Leveraging Multi-modal Data for House Price Prediction: A Case Study in Toronto

Jie Peng Chen
University of Toronto
jp.chen@mail.utoronto.ca

Zixuan Zhou
University of Toronto
zixuantroy.zhou@mail.utoronto.ca

Hanrui Fan
University of Toronto
hanrui.fan@mail.utoronto.ca

Abstract

Accurate house price prediction is crucial for stakeholders such as buyers and investors, especially in dynamic real estate markets. This paper explores a novel approach to house price estimation using multi-modal data from the Toronto Multiple Listing Service (MLS). This dataset includes structured property listing information such as the number of bedrooms and lot size, property photos, and listing descriptions. Our model aims to enhance traditional methods by incorporating unstructured data from these listings, enriched with additional datasets including the proximity to subway and commuter train stations, surrounding schools' scores, census data such as average household income, and recent area sales trends. By employing advanced feature extraction techniques from both textual and visual data, our approach provides a comprehensive analysis of the factors influencing property values. Through a case study in Toronto, we explore the effectiveness of our multi-modal model in providing more precise predictions compared to conventional valuation methods. Although the expected improvements were not realized, this paper underscores the potential of integrating diverse data sources to refine the accuracy of house price predictions. Our source code is available at: https://github.com/frank192168/csc413-project

1 Introduction

Accurate house price prediction is crucial for buyers and investors in dynamic markets like Toronto, where buyers aim to find properties priced fairly or below market value. The challenge lies in accurately determining whether a listed property is appropriately priced relative to the market. In Toronto, properties are listed through the Toronto Multiple Listing Service (MLS), where real estate agents, drawing on their knowledge of the local market, estimate and list prices. However, these estimates can vary significantly, influenced by agents' strategies or sellers' pricing expectations. For instance, agents might intentionally list properties at low prices to spark bidding wars that drive up the final sale price. In this paper, we propose a multimodal model for house price prediction with our collected environmental data, such as local information of schools, transportation, census information and market trends.

Problem Definition. This study explores the integration of various data types, including structured and unstructured data, to predict residential property prices in Toronto using Toronto's MLS data. Our approach incorporates numeric and categorical data, property photos, and textual descriptions from listings. We define our feature set as $X = \{X_{num}, X_{cat}, X_{img}, X_{txt}\}$, where X_{num} and X_{cat} are numeric and categorical features, X_{img} denotes image features extracted from property photos,

Preprint. Under review.

and X_{txt} includes textual features derived from property listing descriptions. Our goal is to develop a predictive model that utilizes these comprehensive inputs to estimate the current market valuation y_t of a house h at the time t it is listed on MLS.

Motivation. Numerous real estate websites utilize machine learning to provide estimated property prices, including "Zestimate" by Zillow[10] and "SigmaEstimate" by HouseSigma[13]. These models, while innovative, typically underutilize unstructured data such as images and textual descriptions, which are crucial for capturing the comprehensive value of a property. For example, the aesthetic appeal of property photos can significantly influence buyer decisions, while detailed descriptions provide insights into aspects like renovations or structural damages, which are vital for accurate valuations.

The use of advanced neural networks, such as Convolutional Neural Networks (CNNs) for analyzing visual data and Recurrent Neural Networks (RNNs) and Transformers for processing textual information, presents a robust methodology for integrating diverse data sources. Although prior research has leveraged structured data combined with either images [2][17] or text [19], the complete integration of structured data with both images and text into a single predictive model has rarely been explored. This study aims to explore the potential of such an integrated approach, examining how effectively these technologies can be combined to enhance the accuracy and reliability of property valuations, even though our initial outcomes did not show the anticipated improvements.

Contributions. Our exploratory contributions are twofold: First, we developed a comprehensive dataset that extends beyond standard MLS data, incorporating additional contextual details such as school rankings, transit options, market trends and census data. Second, we attempted to integrate diverse data types through a multimodal neural network model, aiming to enhance predictive accuracy, though the expected improvements were not realized.

2 Related Work

The prediction of house prices has historically leveraged a wide range of data types and methodologies[9][21], reflecting the multifaceted nature of real estate valuation. This section reviews relevant studies that utilize machine learning techniques for extracting and analyzing features from both structured and unstructured data sources, providing a backdrop against which we position our research.

Structured Data in House Price Prediction. A study of methods and input data types for house price prediction[9] show that previous researches have primarily utilized structured data—focusing on property characteristics such as lot size, number of rooms, and location—to predict house prices using traditional regression models and basic machine learning techniques. These approaches, while foundational, often fail to capture the full complexity of market dynamics. Recent literature indicates a shift towards more sophisticated modeling techniques and richer data inputs to improve prediction accuracy [23][6][8].

Incorporating Unstructured Data. The integration of unstructured data, particularly images, has been less explored in traditional real estate predictive models. Convolutional Neural Networks (CNNs) have shown significant promise in this domain, effectively analyzing visual data from property photos to extract features that influence buyer perceptions and decision-making.[2][18][27] [17][22][25] Similarly, Natural Language Processing (NLP) techniques have been applied to textual descriptions to glean additional insights that are not evident in structured data alone.[19]

Advanced Machine Learning Approaches. Further advancements in deep learning have introduced more sophisticated models such as Long Short-Term Memory networks (LSTMs) and Transformers, which have been used to capture temporal and contextual nuances in data.[32][5][31] These models have proven effective in other domains and are beginning to be explored in real estate for their potential to handle dynamic market conditions and historical data trends.

3 Data Collection and Analysis

Dataset Acquisition. Our study employs the Toronto real estate MLS data provided by the Toronto Regional Real Estate Board (TRREB)[30]. This data spans over a decade and is rich in numerical, categorical, and textual data, with each real estate listing accompanied by multiple photographs. It

dataset	date range	# listings	# images	# features	# enriched features
pre-training	2021-2022	152,722	4,491,246	317	41
training	2023	50,026	1,514,427	317	41
testing	2024-01	2,908	87,400	317	41

Table 1: MLS data

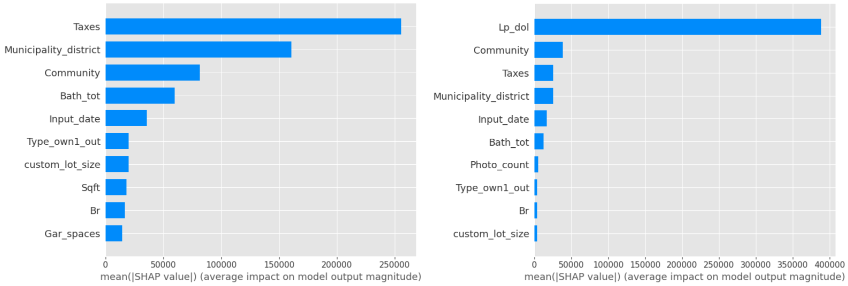


Figure 1: Top 10 feature contributions with/without list price

covers properties not just in the City of Toronto but throughout the Greater Toronto Area (GTA), including both residential and condominium properties. For this study, we focus exclusively on the residential dataset within the GTA, excluding condos. The size of the data we used shown in Table 1. Each listing contains 317 features and 41 additional features that we have collected.

Enrichment of Data Sources. To enhance the utility of the MLS dataset, we incorporated additional data from several external sources. Geolocation data was obtained using the Google Map API[12] and Mapbox API[14], enabling precise spatial analysis. We collected school-related data, including rankings[15] and demographic information[7], and integrated census data from Statistics Canada[3] to enrich our understanding of the socio-economic context of each property. We also calculated the market trends for the neighbourhood of each listing in the TRREB dataset. Information on the proximity of public transit options like subway and train stations was also gathered from Google Map and calculated for each listing.

Data Integration and Preprocessing. The integration process consolidates these diverse data streams into a unified dataset, where each row represents a unique real estate listing. Properties are assigned to specific census subdivisions based on their geolocation, with proximity to relevant schools and transit stations calculated within a 1-2km radius. Preprocessing includes normalization of continuous variables and imputation of missing values. Property photos are uploaded by agents, the resolution, angle, size, number of photos are also vary. Photos are resized to 224×224 pixels for CNN processing, and categorical data are transformed using one-hot encoding. We run a boosting model using all structured features, and plot feature importance graph using SHAP value[20]. This helps us narrow down the feature space. It also shows that when include the list price as feature to predict the actual sold price, it has a huge impact on the prediction, as shown in Figure 1. This is because the list price considered as human experts that estimate our target value.

Data Quality Considerations. The MLS dataset contains 317 fields, with all entries input manually by agents, leading to frequent empty fields and occasional errors such as typos in list prices. We have implemented filters to exclude obvious errors, such as sold prices less than or equal to 10 dollars, and listings outside the GTA. We use listings sold between 2021 and 2023, a period influenced by the COVID-19 pandemic, which saw a significant fluctuation in sold volumes (94,216 in 2021, 58,506 in 2022, and 50,026 in 2023). We also handle potential inaccuracies in geolocation data by only using data where the API's confidence level is over 0.75, ensuring higher reliability in the contextual data linked to each property.

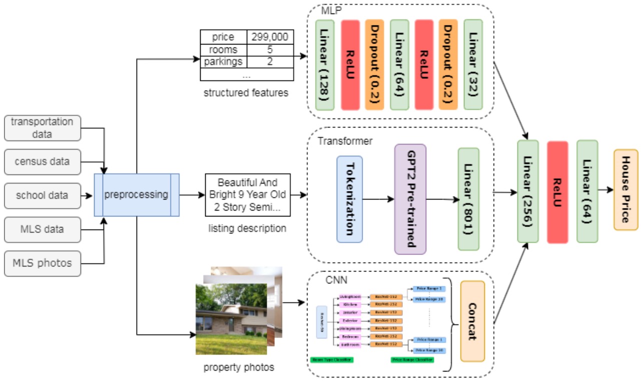
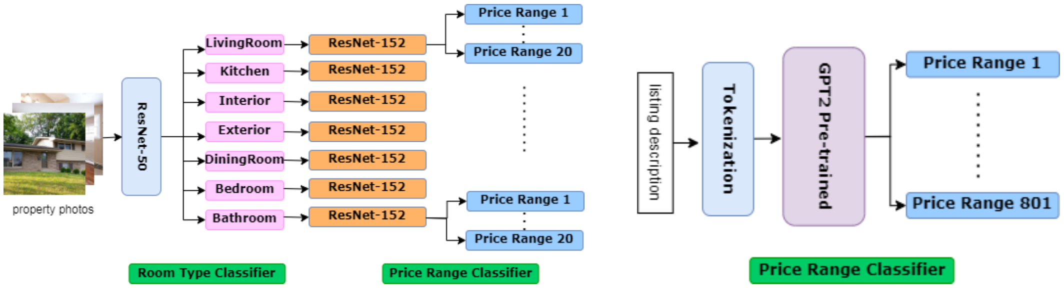


Figure 2: Proposed multimodal architecture



(a) CNN model for image

(b) Transformer model for text

Figure 3: Pre-training models

4 Methodologies

An overview of our house price prediction model is shown in Figure 2, which illustrates the overall architecture of the proposed model. Our multimodal model is comprised of three primary components, each tailored to process a specific type of data:

Multilayer Perceptron (MLP). This component serves as a feature extractor for structured data, applying three layers of perceptrons to distill and interpret the input data, enhancing the model's ability to understand and utilize numerical and categorical data effectively. The MLP is fundamental in processing the structured data before integration with other data modalities in our multimodal approach.

Image Feature Extractor with Pre-trained Classifier. Our image processing module employs multiple convolutional neural networks (CNNs) to analyze image data. Initially, we employ transfer learning on this model by replacing the final fully connected layer with one tailored to the number of output species, utilizing ResNet50[11] as the base model. We train this model on a publicly available labeled dataset[24] of house photos categorized by their labels. Notably, our dataset lacks these labels. To address this, we employ a methodology akin to semi-supervised learning, categorizing the unlabeled data directly. Subsequently, we construct individual CNNs for each category, resulting in a parallel set of CNNs.

Previous research introduced a luxury level parameter[22], which remained undefined and subjective. In our study, we conducted an analysis on all house prices within our private dataset and established multiple non-uniform price intervals. This approach provides a more nuanced understanding of the dataset. For the task of price range classification, we adopt ResNet152[11] as our base model, replacing the final fully connected layer to capture a broader array of features present in images depicting houses across different price ranges. This modification enhances the predictive accuracy of our model compared to standard CNN architectures. By tailoring our models to specific house image categories, we effectively reduce the number of features each CNN needs to capture ([22] [25]). The entire architecture of this part is shown in Figure 3a.

Pre-trained Transformer. Our methodology employs GPT-2[26], a sophisticated pre-trained model, to intricately process and extract features from textual data within property listings. Leveraging the transformer architecture, this approach capitalizes on the model's ability to comprehend and generate human-like text, thus providing profound insights into the descriptive elements of the listings. However, since we cannot assure the generated tokens are numerical, we augmented the model with a classifier for predicting house price intervals by appending a fully-connected layer at the model's end. Given that the descriptions contain unique features distinct from images, we use a more detailed price range classification than that of the CNN model. The architecture of this part is shown in Figure 3b.

Concatenated Features for the Last MLP Layers. Finally, after pretraining the three models using historical data, we freeze their weights and utilize the last layer as feature extractors. Given property listing information as input, we feed the selected numerical and categorical features to the MLP model, use the last feature layer other than the output layer, to output a feature vector. Each property image is fed into the CNN model, which outputs a room type and a price range for each image. If multiple images correspond to a single room type, each predicting different price ranges, we determine the majority price range as the final predicted price range for that room type. In the absence of a majority, we calculate the average and assign the corresponding price range. For each room type, we select the image's features with the highest probability of the predicted price range to represent that room type. If no images are available for a room type, we use the highest probability image's features available; if none are available, we set the feature vectors to zero. Finally, we concatenate the feature vectors for all seven room types in sequence and use this as the output features from the CNN model. For the description information, it feeds to the pretrain GPT-2 model, and also use the last feature layer to output a feature vector. Finally, we concatenate all three feature vectors from three models, and feed into the last MLP layers of our model, and predict the price.

5 Experiments

Data Split. Our data is divided into three distinct sets, each based on the sold date to preserve the time series nature of real estate pricing data, where price distributions may shift over time. We use historical data from 2021 to 2022 for pre-training each expert model. The primary training is then conducted on data from 2023, with testing performed on the most recent listings from January 2024, as shown in Table 1.

5.1 MLP Model

Data Preprocessing. For the MLP model, structured data undergoes feature selection, resulting in 73 key features, including environmental features that we collected. Categorical data are processed using one-hot encoding, increasing feature dimensions to 859. Numerical data are scaled using standard scaler with three settings to assess impact: no scaling, scaling all features, and scaling excluding price-sensitive features like sold price and taxes due to their wide range and critical importance.

Feature Set. As mentioned in Section 3, we evaluate the impact of including versus excluding the list price from the feature sets, as this is typically a key indicator considered by human experts in the real estate industry. Additionally, we assess the influence of enriched data on model performance. Accordingly, we have established four distinct feature sets:

- **MLS:** The MLS structured features, which contains list price.
- **MLS_ENV:** The MLS structured features, with our collected environmental features.
- **MLS_XLP:** The MLS structured features without list price.
- **MLS_XEN:** The MLS structured features without list price and without our collected environmental features.

Model Configuration. The final input dimension post-preprocessing is 859. The model architecture includes three hidden layers, with a grid search determining the optimal configuration from [(128, 64, 32), (256, 128, 64)] for neuron distribution per layer, learning rates of [0.01, 0.001], batch sizes [64, 128], and dropout rates [(0.0, 0.0), (0.2, 0.2)]. The activation function used is ReLU, suitable for regression tasks, and optimization is performed using the Adam optimizer, preferred over SGD due to its stability in handling wide-ranging target values. The loss function is Mean Square Error (MSE). The model predicts the actual sold price directly without any modification.

Training Process. We applied a grid search combined with 5-fold cross-validation to find the best hyperparameters for our model.

Baseline Model. Previous studies[23] leveraging the Toronto MLS data are difficult to directly compare due to the non-public availability of the dataset and the absence of sold price information at the time those studies were conducted. Recent updates to the MLS dataset now include sold prices, providing a valuable feature for more accurate price prediction. Therefore we chose XGBoost as our baseline model due to its established efficacy in house price prediction tasks[28][16]. We use a Python package that implements the XGBoost model for regression problems. By using grid search, the model is configured with 1000 estimators, a learning rate of 0.01, and the Mean Absolute Error (MAE) as the evaluation metric.

Evaluation Metrics. To evaluate the prediction performance of different methods and feature sets, we use the following three metrics. The first two metrics are widely used in prediction evaluation measures, Root Mean Squared Error [33] (RMSE) which calculate the square root of the average of all prediction error, and Mean Absolute Error [4] (MAE) which calculate the average of absolute error for each prediction. The following equations are for RMSE and MAE respectively:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \qquad MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{1}$$

Where $\hat{y}_i$ is the predicted price and y_i is the actual sold price. These metrics measure the average magnitude of the errors in a set of predictions, without considering their direction. Lower values of RMSE and MAE indicate better predictive accuracy.

Additionally, we define the Error Rate (ER), a metric commonly used in real estate platforms[10] to provide intuitive insights into price prediction accuracy: $ER = \left|\frac{\hat{y}-y}{y}\right|$

This metric facilitates easier interpretation for buyers and investors, reflecting the percentage error in predictions. We further analyze the distribution of ER by calculating the median ER of the predictions and the proportion of predictions where ER falls below 5%, 10%, and 20%, respectively. This breakdown helps to quantify the reliability of the model predictions at different levels of accuracy.

Models	Feature Set	MAE	SMRE	ER Value			
				median	<5%	<10%	<20%
XGBoost	MLS	97441.21	210606.37	5.61%	45.59%	75.29%	95.69%
	MLS_ENV	97088.87	195757.1	5.61%	45.26%	73.75%	95.32%
	MLS_XLP	218162.23	400458.02	12.41%	22.66%	41.64%	69.64%
	MLS_XEN	212357.18	375670.94	11.56%	24.5%	44.69%	72.68%
MLP	MLS	92759.73	137561.71	6.25%	39.32%	77.07%	97.14%
	MLS_ENV	100235.87	157491.49	5.07%	49.52%	72.18%	93.52%
	MLS_XLP	225760.19	1207097.2	13.85%	17.88%	36.42%	68.88%
	MLS_XEN	186151.4	375257.91	11.64%	21.09%	42.83%	77.05%

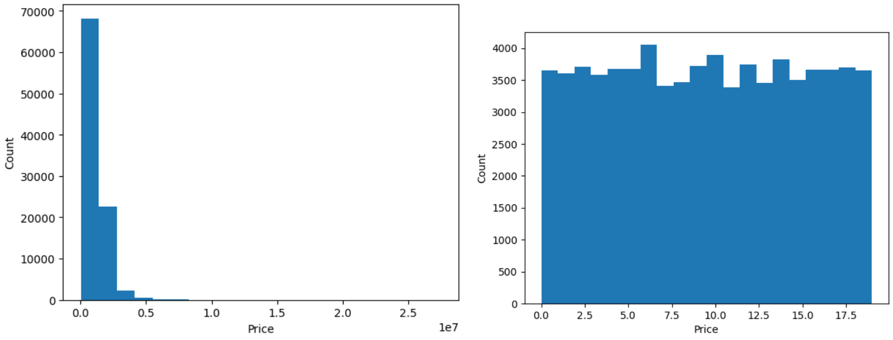
Table 2: Evaluations of Baseline Model and MLP Model

MLP Only Performance Comparison. Our experimental evaluation commenced with an assessment of numerical feature scaling's impact. Employing a standard scaler, we standardized features by eliminating the mean and scaling them to unit variance. Initially, we applied the scaler to all numerical features, including monetary values such as list price and property tax. This resulted in a significant performance enhancement on the validation set. However, the approach proved ineffective on unseen data, yielding an Error Rate (ER) less than 1%, and the predicted prices were anomalously low, in the range of hundreds to thousands of dollars.

Further investigation revealed a vast range in property taxes and list prices, with house prices varying from $500k to over $50 million. Consequently, applying standard scaling to such disparate values was deemed inappropriate. Excluding list price and property tax from scaling yielded reasonable accuracy on unseen data, thus we maintained these settings for subsequent experiments. Grid search revealed that the presence or absence of dropout layers did not markedly affect model performance. The optimal hyperparameters identified were 128, 64, and 32 neurons for the three hidden layers, a learning rate of 0.01, a batch size of 64, and dropout rates of 0.2.

To discern the influence of the list price feature and our collected environmental features, we trained models with consistent settings across various feature sets. Table 2 presents the outcomes on the test set comprising listings sold in January 2024. Notably, excluding the list price feature (MLS_XLP and MLS_XEN) resulted in a performance decline of approximately 22% and 27% for ER < 5%, respectively, with or without environmental features. The inclusion of environmental features elevated the ER < 5% from 39% to 49%, marking a substantial improvement.

Our primary focus metric was on ER < 5% as it provides a more intuitive and realistic gauge for buyers. Error rates of < 10% and < 20% are deemed less critical; for instance, a 20% error rate on a $2 million property implies a potential discrepancy of $200k, which is untenable for most purchasers. In the Toronto market, multimillion-dollar properties are not unusual. The model performance without environmental features did not surpass that of the baseline XGBoost model. Yet, with the inclusion of environmental data, our model demonstrated superior performance compared to the baseline. The effect of temporal dynamics on model accuracy was pronounced. Utilizing a more extensive training dataset did not correlate with performance gains; contrarily, it could potentially diminish accuracy. We experimented with training on data from a single month and predicting for the subsequent three months, observing a consistent decline in accuracy (ER < 5%).



(a) Unbalanced Distribution with equivalent interval

(b) Distribution after Relabeling

Figure 4: Data Distribution

5.2 Image Feature Extractor with Pre-trained Classifier:

Data Processing. We collected data on the number of images corresponding to each house listing, alongside their final selling price and the corresponding folder containing the images from our dataset. Given the variation in the number of images per listing, we opted to select a maximum of three images from each listing for training purposes. Additionally, as part of our preprocessing pipeline, we resized all images to a standardized dimension of 224x224 pixels to align with the input size requirement of ResNet[11]. We begin by partitioning the dataset into price intervals of equal length. However, upon analyzing the data distribution, depicted in the accompanying images, we observed an imbalance in the dataset, with certain categories containing significantly fewer images compared to others. To address this, we performed computational relabeling of the dataset, resulting in multiple non-uniform intervals.

Model Configuration. Our model architecture comprises eight CNNs, consisting of one room type classifier modified using ResNet50 and seven price level classifiers modified using ResNet152. During training, these classifiers were optimized using SGD, with a momentum of 0.9 and weight decay of 0.0001. The room type classifier underwent fine-tuning, and we employed a progressive learning rate schedule, gradually decreasing the learning rates [0.1, 0.05, 0.01, 0.005, 0.0001] to optimize accuracy for approximating room types. We also use train and validation batch size with [64, 128] and pass by dataloader to feed data into the model.

Training Process. Upon initial training, we observed that our validation data exhibited instability and failed to improve, indicative of a potential overfitting issue. To address this, we implemented data augmentation techniques, applying image processing operations such as rotation, horizontal flipping, and cropping to the training data iteratively. This strategy aimed to enhance the model's generalization capabilities, effectively mitigating the rapid increase in training accuracy and slowing down the rate of decline in training loss. Consequently, we observed some improvement in the validation loss. Subsequently, we introduced a dynamic scheduler to adjust the model during training epochs. We experimented with various schedulers, including StepLR[29] for gradually decreasing the learning rate and OneCycleLR[29] for increasing the learning rate from a small value to a larger one before subsequent reduction. Employing grid search, we determined the optimal maximum learning rate for our model. Each training session comprised 100 epochs, during which we monitored key metrics—learning rate, training loss, validation loss, train accuracy, and validation accuracy—using TensorBoard[1]. Model checkpoints were saved every 2 or 5 epochs, facilitating the implementation of early stopping to identify the model with the best generalization ability.

Results. While our approach proved successful in training a highly accurate (87% on validation dataset, epoch 52) room type classifier using the public dataset, we encountered challenges with the price level classifier. Our level classifier under same training strategy doesn't train well. We identified a significant challenge in our dataset where room types within the same price range, such as exteriors, are often accompanied by substantial ambient noise, such as greenery or other common features. This pervasive noise complicates the task of capturing key information accurately. Additionally, we observed that rooms across different price ranges may exhibit similar configurations, potentially leading to the oversight of nuanced differences by convolution operation. These similarities present a formidable obstacle in accurately discerning subtle distinctions in room types and price ranges. This led us to investigate potential sources of noise and outliers within the dataset. Subsequent analysis highlighted the need for further data cleaning and normalization to optimize the performance of the price level classifier.

5.3 Pre-trained GPT-2 Model

Data Preprocessing. Initially, we extracted textual data from the dataset, focusing on the advertisement paragraphs and additional amenities listings. Rather than directly tokenizing the textual data, we performed prompt engineering by appending the string "The sold price is" at the end of each textual sentence to guide GPT-2[26] towards producing more accurate results. Subsequently, we utilized the GPT-2[26] tokenizer to transform the sentences into token input vectors and generated corresponding attention masks.

Model Configuration. As we fine-tuned the pre-trained GPT-2[26] model, we largely maintained the model's hyperparameters unchanged. However, we increased the vocabulary size by one to accommodate the <pad> token required during the tokenization process. Given that we set house price intervals at every $100,000, we established a total of 801 classes, with the output dimension of the final fully-connected layer also set to 801. The loss function employed is Multiclass Cross Entropy Loss. We utilized the AdamW optimization function, a variant of the Adam optimizer that decouples the weight decay process. The batch size of the dataloader was set to 32, and we experimented with learning rate ranges of [5e-5, 1e-4, 1e-3].

Training Process. We conducted three epochs of training on the model with different learning rates. By comparing the training and validation accuracies, we identified the learning rate that exhibited the best performance and proceeded with 10 epochs of training from the initial model to obtain the fine-tuned new GPT model.

Results. Regarding the choice of learning rates, we observed that a learning rate of 1e-3 yielded the best performance in terms of training accuracy, with an approximate 3% improvement for each epoch of training. However, during the subsequent 10 epochs of training with a learning rate of 1e-3, although the training accuracy improved as expected, there was negligible enhancement in the validation accuracy, which remained just shy of 20%. This discrepancy indicates a clear case of overfitting. Upon analysis, we attribute this situation to the lack of crucial features within the textual information collected, that the advertisements may contain exaggerated descriptions of the houses, and the additional amenities may not significantly impact the house prices. Consequently, the model likely only learned features with minimal impact, failing to capture the principal features, thereby resulting in the current overfitting scenario.

5.4 Concatenated Features for the Last MLP Layers

The final part of our model involves integrating features from three pre-trained models, each serving as a feature extractor without further tuning. The training dataset is more recent than the data used for pre-training, as detailed in Table 1. The training configuration mirrors that of the MLP model, employing a similar architecture but adjusted for the different dimensions of the layers.

Model Configuration and Training Process. The input to the final MLP layers is composed of three feature vectors, each emanating from the last feature layer of the respective models. The MLP model outputs a feature vector of length 32. For the CNN model, the last feature layer of the price range classifier outputs 2048 features, with one image feature vector per room type from the room type classifier, resulting in a total vector size of 2048×7. The GPT-2 model outputs a feature vector of size 768. The total input size for our final MLP layers is $32 + (2048 \times 7) + 768 = 15136$.

The architecture includes two hidden layers. A grid search was employed to optimize the configuration from choices such as [(256, 128), (256, 64), (128, 64)], learning rates of [0.01, 0.001], batch sizes [64, 128], and without the use of dropout. The loss function used is Mean Squared Error (MSE), and the optimizer is the Adam optimizer. A grid search with 5-fold cross-validation was applied to identify the optimal hyperparameters, with the best configuration found to have neuron counts of (256, 64) for the two layers.

Results. As presented in Table 3, there was no improvement in performance compared to our standalone MLP model and the baseline model. In fact, there was a slight decrease in accuracy for the ER < 5%. Further investigation suggested that the independent pre-training of models for images and textual descriptions might not have effectively captured salient features. For instance, visually similar images or descriptions might vary significantly in price due to location differences or agent-driven embellishments in text. To address this, we propose a joint training approach where image features could integrate with location data from the MLP model, and textual features could highlight critical issues impacting price. Unfortunately, due to computational constraints, this integrated training approach was not feasible within the scope of this project.

Models	Feature Set	MAE	SMRE	ER Value			
				median	<5%	<10%	<20%
XGBoost	MLS_ENV	97088.87	195757.1	5.61%	45.26%	73.75%	95.32%
MLP	MLS_ENV	100235.87	157491.49	5.07%	49.52%	72.18%	93.52%
Final Model	MLS_ENV	83306.94	123220.2	4.83%	48.21%	71.39%	93.44%

Table 3: Evaluations of Baseline Model and Final Model

6 Conclusions

In this paper, we explored the integration of multimodal data to enhance house price prediction accuracy in Toronto's competitive real estate market. Our study combined structured, visual, and textual data, employing advanced neural networks such as MLPs, CNNs, and GPT-2. Despite the sophisticated methodology and comprehensive data approach, our study did not achieve the expected improvement in prediction accuracy. This investigation highlighted several critical aspects: First, the complexity of integrating and processing diverse data types in a unified model presents significant challenges, particularly in terms of data quality and compatibility. Second, while the use of multimodal data is theoretically advantageous, the practical implementation and tuning of such models require careful consideration of the underlying market dynamics and data characteristics.

Our research provides insights into the potential and limitations of using multimodal data in real estate valuation. The study underscores the necessity for further research to optimize data preprocessing techniques and explore more robust model architectures that could more effectively capture the nuances of the market. Looking ahead, we aim to refine our approach by exploring more adaptive models that can dynamically integrate changing market conditions and data streams. Further research will also investigate the application of these insights in related areas such as market trend analysis and real estate investment strategies.

The description of contribution is in the appendix A.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Eman Ahmed and Mohamed Moustafa. House price estimation from visual and textual features. *CoRR*, abs/1609.08399, 2016.

[3] Statistics Canada. Census of population. `https://www12.statcan.gc.ca/census-recensement/index-eng.cfm`, 2024. Accessed: 2024-04-18.

[4] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014.

[5] Xiaochen Chen, Lai Wei, and Jiaxin Xu. House price prediction using LSTM. *CoRR*, abs/1709.08432, 2017.

[6] Sarkar Snigdha Sarathi Das, Mohammed Eunus Ali, Yuan-Fang Li, Yong-Bin Kang, and Timos Sellis. Boosting house price predictions using geo-spatial network embedding. *CoRR*, abs/2009.00254, 2020.

[7] King's Printer for Ontario. School information and student demographics - dataset - ontario data catalogue. `https://data.ontario.ca/en/dataset/school-information-and-student-demographics`, 2024. Accessed: 2024-04-18.

[8] Guangliang Gao, Zhifeng Bao, Jie Cao, A. Kai Qin, Timos Sellis, and Zhiang Wu. Location-centered house price prediction: A multi-task learning approach. *CoRR*, abs/1901.01774, 2019.

[9] Margot Geerts, Seppe vanden Broucke, and Jochen De Weerdt. A survey of methods and input data types for house price prediction. *ISPRS International Journal of Geo-Information*, 12(5), 2023.

[10] Zillow Group. What is a zestimate? `https://www.zillow.com/z/zestimate/`, 2024. Accessed: 2024-04-18.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Google Inc. Google maps platform documentation | geocoding api. `https://developers.google.com/maps/documentation/geocoding`, 2024. Accessed: 2024-04-18.

[13] HouseSigma Inc. Why are some estimated values not correct? `https://housesigma.com/blog-en/faq/change-data-on-housesigma/the-estimated-values-are-not-correct/`, 2022. Accessed: 2024-04-18.

[14] Mapbox Inc. Geocoding v5 api: Api docs. `https://www.compareschoolrankings.org/`, 2024. Accessed: 2024-04-18.

[15] Fraser Institute. School ranking. `https://www.compareschoolrankings.org/`, 2024. Accessed: 2024-04-18.

[16] Shashi Bhushan Jha, Radu F. Babiceanu, Vijay Pandey, and Rajesh Kumar Jha. Housing market prediction problem using different machine learning algorithms: A case study. *CoRR*, abs/2006.10092, 2020.

[17] Zona Kostic and Aleksandar Jevremovic. What image features boost housing market predictions? *CoRR*, abs/2107.07148, 2021.

[18] Stephen Law, Brooks Paige, and Chris Russell. Take a look around: Using street view and satellite images to estimate house prices. *ACM Transactions on Intelligent Systems and Technology*, 10(5):1–19, September 2019.

[19] Yansong Li, Paula Branco, and Hanxiang Zhang. Imbalanced multimodal attention-based system for multiclass house price prediction. *Mathematics*, 11(1), 2023.

[20] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017.

[21] CH. Raga Madhuri, G. Anuradha, and M. Vani Pujitha. House price prediction using regression techniques: A comparative study. In *2019 International Conference on Smart Structures and Systems (ICSSS)*, pages 1–5, 2019.

[22] Ali Nouriani and Lance Lemke. Vision-based housing price estimation using interior, exterior & satellite images. *Intelligent Systems with Applications*, 14:200081, 2022.

[23] Hao Peng, Jianxin Li, Zheng Wang, Renyu Yang, Mingzhe Liu, Mingming Zhang, Philip S. Yu, and Lifang He. Lifelong property price prediction: A case study for the toronto real estate market. *CoRR*, abs/2008.05880, 2020.

[24] Omid Poursaeed, Tomáš Matera, and Serge Belongie. Vision-based real estate price estimation. *Machine Vision and Applications*, 29(4):667–676, 2018.

[25] Omid Poursaeed, Tomas Matera, and Serge J. Belongie. Vision-based real estate price estimation. *CoRR*, abs/1707.05489, 2017.

[26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[27] Sina Jandaghi Semnani and Hoormazd Rezaei. House price prediction using satellite imagery. *CoRR*, abs/2105.06060, 2021.

[28] Hemlata Sharma, Hitesh Harsora, and Bayode Ogunleye. An optimal house price prediction algorithm: Xgboost. *Analytics*, 3(1):30–45, January 2024.

[29] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. *CoRR*, abs/1708.07120, 2017.

[30] TTREB. Toronto regional real estate board. `https://trreb.ca/`, 2024. Accessed: 2024-04-18.

[31] Darniton Viana and Luciano Barbosa. Attention-based spatial interpolation for house price prediction. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '21, page 540–549, New York, NY, USA, 2021. Association for Computing Machinery.

[32] Pei-Ying Wang, Chiao-Ting Chen, Jain-Wun Su, Ting-Yun Wang, and Szu-Hao Huang. Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism. *IEEE Access*, 9:55244–55259, 2021.

[33] Cort J. Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30(1):79–82, 2005.

Appendices

A Description of Individual Contribution

Jie Peng collected the MLS data; he also collected extra data (e.g., census, school, transit, market trends), handled the data collection and preprocessing, and worked on extracting features from the structured data. He experimented with the impact of feature contributions from listing prices and those collected extra data.

Hanrui worked on extracting features from property images, experimenting with and developing the idea of using classifiers to handle property photos, then using those features to classify the price within given ranges.

Zixuan worked on extracting features from the textual descriptions of the listings, experimenting with and adapting a pretrained GPT-2, with some input modifications, to classify the price within given ranges.

In the end, we worked together to tune the last FC layers of our final model.

Paper Writing:

- **Abstract:** Hanrui
- **Introduction:** Zixuan
- **Related Work and Data Analysis:** Jie Peng
- **Methodology and Experiments:** Jie Peng, Zixuan, Hanrui
- **Conclusion:** Hanrui

GPT checked our grammar and polish our writing.