models, with Random Forest achieving optimal performance for rental properties (RMSLE 0.1121) and Gradient Boosting Regressor excelling for property values (RMSLE 0.1851). The analysis revealed that the relative simplicity of rental data categories contributed to more accurate predictions compared to the complex property dataset.

While the Real Estate Alpha Calculator represents a significant advancement in quantitative real estate investment analysis, several limitations should be acknowledged. The tool does not account for potential property appreciation over time, and predictions for certain areas are constrained by limited data availability. Additionally, some property-specific risks may not be fully captured by the systemic risk assessment. Despite these constraints, the calculator provides investors with a valuable, data-driven approach to initial property screening, effectively bridging the gap between financial modeling and practical real estate investment decisions.

References

[1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[2] Canadim. Top 5 cheapest cities to live in canada: Affordable living explored, 2024. Accessed: 2024-09-19.

[3] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

[4] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems*, pages 155–161. MIT Press, 1997.

[5] Tony Duan, Anand Avati, Daisy Yi Ding, Khanh K Thai, Sanjay Basu, Andrew Y Ng, and Alejandro Schuler. Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*, pages 2690–2700. PMLR, 2020.

[6] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 1999.

[7] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data, 2023.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[9] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[10] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings, 2020.

[11] Michael C. Jensen. The performance of mutual funds in the period 1945-1964. *Journal of Finance*, 23(2):389–416, 1968.

[12] Michael C. Jensen, Fischer Black, and Myron S. Scholes. The capital asset pricing model: Some empirical tests. *Studies in the Theory of Capital Markets*, 1972.

[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

[14] nesto.ca. Montreal housing market outlook 2024, 2024. Accessed: 2024-09-19.

[15] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pages 6638–6648, 2018.

[16] Ratehub.ca. Montreal mortgage rates - best mortgage rates montreal, 2024. Accessed: 2024-09-19.

[17] Société de transport de Montréal. Maps, 2024. Accessed: September 23, 2024.

[18] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[20] Guri Zabërgja, Arlind Kadra, and Josif Grabocka. Tabular data: Is attention all you need? *arXiv preprint arXiv:2402.03970*, Feb 2024.

[21] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Appendices

A Appendix: Implementation Details

Here are implementation details for the machine learning models.

Table 4: Model Architectures and Parameters

Model	**Dim**	**Depth**	**Heads**	**Dim_head**	**Params (M)**
FT-Transformer	192	6	4	192	6.4
Tab-Transformer	192	6	4	192	10.4
ResNet	256	8	NA	NA	2.6
DeepStack	256	3	NA	NA	X.X

Table 5: Torch Training Hyperparameters for Grid Search

Hyperparameter	**Values**
Optimizer	AdamW
Epochs	100
Learning Rate (lr)	{1e-5, 1e-4, 1e-3}
Weight Decay	{1e-5, 1e-4}
Scheduler	CosineAnnealingWarmRestarts(optimizer, T_0=10, T_mult=2)
Batch Size	32
Optimizer Momentum ($\beta 1, \beta 2$)	(0.9, 0.999)

Algorithm 2 Beta Calculation

```
property_std    ▷ Prices paired with their observed time for a given property type & neighborhood
market_std      ▷ Prices paired with their observed dates for a given property type
property_std = standard_deviation(property_prices)
market_std = standard_deviation(market_prices)
correlation = calculate_weekly_correlation(property_prices, market_prices)
beta = correlation × (property_std / market_std)
return beta
```

Algorithm 3 Weekly Correlation Calculation

```
property['creation_date'] = convert_to_datetime(property['creation_date'])
market['creation_date'] = convert_to_datetime(market['creation_date'])
property['year_week'] = extract_iso_week(property['creation_date'])
property['year'] = extract_year(property['creation_date'])
property['year_week'] = format_year_week(property['year'], property['year_week'])
market['year_week'] = extract_iso_week(market['creation_date'])
market['year'] = extract_year(market['creation_date'])
market['year_week'] = format_year_week(market['year'], market['year_week'])
property_grouped = group_by(property, 'year_week').calculate_median('price')
market_grouped = group_by(market, 'year_week').calculate_median('price')
matching_weeks = intersect(property_grouped.index, market_grouped.index)
property_medians = [property_grouped[week] for week in matching_weeks]
market_medians = [market_grouped[week] for week in matching_weeks]
if len(property_medians) > 1 and len(market_medians) > 1 then
    correlation = calculate_correlation(property_medians, market_medians)
else
    correlation = None    ▷ Not enough data for correlation
end if
return correlation
```

Table 6: Grid Search Hyperparameters for Various Models

Model	Hyperparameter	Values
RandomForestRegressor	*n_estimators*	50, 300
	max_depth	3, 20
	min_samples_split	2, 10
	min_samples_leaf	1, 4
	max_features	0.1, 1.0
SVR	*C*	0.1, 10
	gamma	0.01, 1
	epsilon	0.1, 0.5
XGBRegressor	*n_estimators*	50, 100, 200, 500
	max_depth	3, 5, 7, 10
	learning_rate	0.01, 0.1, 0.2, 0.3
	subsample	0.5, 0.8, 0.9, 1.0
	colsample_bytree	0.3, 0.7, 1.0
CatBoostRegressor	*iterations*	50, 300
	depth	4, 10
	learning_rate	0.01, 0.3
	l2_leaf_reg	1, 5
	bagging_temperature	0.0, 1.0
GradientBoostingRegressor	*n_estimators*	50, 300
	max_depth	3, 10
	learning_rate	0.01, 0.3
	subsample	0.5, 1.0
	max_features	0.1, 1.0
LGBMRegressor	*n_estimators*	50, 300
	max_depth	-1, 20
	learning_rate	0.01, 0.3
	num_leaves	20, 100
	subsample	0.5, 1.0
NGBRegressor	*n_estimators*	50, 300
	learning_rate	0.01, 0.3
	minibatch_frac	0.5, 1.0
	col_sample	0.5, 1.0
	max_depth	3, 10
Ridge	*alpha*	0.1, 1.0, 10.0, 100.0
	solver	auto, svd, cholesky, lsqr, sparse_cg, sag, saga
Lasso	*alpha*	0.1, 100.0
	max_iter	1000, 5000
	tol	0.0001, 0.01
ElasticNet	*alpha*	0.1, 1.0, 10.0, 100.0
	l1_ratio	0.1, 0.5, 0.9
	max_iter	1000, 5000
	tol	0.0001, 0.001, 0.01