# Predictive Analysis of Armenian Real Estate Prices: A Deep Learning Approach

Authors: Nemrout Safarian & Petros Tepoyan

Supervisor: Dr. Arman Asryan

*B.S. in Data Science*
*American University of Armenia*

***Abstract*— Past research has successfully shown that apartment prices are often influenced by size of apartment, presence of balcony, presence of shopping malls, and other significant factors. Although these characteristics are very important and impactful, there is still room for innovation and exploration when it comes to real estate price prediction. To this end, this study introduces an innovative approach to conduct a predictive analysis of apartment prices: the encapsulation of image data with traditional tabular datasets. By combining the visual features of an apartment with its categorical and numerical data, research aims to level up the understanding of property valuation which can essentially lead to a more precise and inclusive price prediction.**

***Keywords*—Predictive Analysis, Armenian Real Estate Market, Neural Networks, Deep Learning, Economic Indicators, Armenian Realty Market, Housing Price Prediction, Geolocation Engineering**

## I. INTRODUCTION

Real estate price prediction is a very important task for apartment buyers, apartment sellers, as well as mortgage and housing organizations. In the data-driven world where there are various real estate platforms, it is crucial for individuals and organizations to make use of it. Apartment buyers, for example, are particularly interested in knowing the estimated prices for them to be able to plan and prepare for their purchases [1]. When it comes to apartment sellers, on the other hand, understanding the current trends in the market is the goal. This can help them propose a more proper and appropriate price [1]. Finally, different mortgage and housing corporations can adjust their policies to give those who have economic disadvantages the chance to own an apartment [1].

Recently, the Armenian realty market has had a great jump, and this is because in the country, housing is not only considered to be buying an apartment for personal use, but it is also a clever investment strategy [2]. Hence, for all the interested parties, having a model that can evaluate and predict apartment prices can be an irreplaceable tool which will save their time, requiring minimum to none effort. The purpose of this study is to build models that will be inclusive enough to involve different characteristics and features of an apartment, and produce a price prediction for it.

Another advantage of having such models is to get insights from historical data. Behind every Machine Learning or Deep Learning model, there is a very thorough and meticulous data exploration process which can unleash driving factors and reasons for different phenomena. The realty market is very dynamic and sensitive, and it can be affected by political, economic, and social factors. Armenian realty market is no exception. In his study, Sargsyan [2] argues that the conflict in Ukraine was a driving factor for major relocations to Armenia both from Ukraine and Russia. Because of this rapid growth in population, there was a very rise in rent and sales prices in the realty market.

There are different approaches and models when it comes to apartment price prediction. Limsombunc et al.'s study [3] demonstrated that one of the most common models is the hedonic price model. According to the theory behind this model, customers tend to buy apartments that encapsulate attributes which would potentially maximize their utility functions [3]. This theory suggests that an apartment can be represented as an "aggregation" of individual attributes. Some other attributes are number of bathrooms, number of fireplaces, and parking facilities Characteristics such as number of rooms, location, and size are among the most commonly used attributes for price prediction according to this model. [3]. In the original hedonic theory, however, the external factors of the apartment were not considered [1]. In other words, room number and size could be considered, but the location of it – which is an external characteristic – was not being considered [9]. However, this initial version of the theory was shown to be not representative enough of an apartment, as external factors such as location can in fact impact the price [1]. This is why Frew and Wilson [10] went on to modify the model by adding location as an attribute [1].

Multiple other Machine Learning algorithms have been developed throughout the years for apartment price prediction. RF, SVM, and GB have been common options, however some other researchers have also experimented with neural network solutions [1]. Wang et al. [5], for example, experimented with SVM by creating 2 models -- one with default parameters and another one with particle swarm optimization (PSO) [5]. A backpropagation neural network was also created separately for comparison. The SVM model ended up showing better results than the neural network one, the PSO-SVM model having the better results [5].

Recently, Hong et al. [6] conducted an evaluation of attributes in apartment price prediction using RF and compared it to the hedonic model discussed earlier [1]. For this specific experiment, the least squared method was used [1]. The results showed that RF had an advantage and was able to capture the non-linearity issue better than the hedonic model [1]. Some other studies focused on boosting-based solutions instead. For predicting apartment prices in a city called Karachi in Pakistan, Extreme Gradient Boosting was used, and it was a huge success [7].

A very innovative project was done by Zhao et al. [8] During their study, they substituted the last layer of a convolutional neural network with an XGBoost model [1]. They worked with both numeric/categorical data and image data [1]. A convolution neural network was able to extract features from the images, yet the newly added XGBoost was the final "tunnel" through which the processed data went, resulting in the final prediction of the apartment price [1]. Here, a boosting algorithm was able to outperform the other algorithms [1]. This is an exemplar process which goes to show that an existing algorithm can always be altered and adjusted to particular needs and conditions. Moreover, this is also a great example of combining different algorithms for better and more inclusive results.

Times Series Analysis is another approach that is quite common when it comes to apartment price prediction. Moreover, very often, it is used in conjunction with deep learning solutions. For example, lately, Chen et al. [11] was able to predict apartment prices in the four largest cities in China using only numerical data from 2004 and 2016 [1]. The authors used an RNN model and a Long Short-Term Memory (LSTM) model [1]. The final product of this combination ended up showing much better results than the traditional approach – that is, the autoregressive integrated moving average (ARIMA) [1].

However, it is important to note that although, in terms of apartment price prediction, the combination of deep learning and time series forecasting has demonstrated an advantage compared to more traditional statistical approaches and methods, this hasn't yet been verified for traditional apartment price prediction. In this case, models like XGBoost are more likely to yield better results.

Deep Learning solutions, in general, have become very popular recently due to their great ability of dealing with a large amount of data. In their study, Sirignano et al. [13] suggested a neural architecture with a softmax function to forecast the mortgage risk based on more than 120 million loan datapoints in the USA [4]. A similar study was done by Poursaeed et al. [12] where the valuation of apartments was predicted taking into account their external and internal appearances [4]. Clearly, this study was done after the hedonic theory was updated, as the involvement of external characteristics of an apartment had become more and more apparent and necessary. Frameworks like this use the convolutional neural network DenseNet [14] to classify images of apartments to their corresponding categories, such as bathroom, bedroom, and kitchen [4].

To sum up, there is more than enough evidence to suggest that when conducting apartment price analysis, one certain approach might not be enough. Combining, adding, replacing are all viable options when it comes to working with algorithms and models. Moreover, it is also clear that external characteristics of an apartment are just as important for its valuation and should be fully engaged in the model building process.

In this study, we aim to build and analyze deep learning models that will take into account both tabular data and image data scraped from three different Armenian real estate websites. Although we aim to achieve as good results with the models as possible, our main goal is to compare and contrast these models with each other, along with their advantages and disadvantages. We review the previous research studies to get acquainted with different approaches and methodologies to come up with a well-researched and appropriate project that can possibly be considered as the revamped version of the previous attempts in this topic.

## II. Data

### A. Data Acquisition

For the sake of this analysis, we have scarped data from from three Armenian real estate platforms: MyRealty [15], Bnakaran[16], and Bars [17]. The data encapsulates both tabular data and image data. For scraping the data, we chose the Python programming language for its flexible and robust libraries. More specifically, initially, we employed the Selenium library. This specific choice was influenced by the fact that the page indices of the platform were not directly represented in the page URL. Therefore, we needed a more dynamic interaction with the websites. Later, as we moved forward with scraping, we noticed that the page indices could be found inside the HTTP request. This allowed us to continue the work quicker, as we were able to switch to BeautifulSoup instead which led to a more optimal scraping process.

### B. Data Description

After successfully scraping the 3 datasets, we then had to merge them all to have one entity of observations. The merged version of these 3 datasets had the following structure:

- **Source**: Identifies the website from which the data was obtained.
- **ID**: Unique identifier for each observation.
- **Price**: Listed price of the apartment.
- **Area**: Total area of the apartment in square meters.
- **Rooms**: Number of rooms in the apartment.
- **Floor** and Storeys: The floor number of the apartment and the total number of storeys of the whole building.
- **Building Type**: The type of building categorized in various types.
- **Condition**: Overall condition of the apartment.
- **Bathroom Count:** Number of bathrooms in the apartment.
- **New Building:** Indicator whether the apartment is new.
- **Features:** Binary indicators for amenities (Internet, Heating System, etc.)

### C. Data Cleansing and Preprocessing

After successfully obtaining the data from the 3 platforms, a meticulous process of data cleansing and refining started taking place. We mapped characteristics such as building type, construction type, and apartment condition among the datasets to maintain a uniform structure. In addition, we had to remove those attributes that were not apparent in all the datasets (neither could they be obtained through feature engineering), such as bedroom count. We then proceeded to feature dummification where we had to adjust this methodology to our needs. Specifically, we ended up keeping the last category, as opposed to the common practice. This is because our dataset does, in fact, have apartments which do not have any facilities.

It is important to note that after our thorough research, we made the decision to include the geolocation of the apartments as well. Bnakaran [16] had the latitude and longitude for each apartment which we were able to scrape. These details, however, were missing in the other two platforms [17,18]. To this end, Open Street Maps (OSM) was employed to obtain the latitude and longitude of the apartments from those platforms. We provide OSP with the address of the apartments and this step resulted in two scenarios: one scenario was when

OSM returned points on the map. The second scenario yielded a polygon. In this case, we chose the centroid of the polygon to determine and latitude and longitude.

For the sake of data consistency, we also employed the thread locking mechanism which eliminated data redundancy and repetition. This process, known as synchronization, was a defining step towards obtaining a consistent and well-arranged dataset.

Another branch of our preprocessing was geospatial intelligence. Three services were introduced into our study: GeoService, AddressToCoordinateConverter, and MapFeatureAggregator. The first one was responsible for interactions with OSM. This service took care of address-to-coordinate conversion, as well as identifying amenities near the apartments. AddressToCoordinateConverter is very similar to GeoServices, but it is even more advanced. This service can work with text data using fuzzy matching. For instance, this service allowed us to match "Baregamutiun" to "Barekamutyun."

Finally, MapFeatureAggregator was integral for contextual location-based features. It successfully detected significant locations and created a radius to aggregate amenities close to the apartment.

As mentioned earlier, we also scraped images of apartments. During this process, we particularly encountered three different scenarios:

- Apartments with at least one image.
- Apartments without images.
- Apartments with images but no listed price.

Because our deep learning model required one image and row from the tabular data, we had to create a subset of the initial dataset where we exclusively had observations that had both at least one image and an associated price. When it comes to outliers, we introduced a threshold of $Z > 4$ to detect an outlier. This step was integral in ensuring reliable and unbiased data.

The final stage of preprocessing included a resolution of a critical challenge regarding the apartment pricing data among the datasets. Some listings had the full price of the apartment as their "price" attribute, while some others had price per square meter (psqm).

To tackle this issue, we form an underlying assumption that any price that is less than \$30,000, refers to the price per square meter for the apartment, and any price above than \$30,000, refers to the full price of the apartment. Figure 1. portrays this situation quite clearly. As we can see in the plot, first of all, the majority of apartments are located on the right side of the threshold. Also, obviously, the area cannot logically participate in predicting apartment price per square meter; rather, it can help predict the full price of the apartment. For these reasons, under our underlying assumption, we then converted those prices that were associated with square meter by multiplying them by the area of the apartments to get the full price. After this step, we conducted a very thorough review of all the datasets to ensure that the pricing data was accurate. For these reasons, under our underlying assumption, we then converted those prices that were associated with square meters by multiplying them by the area of the apartments to get the full price.


*Fig. 1*

After this step, we conducted a very thorough review of all the datasets to ensure that the pricing data was accurate.

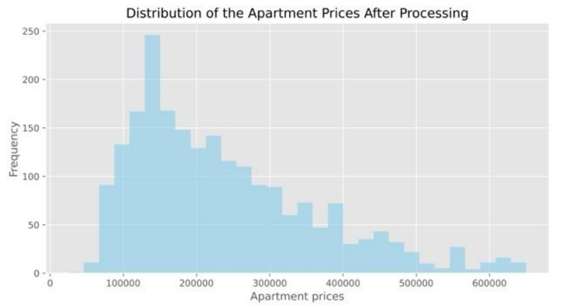


*Fig. 2*

In Fig 2., above, we can see the adjusted version of the prices. Obviously, we have a unimodal distribution now. In addition, the distribution appears to be somewhat normal, and it is easier to interpret overall. Finally, the data points now cluster around the mean price, which means that the variance among the datapoints reflects the market variations.

### D. Data Summarization and Visualization

Visualizing data always can be a very rewarding experience, as it can unleash interesting insights quickly and easily. It is also a form of story-telling that is integral in communicating outcomes and findings to the target audience.


*Fig. 3*

In this section, we aim to visualize and summarize the previously cleansed data to demonstrate some of its characteristics. In Fig. 3, we can see that there is a positive correlation between apartment area and apartment price. Also, we can see that the datapoints are much denser at the lower part which indicates the fact that most of the apartments are on the more affordable side.

Fig. 4, on the other hand, reveals that most apartment tend to be on the lower floors, as opposed to those on the higher floors which are less common. Another assumption is regarding elevators, those buildings that have no elevators tend to be clustered around occupied low-floor apartments.

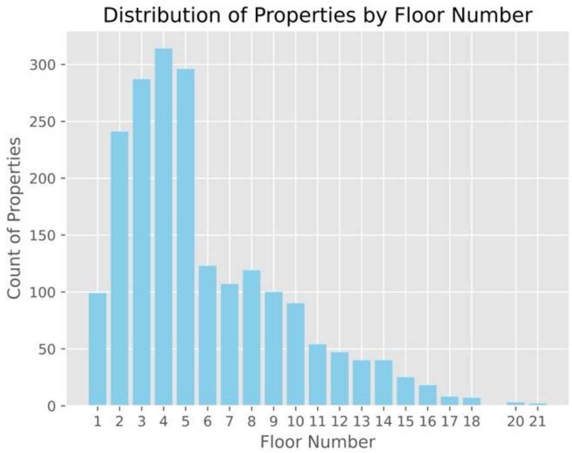


*Fig. 4*

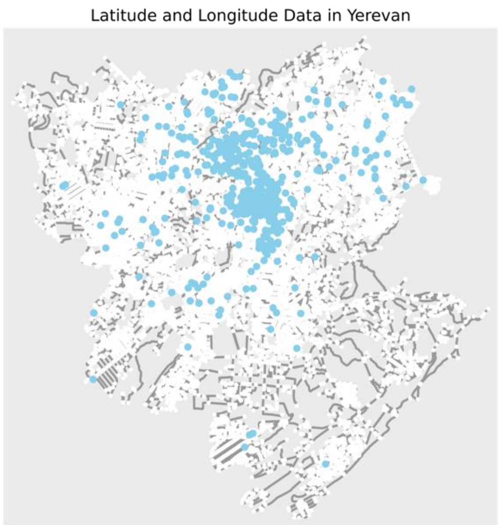


*Fig. 5*

As mentioned earlier, latitude and longitude are among the important attributes of apartments when it comes to price prediction. In Fig. 5, we can see that there is a clearly high density of apartments in the center of Yerevan. We can also see some isolated points on the map which refer to some sub-urban areas where there are not many apartments. These can also be industrial areas or parks. Usually, maps like these can also be very useful for real estate professionals for them to have a better understanding of the market and have an idea of the spread of property types within a specific area.

In Fig. 3, we already noticed the positive correlation between apartment price and apartment area. Fig. 6 also confirms that which makes sense because larger properties obviously cost more. In addition, there is a positive correlation between apartment area and number of rooms. Some of the less significant correlations are those of floors and storeys with price. This suggests that the former might not be as significant when predicting the latter.

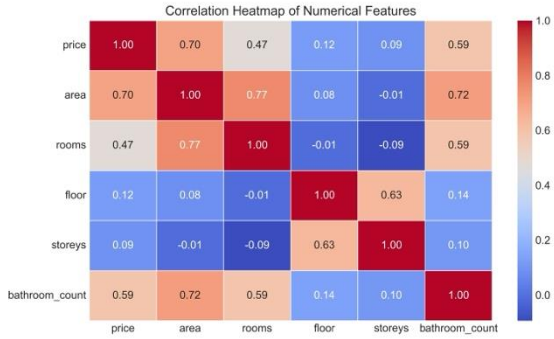


*Fig. 6*

Fig. 7 reveals to us that the median price of new buildings is higher than that of the old buildings, suggesting that the new buildings tend to be more expensive. Also, the height of the boxes suggests that for both types of buildings, there is a diverse range of prices. Nevertheless, the range of the new building is wider which means that here we have a bigger variability of prices. Finally, the whiskers of the plot show that if we compare the highest priced old buildings and the highest priced new buildings, the latter are more expensive. This goes to show that overall, the most expensive buildings tend to be newly built constructions.

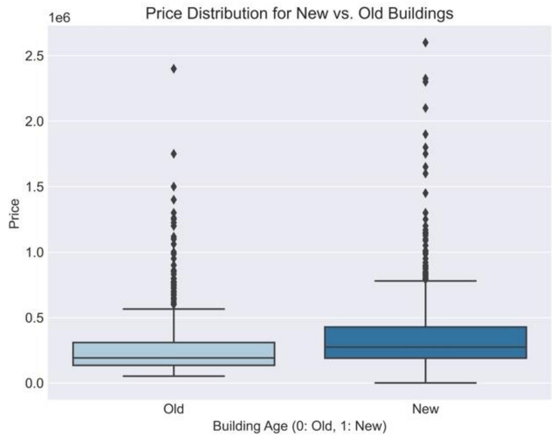


*Fig. 7*

## III. METHODS

### *A. Model Selecton*

Past research has shown that using XGBoost, RF, and similar algorithms gives adequate results when it comes to real estate price prediction. When it comes to our research, it incorporates image data along with tabular data, which means that experimenting with traditional Machine Learning approaches might not be sufficient for us. This is why we came up with the following scenarios:

- Classical models
  - Pros: Fast, reliable, known to produce good results, easy to do grid research with.

- Cons: Not robust enough, does not leverage images.

- A Deep learning model with images and tabular data.
    - Pros: Leveraging images, may potentially be the most robust model.
    - Cons: Difficult and long training, not guaranteed to reach good results, the model can become too heavy.
- A Deep Learning model with tabular data and without images.
    - Pros: Fast, not very much reliable, not guaranteed to produce good results.
    - Cons: More difficult to handle overfitting, much more difficult to do hyper parameter tuning.

In terms of classical models, we experimented with XGBoost, Random Forest, Linear Regression, or KNN, Ridge, Lasso, Bayesian Ridge, Elastic Net, and Decision Trees. With these models, our purpose was to first experiment with approaches that are more traditional and conventional. In other words, we thought that trying out some of these algorithms would give us a general understanding about how our datasets can potentially behave when put in use. Again, given the very diverse nature of our data, our main goal was to experiment with different approaches to get to the more optimal direction, and working with classical models was our first step towards this journey.

When it comes to experimenting with a deep learning that will not involve apartment images, our purpose here is to just improve upon the classical models. For this scenario, we are going to use Feedforward neural netwoks. Pros and cons are:

- **Wide**: Few layers and a big number of neurons in each layer.
- **Deep**: Many layers, but not as many neurons in each layer.
- **Both**: The combination of Wide and Deep.

Finally, in the case of the Deep Learning model that will include images, our purpose is mainly experimentation, rather than a high accuracy score. Although we do aim to achieve a high score, our bigger goal with this experiment is to find out which methods can or cannot work when it comes to apartment price prediction and how data scraped from real estate websites can be used in Deep Learning. In addition, our research study is mainly focused on comparing and contrasting different methods. In Fig. 8, we visualized the conceptual representation of how we imagine the third model – that is, the Deep Learning involving images – can be constructed. Here we also will employ Inception V4[18].

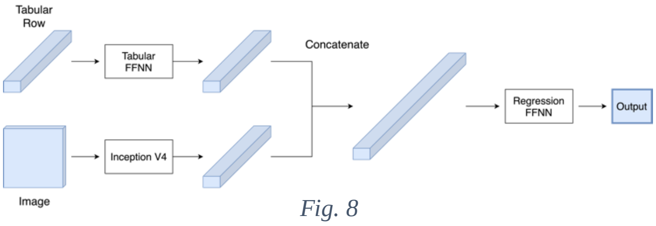


*Fig. 8*

In Fig. 9, we can also see the overall schema of the network [18]. Essentially, this is an image classification model, similar to AlexNet, YOLO, VGGNet, ResNet, etc. We

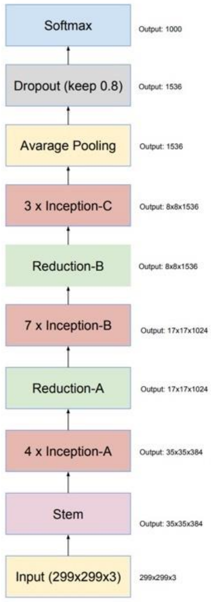


Figure 9. The overall schema of the Inception-v4 network. For the detailed modules, please refer to Figures 3, 4, 5, 6, 7 and 8 for the detailed structure of the various components.

*Fig. 9*

think that this can potentially be a good addition to our study because it is relatively lightweight, yet very robust.

### *B. Model Training*

In this section, we are going to discuss the training process of the models. We will cover all the three approaches separately – that is, the classical approach, the Deep Learning approach with images, and the Deep Learning method without images.

In terms of the classical method, the training process was quite usual -- in the beginning, we initialized the different models (XGBoost, Lasso, RF, etc), then we trained them using the subsets X_train and Y_train which later on was also evaluated with the respective X_test and Y_test. However, it is important to mention that this was the very initial version of the process. The next step was the extensive grid search that we conducted. For this process, we introduced hyperparameters with which the grid search algorithm was able to work and find the most optimal combination.

When it comes to the Deep Learning approach with images, in this case, we incorporated GPUs provided by the American University to be able to fulfill the computational demands of the algorithm. The training process was employed on PyTorch. In this case, the dataset was split into training, validation, and test sets with an 80/10/10% distribution. For future optimizations, we are planning to assign different tensors to different computational units.

In the case of our Deep Learning method that includes tabular data but no images, we went with CPU, instead of

GPU. The main reason was that GPU took more time for training, given the specific structure of this algorithm. The training, validation, and test partition was the same as in the previous one – 80/10/10%. A key component of our model training process was assessing losses on both validation and test sets. In addition, we also paid attention to the gradients during the training process to be able to detect vanishing or exploding gradients. For this model we employed the Adam optimizer as well. To tackle overfitting, we also used L2 regularization. We chose L1 Loss function (or, Mean Absolute Error) for its flexibility and robustness in regression-based tasks. We also integrated a learning rate scheduler in order for us to optimize the learning rate. In terms of the number of epochs for this algorithm, it was 300,000. The batch size, on the other hand, was at 64. Overall, we can say that this approach was a combination of computational efficiency and robustness.

## IV. Results & Discussion

As mentioned earlier, for the purpose of this project, we decided to go with 3 approaches. In this section, we are going to go over the results of each scenario, evaluate them, and draw conclusions. We chose the following metrics to evaluate the models:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Starting with the first scenario, here we mainly employed some of the most traditional and classic approaches, such as XGBoost, Ridge, Lasso, Linear Regression, etc.

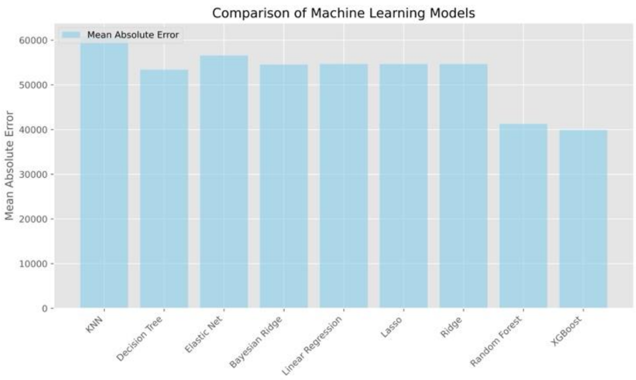


*Fig. 10*

Fig. 10 unveils the absolute error for all the classical models that we employed. As we can see, the least absolute error that we have is for the XGBoost model. For this case, the difference between the actual and the predicted prices of the apartments was $70,000. It is important to note that this result was obtained after hyperparameter tuning. In other words, in terms of the classical models, that was the best result we could obtain. It was the best way we could explain the target variable given the features we had.

In Tab. 1, on the other hand, we see the results of our other metric -- $R^2$. As we can observe, in this case as well, the top result belongs to the XGBoost model, and the second best is Random Forest, followed by Ridge. Overall, both metrics report almost the same result in terms of the classical models. The main conclusion that came to in terms of the classical models is that it is very limited, because this is the result after hyperparameter tuning.

| | Model | Mean Absolute Error | R2 Score |
|---|---|---|---|
| 3 | XGBoost | 39904.261500 | 0.760666 |
| 2 | Random Forest | 41247.893518 | 0.725612 |
| 6 | Ridge | 54653.634731 | 0.636272 |
| 7 | Lasso | 54658.570407 | 0.636187 |
| 0 | Linear Regression | 54659.490380 | 0.636184 |
| 9 | Bayesian Ridge | 54527.613682 | 0.636123 |
| 8 | Elastic Net | 56576.920800 | 0.607954 |
| 1 | Decision Tree | 53388.483061 | 0.529012 |
| 5 | KNN | 60799.187617 | 0.512358 |

*Tab. 1*

For the case of the Deep Learning approach where we don't include images, we tried different architectures, such as Wide and Deep. We also tried batch normalization and dropouts. In addition, we also employed different activation functions, such as plain ReLU, Sigmoid, and Leaky ReLU. However, it is important to note that the plain ReLU outperformed all the other activation functions. We

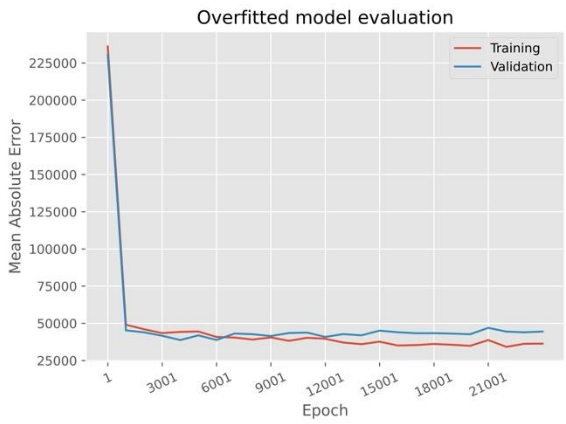


*Fig. 11*

employed one hidden layer and 32 neurons, along with a dropout with a very small probability. In Fig. 11, we can see the overfitting rate for different epochs. The absolute error is around $40,000. We reached this result by having 90% train size. In other words, we can say that by taking all these steps for the case of this deep learning approach, we got a very similar result to that of the classical approach.

For our deep learning model incorporating images, although we were able to utilize the GPU resources provided by the American University of Armenia, we had challenges, we had challenges with the training time. In terms of the epoch threshold, we set it to 10. This limit was a balance between practical constraints and sufficient time for training. Although our model had over hundreds of millions of weights and hundreds of thousands of neurons, it still could not generalize in a feasible amount of time.

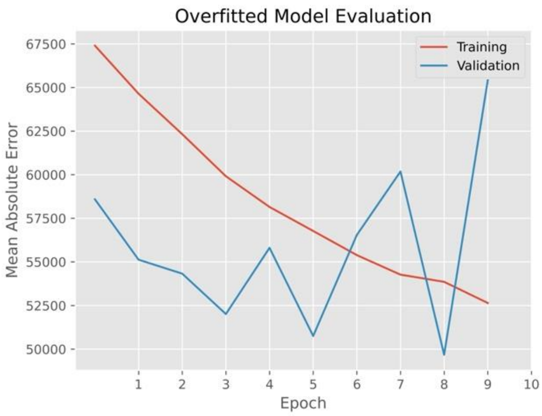


*Fig. 12*

In Fig. 12, we can see the relationship between the absolute error of the model and the number of epochs it processed. According to these 10 epochs, the best the model could do was an absolute mean error of around $49,000. This is if we decide to early stop the training process at epoch 8, after which the model fails to generalize.

Now, although we applied all known techniques to increase accuracy and decrease overfitting, we could not proceed far enough and beat the classical models. But it does not mean they are the best; it means our research next time should focus on getting cleaner data and much more powerful hardware to be able to run robust models for millions of epochs. We found out, that when it comes to the DL model without images, the algorithm requires 300K epochs to converge and start overfitting. In addition, when it converges, it does not converge to something practical - 40K mean absolute difference is still huge and cannot be used in real life to estimate the price of valuable properties. It is also important to note that when we observe that training with images takes long, we need to consider that training without images also takes long, because in that case we need to train for millions of epochs. We make an educated guess that hypertuning the model, getting more quality data (even without images) and training longer will yield a better result than the classical models. We may only guess that for the DL model with images it would require as many epochs. That, in return, requires an ensemble of GPUs ready to train those images.

## V. LIMITATIONS & FUTURE WORK

The journey of predictive modelling is never easy. There are many layers to the process, and some of the challenges might not be possible to overcome when both the time and the resources are limited. In terms of this study, although we conducted this research as thoroughly as we could, there are still some limitations which prevented us from getting the results that we would have liked to obtain. More specifically, the limitations can be divided into two parts: data limitations and model limitations.

### A. *Data Limitations*

Images from real estate platforms most often contain watermarks. Although removing such watermarks is possible, it requires a lot of computational power and efforts. Given our limited time and resources, we made the decision to train the model with the watermarks, “forcing” the models to learn and interpret the watermarks. This, of course, may have impacted the overall performance of the model.

Although the three websites we scraped the data from still were able to provide us with enough to work with, List.am [20] is one of the biggest and most prominent real estate (and not only) websites in Armenia. Hence, being able to incorporate data form this website would have massively benefited the results. However, there is a huge inconsistency on the website in terms of its data structure and scraping it and adjusting it to the needs of this research project would not have been optimal.

The prices of apartments in Armenia now are heavily influenced by the current geopolitical situation, as well as the regional conflicts in the country. Hence, it was not possible for us to build a model that would be able to generalize the prices as well as we would want it to. In other words, the market volatility is very sensitive and quite unstable right now, so this was another challenge without which the results would have been better.

### B. *Model Limitations*

Although we had the amazing opportunity to make use of the GPU resources provided by the American University of Armenia, the computational demands of the models – such as Inception V4 or Inception V3 – were still challenging to fulfill. It took almost 10 minutes for one epoch to train and for us to achieve convergence, we needed hundreds and thousands of epochs which is impractical.

Although including images in any kind of predictive analysis can be a very rewarding and eventful experience, it also comes with its challenges and disadvantages. The integration of the images into our model was no exception. While this step made our analysis much more inclusive and diverse in terms of the data being used, it also increased the training time very much and demanded much more efforts from us when it came to coming up with different methods and techniques to make this integration as effective as possible.

Of course, because of such limitations, our model did get affected. We had to change and adjust some of our initial plans and roadmap attributes to these limitations. Moreover, along the way, we also had to accept a certain level of noise in our data for the sake of not losing too much data by dropping observations that could potentially be valuable for us. In other words, we had to make compromises in order to reach an optimal performance in terms of the model.

## C. Future Research

In future research endeavors, encapsulating more computational power into the process could be integral. Whether it is a bigger number of GPU resources, or multiple different devices, enriching our set of resources in terms of computation can change the trajectory of the project.

The Deep Learning results of this research study showed that incorporating Inception V4 and Inception V3 can greatly improve the training process, especially the ones that did not involve image data. These models can potentially even be a baseline or a starting point for another model which will involve images. However, in order to achieve this, we will not only need more neural networks and more computational power, as mentioned earlier, but also more data. As much as scraping List.am [20] is a complex process, it is still undeniable that it can provide with data that is very valuable for this analysis and representative of the Armenian retail market. Another task that can be done in the scope of the future research of this project is managing to successfully scrape the platform.


## Acknowledgments

We would like to extend our gratitude to the American University of Armenia. Specifically, we thank David Davidian for providing the institution with GPU resources which played an integral part in our research study.


## References


[1] H. Zhang, Y. Li, and P. Branco, “Describe the house and I will tell you the price: House price prediction with textual description data,” *Natural Language Engineering*, pp. 1–35, Jul. 2023, doi: 10.1017/s1351324923000360.

[2] D. Sargsyan, “Predicting Apartment Prices in Yerevan: A Machine Learning approach and Real estate Market analysis,” *ResearchGate*, Jul. 2023, [Online]. Available: https://www.researchgate.net/publication/372807625_Predicting_Apartment_Prices_in_Yerevan_A_Machine_Learning_Approach_and_Real_Estate_Market_Analysis

[3] V. Limsombunc, C. Gan, and M. Lee, “House Price Prediction: Hedonic Price Model vs. Artificial Neural Network,” *American Journal of Applied Sciences*, vol. 1, no. 3, pp. 193–201, Mar. 2004, doi: 10.3844/ajassp.2004.193.201.

[4] B. K. De Aquino Afonso, L. C. Melo, W. Dihanster, and L. Berton, “Housing Prices Prediction with a Deep Learning and Random Forest Ensemble,” ResearchGate, Sep. 2019, [Online]. Available: https://www.researchgate.net/publication/335527230_Housing_Prices_Prediction_with_a_Deep_Learning_and_Random_Forest_Ensemble

[5] Wang, X., Wen, J., Zhang, Y. and Wang, Y. (2014). Real estate price forecasting based on SVM optimized by PSO. Optik 125(3), 1439–1443.CrossRefGoogle Scholar

[6] Hong, J., Choi, H. and Kim, W.-s. (2020). A house price valuation based on the random forest approach: The mass appraisal of residential property in south Korea. International Journal of Strategic Property Management 24(3), 140–152.CrossRefGoogle Scholar

[7] Ahtesham, M., Bawany, N. Z. and Fatima, K. (2020). House price prediction using machine learning algorithm-the case of Karachi city, pakistan. In *2020 21st International Arab Conference on Information Technology (ACIT)*, IEEE, pp. 1–5.CrossRefGoogle Scholar

[8] Y. Zhao, G. Chetty, and D. Tran, “Deep Learning with XGBoost for Real Estate Appraisal,” *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2019, doi: 10.1109/ssci44817.2019.9002790.

[9] S. Rosen, “Hedonic prices and implicit markets: product differentiation in pure competition,” *Journal of Political Economy*, vol. 82, no. 1, pp. 34–55, Jan. 1974, doi: 10.1086/260169.

[10] J. Frew and B. Wilson, “Estimating the Connection between Location and Property Value,” *Journal of Real Estate Practice and Education*, vol. 5, no. 1, pp. 17–25, Jan. 2002, doi: 10.1080/10835547.2002.12091579.

[11] Chen, X., Wei, L. and Xu, J. (2017). House price prediction using LSTM. *arXiv preprint arXiv:1709.08432*.Google Scholar

[12] O. Poursaeed, T. Matera, and S. Belongie, “Vision-based real estate price estimation,” *Machine Vision and Applications*, vol. 29, no. 4, pp. 667–676, Apr. 2018, doi: 10.1007/s00138-018-0922-2.

[13] J. Sirignano, A. Sadhwani, and K. Giesecke, “Deep learning for mortgage risk,” *arXiv (Cornell University)*, Jul. 2016, doi: 10.48550/arxiv.1607.02470.

[14] “Densely connected convolutional networks,” *IEEE Conference Publication | IEEE Xplore*, Jul. 01, 2017. https://ieeexplore.ieee.org/document/8099726

[15] “Real estate,” *Real Estate*. https://myrealty.am/en

[16] “Apartments in Yerevan, Real Estate in Armenia, houses, office and retail spaces and plots | Bnakaran.com.” https://www.bnakaran.com/en

[17] “Bars Real estate company,” *bars.am*. https://bars.am/en

[18] “Papers with Code - Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” Feb. 23, 2016. https://paperswithcode.com/paper/inception-v4-inception-resnet-and-the-impact

[19] https://www.researchgate.net/figure/Difference-between-inception-v3-and-inception-v2_fig3_350782686

[20] “List.am - Հայաստանի անվճար հայտարարությունների կայք.” https://www.list.am