ST. FRANCIS XAVIER
UNIVERSITY

CSCI - 525: Machine Learning Design

Dr. Jacob Levman

Phase 2 Report

Khan, Salal Ali
202307216 - x2023flb@stfx.ca

Hussain, Moayadeldin
202407556 - x2024ghl@stfx.ca

Javed, Muhammad
202303808 - x2023dvh@stfx.ca

Oct 3, 2024

Abstract

In this work, we introduce a comprehensive study on the Airbnb's 2023 New York City dataset and how machine learning could be introduced to address different questions and enhance either the price prediction, or minimum nights stay, according to the business needs. We showed the feasibiliy of such implementation using the emerging df-analyze package which provides a convenient CLI end-to-end framework for running ML algorithms. Before applying df-analyze, we proposed an extensive preprocessing for the dataset that significantly enhanced the results. We applied 4 different machine learning models, including GANDALF, emerging deep learner for tabular data. The best model for Price prediction is SGD Assoc with 0.149 MAE. Moreover, best model for Minimum Nights prediction is LGBM Embed-Linear. Although GANDALF didn't achieve the best results, we believe it achieved comparable outcome and gives a room for improvement in the future.

Introduction

We are currently analyzing Airbnb's 2023 New York City dataset and are using machine learning to predict property prices. The real estate industry is critical for economic development and society progress, reflecting the aspirations of people, families and the overall social health of the region [1]. In the past, knowing property prices relied heavily on human experience and conventional methods. However, these approaches usually did not manage to success to account for the complexity and non-linear relationships of housing market data. With machine learning advancements, the scene has changed with the ability to extract precious information from large volumes of data [2]. Addressing the challenges, predicting house prices accurately remains demanding due to the multitude of influencing factors, as location, size, house quality, condition, etc. [3]. We are looking forward in our work to acquire knowledge from the literature of different models used and address the existing challenges in such problem.

One of the most interesting research work we found on utilizing AI in Airbnb datasets price prediction was founded in [4] by Stanford Researchers. They used plenty of models to address their target, including Ridge Regression, K-means Clustering, Support Vector Regression, Neural Netowrks and Gradient Boosting Tree Ensemble. They also used Feature Selection and Sentiment Analysis. Sentiment Analysis refers to the application of natural language processing to identify and classify subjective opinions in source materials (e.g., a document or a sentence) [5]. However, this topic is beyond our scope in our Project 1 at the course becuase it depends on transforming the words and sentences to a different feature space using a technique called word embeddings, as Word2Vec [6]. Moreover, the work in [7] focused also on Predicting Airbnb Listing Price with multiple models, utilizing Boston Airbnb open data set from Kaggle, which includes 3, 585 listings between 2016 and 2017. They used Linear Regression, K-nearest neighbor regression and Gradient Boosting regression. The authors in [3] depended on using Linear Regression, and Random Forests. In [2], Linear Regression, Random Forest Regression and Gradient Boosting Regression.

In [4], their experiments involved Mean absolute error (MAE), mean squared error (MSE) and R2 score were used to evaluate their training models. Among the models tested, Support Vector Regression performed the best and produced 69% R2 score, 0.147 MSE on test split, and 0.2132 on MAE. In [3], the Random Forest Regressor model had high accuracy store with 97.3% for training set and 82.3% for the testing set, in comparison with the linear regression model with only 79.4% and 58.9% for training and testing. Hence, the authors concluded that the Random Regression model is the best model for study. In [2], they used MAE, RMSE and R2 Score, mostly similar to work in [4], as performance metrics. The authors in [2] found that R2 scores in their work exceeds that in [4] for two models of the three they used, achieving 0.822 and 0.828 with both Random Forest and Gradient Boosting repsectively.

We hypothesize that the use of open source machine learning software applied to New York AirBnb's 2023 dataset may produce useful technology for House Prices prediction and analysis. We hypothesize that applying different Machine Learning, and Preprocessing technqiues, according also to our discussions with Dr. Jacob, will yield to improvments in property price and minimum nights.

Methods

In this methods section, we present our methodology in data preprocessing, analyzing prices and minimum nights, and how to utilize different encoding techniques and correlation between attributes to optimize our results. The original dataset contains 42,931 records and 18 features as shown in Table 1

The first step to preprocess our dataset was to remove columns that we may consider as redundant or would degrade the performance of our model, becuase of their irrelevancy treating it as noise, or because they wouldn't give too much information according to the discussion we had with Dr. Jacob Levman during his office hours sharing thoughts about the project.

#	Column	Non-Null Count	Dtype
0	id	42931 non-null	int64
1	name	42919 non-null	object
2	host_id	42931 non-null	int64
3	host_name	42926 non-null	object
4	neighbourhood_group	42931 non-null	object
5	neighbourhood	42931 non-null	object
6	latitude	42931 non-null	float64
7	longitude	42931 non-null	float64
8	room_type	42931 non-null	object
9	price	42931 non-null	int64
10	minimum_nights	42931 non-null	int64
11	number_of_reviews	42931 non-null	int64
12	last_review	32627 non-null	object
13	reviews_per_month	32627 non-null	float64
14	calculated_host_listings_count	42931 non-null	int64
15	availability_365	42931 non-null	int64
16	number_of_reviews_ltm	42931 non-null	int64
17	license	1 non-null	object

Table 1: NYC AirBnb 2023 Dataset Information

Hence, we decided to drop *'host_id, host_name, last_review, reviews_per_month, calculated_host_listings_count, number_of_reviews_ltm, license'* attributes.

It is implemented also in the code a function called 'gettingInsights()' which prints out the number of unique elements in 'neighbourhood, neighbourhood_group, room_type' attributes. Moreover, it prints out the unique elements in the last two aforementioned attributes.

As mentioned in the project proposal, our first two main goals is to predict the minimum number of nights one user may spend in one apartment, and also the prices. In order to achieve that with higher accuracy, we decided to remove what we considered as 'outliers' during our data exploration, these outliers would be defined as:

1. Prices: We found out there are 27 records having the label price "zero". We consider this as noise and practically unachievable because basically no one would rent their property in real life for free (unless there are some relationship between the landlord and the tenant that we wouldn't be able to figure out from our features, and something out of the scope and focus of our work.) So we decided to remove all these 27 records.

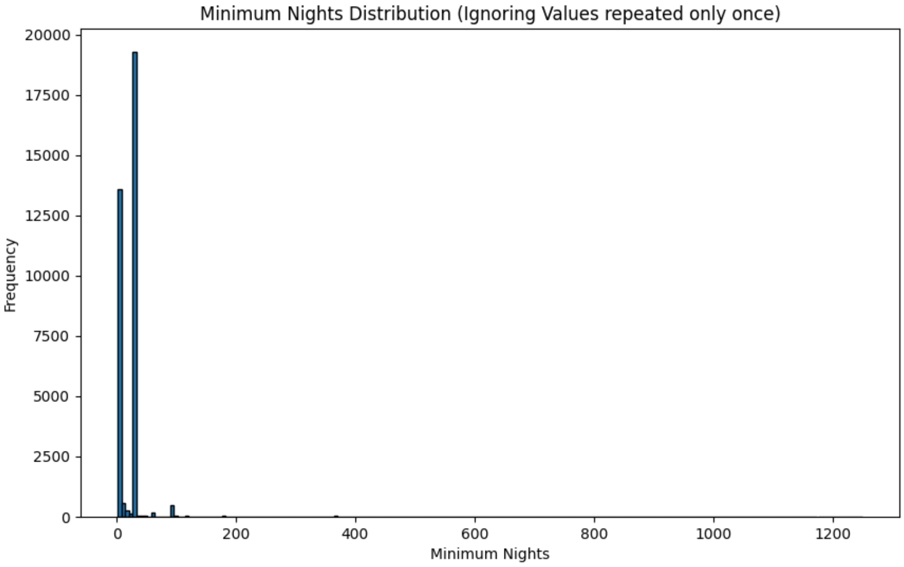

Figure 1: Histogram showing the frequency of Minimum Nights values after preprocessing

2. Minimum Nights: We defined outliers as the number of of nights that appear only once. This means we retail all records where the minimum nights value is repeated at least twice. During our exploration, we found out 7,946 records falling under this criteria. Hence, we decided to remove them.

After cleaning the dataset, the number of records left reduced to 34,977 records, which indicates basically that many houses that were given the label price ”zero” had a minimum nights stay number that was repeated only once, we consider this as another indication of them being truly outliers. The distribution of Minimum Nights attributes after removing the outliers defined in (2) below is shown as in Figure 1.

Another main point in the target of our work is to apply different types of Encodings and Natural Language Processing techniques to see how they may enhance the performance, in our implementation, we choose to apply Label Encoding, One Hot Encoding and Bag of Words technique.

After our discussions with Dr. Jacob, we decided to use Label Encoding technique on the *‘neighbourhood_group’* feature, as the feature with the largest number of unique values, it is convenient to label encode it to avoid dimensionality increasing. In Table 2, the top 5 most frequent labels are shown. It is obvious that for example Bedford-Stuyvesant and Williamsburg are by considerable far the most frequent neighbourhoods in the dataset.

Label Encoded Neighbourhood	**count**
12 (Bedford-Stuyvesant)	2668
216 (Williamsburg)	2385
96 (Harlem)	1806
27 (Cambria Heights)	1570
129 (Midtown)	1407

Table 2: Top-5 Most Frequent Label Encoded Values

Regarding the One Hot Encoding, when we say we store one-hot encoded values directly, we refer to saving an actual one or zero for each element of a vector [8]. We decided we are going to encode both the *'neighbourhood_group'* and 'room_type'. There are 5 Neighbourhood Group Set Elements which are 'Queens', 'Manhattan', 'Staten Island', 'Bronx', 'Brooklyn'. Moreover, there are 4 Room Type Set Elements which are 'Entire home/apt', 'Shared room', 'Hotel room', 'Private room'. Each of these attributes were used resulting in adding 9 additional features to the dataset, each one represents OHE for the specific attribute.

On the Bag of Words technique, we consider our implementation in this area a contribution in this type of work. The bag of words (BOW) method can be employed to treat each remaining word as a feature of the document [9]. According to our discussions with Dr. Jacob, we needed to determine the most relevant words in the names of properties. In order to do so, the literature review in this area usually focus on using some kind of Word Embedding technique to handle such case. However, we thought that in this Project 1 work we may address it in a different, easier and much innovative way.

We decided to pick up the top 100 frequent words in our name column and check for the words that has high correlation with the price feature, we put a threshold of positive/negative correlation (+/-0.02) in order to determine whether this word is relevant or not, we used Pearson's correlation coefficient. Correlation coefficients are utilized to evaluate the strength and direction of linear relationships between variables pairs. [10]. Our approach returned eight different words with highest correlation which were 'luxury', 'hotel', 'floor', 'city', 'private', 'in', 'cozy', 'room'. Afterwards, we applied One Hot Encoding to make each occurrence of a word in a record counts, and finally, we append the result of the encoding to the dataframe as columns.

Results

In Phase 2, we ran df-Analyze only once on one target feature, which was the price. Moreover, we have performed Label Encoding, One Hot Encoding and dropping the redundant features.

In Phase 3, We ran df-Analyze four times. The first two runs focused on predicting the target Minimum Nights and Price features separately, following the full data preprocessing guideline described in Section Methods. Additionally, the third and fourth runs aimed to predict the same target features, but this time after removing the number_of_reviews feature. This suggestion comes from Dr. Jacob suggestion he offered in the feedback we received on Phase 2 results, which can be interpreted also from Table 3.

We compare the results from Phase 2 of the df-Analyze run, which we dedicated on predicting the price feature, with the outcomes from two runs in Phase 3, where we predicted the price feature once including the number_of_reviews feature and once without. We evaluate the performances on the holdout set. The privilege of using df-Analyze [11] is that it allows us to explore such models with different types of data embeddings. Hence, we introduce some of the models for comparison purposes and full results is uploaded among this project's materials. The models we will introduce are LGBM (Light Gradeint-Boosting Machine) with linear embeddings, SGD (Stochastic Gradient Descent) with association, and RF (Random Forest) with pred selection. Moreover to remain consistent with other works provided in the literature, we will refer to Mean Absolute Error (MAE), Mean Squared Error (MSE), and R2 score as our performance metrics.

Phase Number	**Preprocessing Steps**
Phase 2	Label Encoding, One Hot Encoding, Removing Redundant Features.
Phase 3	Label Encoding, One Hot Encoding, Removing Redundant Features, Implementing Bag of Words with Applying Pearson Correlation, Removing outliers from Minimum Nights and Prices, Trying to figure out whether removing 'number of reviews' feature can enhance the results.

Table 3: Comparing Preprocessing Steps for Phases 2 and 3

Phase Number	**MAE**	**MSE**	**R2**	**Method**
Phase 2	0.163	1.664	-0.016	LGBM Embed-Linear
Phase 3	**0.150**	**0.239**	0.065	LGBM Embed-Linear
Phase 3 with removing Number of Reviews	0.158	0.849	**0.074**	LGBM Embed-Linear
Phase 2	0.159	1.608	0.018	RF Pred
Phase 3	**0.152**	**0.235**	0.079	RF Pred
Phase 3 with removing Number of Reviews	0.156	0.838	**0.086**	RF Pred
Phase 2	0.154	1.633	0.002	SGD Assoc
Phase 3	**0.149**	**0.242**	0.052	SGD Assoc
Phase 3 with removing Number of Reviews	0.152	0.831	**0.094**	SGD Assoc

Table 4: df-Analyze Results of Predicting Price Feature

The results are shown in Table 4. It is clear that our innovative approach of introducing Bag of Words with Pearson's Correlation coefficient, along with removing Prices and Minimum Nights outliers have significantly improved the price prediction. It is also obvious that removing 'number_of_reviews' feature didn't improve the loss percentages. However, it has impacted the R2 score with making it getting higher in the three models.

Regarding predicting Prices, we compare the results from Phase 3

Regarding predicting Minimum Nights, we also compare the results from Phase 3 without removing number of reviews, and Phase 3 with removing number of reviews feature, as we didn't run df-analyze on Minimum Nights as a target in Phase 2. From the results shown in Table 5, it is clear that LGBM performs better when 'number_of_reviews' features is removed. On the other hand, it is also clear that SGD performs better when this feature is kept among the dataset. Random Forests MAE loss was better with the feature's presence, and MSE and R2 values were better with the feature's absence.

Phase Number	**MAE**	**MSE**	**R2**	**Method**
Phase 3	0.195	0.241	0.119	LGBM Embed-Linear
Phase 3 without Number of Reviews	**0.187**	**0.181**	**0.231**	LGBM Embed-Linear
Phase 3	**0.194**	0.246	0.100	RF Pred
Phase 3 without Number of Reviews	0.195	**0.206**	**0.126**	RF Pred
Phase 3	**0.230**	**0.274**	**-0.001**	SGD Assoc
Phase 3 without Number of Reviews	0.243	0.276	-0.156	SGD Assoc

Table 5: df-Analyze Results of Predicting Minimum Nights

We can deduce from the results that our further preprocessing in Phase 3 made a huge enhancement in the results compared to Phase 2 in properties price predictions, which shows the robustness of our approach. Moreover, removing 'number_of_features' attribute as per Dr. Jacob suggestion seems to have a negative impact on the losses of prices predictions, but it improves the R2 score. We have shown also from the illustration of our results that our method is competent with other similar and proven work in the literature. Regarding predicting Minimum Nights, we compared both of our Phase 3 runs on this target attribute and it was shown that removing 'number_of_features' attributes may likely produce an enhancement in some models performances.

GANDALF Results

In Phase 4, we are required to run the emerging deep learner devoted to tabular data, known as GANDALF. Deep Learning techniques, in general, isn't known to excel and give state-of-the-art results on tabular data. In [12], it was shown that XGBoost achieves better results than the deep learners across the datasets, including the datasets used in some paper, where they authors proposed the deep models. However, combining deep learners with XGBoost algorithm for example shows promising results.

Here in our work, we compare the results we achieved using GANDALF deep learner with the results we achieved in Phase 3 with the other Machine Learning Models. Training and Testing Process are the most important factor affecting the success of machine learning [13]. Hence, following the same consistent procedures when computing the results is deemed extremely critical. We follow the same approach of comparing the hold-out set in Phase 2 and 3, and compare according to the same loss function.

Model	**Selection**	**MAE**	**MSE**	**R2**
gandalf	assoc	**0.154**	**0.291**	**0.054**
gandalf	none	0.157	0.303	0.015
gandalf	pred	0.173	0.373	-0.214

Table 6: Predicting Price

Model	**Selection**	**MAE**	**MSE**	**R2**
gandalf	assoc	**0.264**	**0.215**	**-0.034**
gandalf	pred	0.287	0.230	-0.108
gandalf	none	0.287	0.231	-0.114

Table 7: Predicting Minimum Nights

Model	**Selection**	**MAE**	**MSE**	**R2**
gandalf	assoc	**0.266**	**0.214**	**-0.030**
gandalf	none	0.279	0.222	-0.068
gandalf	pred	0.424	5.446	-25.244

Table 8: Predicting Minimum Nights without Reviews

Model	**Selection**	**MAE**	**MSE**	**R2**
gandalf	pred	**0.157**	0.300	0.026
gandalf	none	0.163	**0.288**	**0.065**
gandalf	assoc	0.168	0.305	0.010

Table 9: Predicting Price without Reviews

It is obvious in Tables 6, 7, 8 that GANDALF with assoc selection achieves the best performances in terms of MAE, MSE, R2 scores. However, for Table 9 GANDALF with none selection achieves the better MSE and R2 scores.

As there are multiple models on different datasets, we apply Majority Voting and choose GANDALF with assoc selection model to compare with our results in Phase 3.

We provide the updated results tables in the following paragraph. At first, we thought that it would be better to remove Table 4 and 5 from the previous section and provide only the updated version in the manuscript. However, for the sake of illustrating our extensive work through the whole project, we prefer to keep our worked tracked by detail. In any future versions of this manuscript that doesn't take into account the timeline (phases) of the work, this shall be edited to remove redundancy.

Phase Number	**MAE**	**MSE**	**R2**	**Method**
Phase 2	0.163	1.664	-0.016	LGBM Embed-Linear
Phase 3	**0.150**	**0.239**	0.065	LGBM Embed-Linear
Phase 3 with removing Number of Reviews	0.158	0.849	**0.074**	LGBM Embed-Linear
Phase 2	0.159	1.608	0.018	RF Pred
Phase 3	**0.152**	**0.235**	0.079	RF Pred
Phase 3 with removing Number of Reviews	0.156	0.838	**0.086**	RF Pred
Phase 2	0.154	1.633	0.002	SGD Assoc
Phase 3	**0.149**	**0.242**	0.052	SGD Assoc
Phase 3 with removing Number of Reviews	0.152	0.831	**0.094**	SGD Assoc
Phase 4	0.154	0.291	0.054	GANDALF Assoc
Phase 4 (with removing Number of Reviews)	0.157	0.300	0.026	GANDALF Assoc

Table 10: df-Analyze Final Results of the 3 Phases Predicting Price Feature

Phase Number	**MAE**	**MSE**	**R2**	**Method**
Phase 3	0.195	0.241	0.119	LGBM Embed-Linear
Phase 3 without Number of Reviews	**0.187**	**0.181**	**0.231**	LGBM Embed-Linear
Phase 3	**0.194**	0.246	0.100	RF Pred
Phase 3 without Number of Reviews	0.195	**0.206**	**0.126**	RF Pred
Phase 3	**0.230**	**0.274**	**-0.001**	SGD Assoc
Phase 3 without Number of Reviews	0.243	0.276	-0.156	SGD Assoc
Phase 4	0.264	0.215	-0.034	GANDALF Assoc
Phase 4 (without number of reviews)	0.266	0.214	-0.030	GANDALF Assoc

Table 11: df-Analyze Final Results of the 3 phases Predicting Minimum Nights

Across 3 different phases, we have compared the results of four different models; LGBM Embed-Linear, RF Pred, SGD Assoc, and GANDALF Assoc for two different target variables, Prices and Minimum Nights, each of them was predicted once without Number of Reviews feature and once with it.

GANDALF Assessment

While examining the results carefully, we can observe that GANDALF achieves comparable results. For example for predicting Price feature in Table 10 we can observe that GANDALF Assoc achieves MAE of 0.154 while the best results achieved was by SGD Assoc achieving MAE 0.149. The amount of ‘comparability’ can be subjective in this case, as for example, taking the same Mean Absolute Error (MAE) metric into account this time for Table 11, we find that the best model was LGBM Embed-Linear with value equal to 0.187, while Gandalf Assoc achieved 0.264, this is more than 70% increase in the error, and it was the highest error in general for the four different models we applied. According to the results above, GANDALF has the potential to compete with the top other Machine Learning Algorithms. However, it would essentially need further investigation backed with work on other datasets.

It is important to note that GANDALF results didn’t include feature selection, where –feat-select parameter in the bash script with GANDALF was selected to None. One reason justifying this would be that in order to produce proper comparison of the deep learner with other models, we have ran GANDALF for multiple times on different targets, taking considerable amount of time to complete. We will address this part more in the Redundancy Aware Feature Selection section.

Discussion

In general, we have been able to demonstrate the feasibility of applying different algorithms, including classical and deep learners (GANDALF) on our tabular dataset, yielding promising results, which we consider one of the biggest contributions of this work. The leading algorithm for predicting Minimum Nights was SGD Assoc we ran on Phase 3. On the other hand, the leading algorithm for predicting Price feature was RF Pred. We think that our study may help in business improvements directly related to room booking, placements, and costs in New York City, with the studies focusing on smaller scales usually lead to tangible improvement, as in [14], where the authors studied AirBNB in Corisca, and has been able to identify the areas with lower prices for rental.

Before beginning of the project, we have put three main points to address throughout our timeline, and we believe, we have been able to address them all in detail, achieving prosperous results

1. Predicting Prices: We have shown that SGD Assoc was able to achieve the lowest MAE and MSE losses in Phase 3, and the same model in the

same phase achieved the highest R2 score but with removing Number of Features.

2. Predicting Minimum Nights: We have shown that LGBM Embed-Linear was able to achieve the lowest MAE, MSE and highest R2 score in Phase 3.

3. Preprocessing and Apply Encoding Techniques: This was our key contribution, and it's something that we consider may set our project apart from the others in class as bonus. For Phase 2, we applied Label Encoding, One Hot Encoding, and removed redundant features. In Phase 3, we have implemented Bag of Words, handling unusual values, and number of reviews feature. We believe that our approach of preprocessing was effective and efficient empirically, as it leads to the best results as we shown in Results section, and hypothetically, as we tried to justify in Methods section.

Literature Comparison

It is difficult to compare our model's performances with other literature review mentioned in the introduction, as we aren't working on the same datasets or on a common benchmark ones where we can evaluate on the same baseline. We provide an extensive analogy with the results in [4] as we believe this is the nearest to our work. In [4] they used the same 3 performance measures we have used in our analysis (MAE, MSE, R2 Score) which are the same statistical validation functions employed in df-analyze. The authors best model's performance achieved 0.147 MSE error on the test set. We achieve comparable performance of 0.239 and 0.235 with our two best models, LGBM and Random Forests respectively. Moreover, they achieve 0.2132 as MAE. Our df-Analyze best MAE perofrmances are 0.150 and 0.149 with LGBM and SGD respectively. In [15], the author of the work using Random Forest Regressor achieved MAE of 0.252, which we achieved much better of 0.149. On the other hand, they achieved MSE of 0.166, which is lower than our best values (0.235). Our model achieving better MAE but not MSE, indicates that our model is robust to our outliers, another indication of the effectiveness of our preprocessing approach. It is noticeable that in most of the cases, df-analyze is capable of performing similarly with the literature work, or even better.

Redundancy Aware Feature Selection

We have applied the feature selection in Phase 3 results. We divide our discussion into two parts according to the target.

As we set the price as the target, the selected feature by the Wrapper-Based Feature Selection approach was *neighbourhood_labelencoded* with MAE equal to -0.20374. Regarding the Filter-Based Feature Selection Summary, the most important continuous feature was the *room* feature with MAE = -7.62e-03.

The most important categorical feature was *oheencoded_Entire_home/apt* with MAE = -2.552e-02. Based on our knowledge of the dataset, we consider all these selections properly valid, as rooms, apartments, and neighbourhood encoding are all highly correlated to the target price. For example, if some apartment has multiple rooms is located in an expensive neighbourhood, this will make it more costly. On the other hand, if it is an hotel studio in a cheap neighbourhood, it will cost less. These features are those who control and generate the price number.

As we set the minimum nights as the target, the selected feature by the Wrapper-Based Feature Selection approach was *number_of_reviews* and *availability_365* with MAE equal to -0.243 and -0.241 respectively. Regarding the Filter-Based Feature Selection Summary, the most important continuous feature was *number_of_reviews* by a big margin, with a MAE equal to 5.03e-03, the second most important continuous feature was *availability_365* equal to *8.94e-03*, with the most important categorical feature was *onehotencoded_Manhattan* with MAE equal to 7.86e-03. We believe also these results are valid according to our interpretation of the dataset, and they also consolidate the results previously discussed in the Results section, where the MAE for RF Pred, SGD Assoc, and GANDALF increases when we remove *number_of_reviews*, and it also enforces our decision that we consider this feature relevant.

Discussion of GANDALF Results

It is noticed that in our work, GANDALF achieves resutls close to the optimum but not the best, although people may argue that Deep Learners are those used extensively these days and should be expected to excel even in Tabular datasets problems, this wasn't the case for us. In many cases, it could achieve results better than the classic models, as achieving higher R2 score than any model that was running in Phase 2 for target price —as we believe our preprocessing approach made a significant difference, and achieving less MSE for target Minimum nights compared to Phase 3 running using

SGD Assoc. However, it didn't manage to get the best results on any of our benchmarks. One downside also we believe in GANDALF is the amount of time it takes in running. Although our dataset is relatively compact and doesn't contain hundreds of thousands of samples or even millions as it is usually the case with tabular data, it took considerable longer time to run. In the future, we hope the authors and developers of such method will be able to address this concern specifically.

Conclusion

We have analyzed AirBNB dataset for apartments and houses available in New York City. We managed to predict the number of minimum nights as a stay, the price, and we proposed an effective preprocessing technique which we consider the main out-of-the-box idea we got in this work. We managed to run 4 different models, including the emerging deep learner GANDALF, we discussed the best performed models, the drawbacks for each implementation and why we think that our study would provide a significant contribution to the community in such area of research.

References

1. Li, Chenxi. (2024). House price prediction using machine learning. Applied and Computational Engineering. 53. 225-237. 10.54254/2755-2721/53/20241426.

2. el Mouna, Lale & Hassan, Silkan & Haynf, Youssef & Nann, Mohamedade & Koumetio Tekouabou, Cédric Stéphane. (2023). A Comparative Study of Urban House Price Prediction using Machine Learning Algorithms. E3S Web of Conferences. 418. 10.1051/e3sconf/202341803001.

3. Maloku, Fatbardha. (2024). House Price Prediction Using Machine Learning and Artificial Intelligence. Journal of Artificial Intelligence & Cloud Computing. Volume 3(4): 1- 10. 1-10. 10.47363/JAICC/2024(3)357.

4. Rezazadeh Kalehbasti, P., Nikolenko, L., Rezaei, H. (2021). Airbnb Price Prediction Using Machine Learning and Sentiment Analysis. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2021. Lecture Notes in Computer Science(), vol 12844. Springer, Cham. https://doi.org/10.1007/978-3-030-84060-0_11

5. Luo, Tiejian & Chen, Su & Xu, Guandong & Zhou, Jia. (2013). Sentiment Analysis. 10.1007/978-1-4614-7202-5_4.

6. Mikolov, Tomas & Kai, Chen & Corrado,Greg and Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. https://arxiv.org/abs/1301.3781

7. Wang, Haoqian. (2023). Predicting Airbnb Listing Price with Different models. Highlights in Science, Engineering and Technology. 47. 79-86. 10.54097/hset.v47i.8169.

8. Hancock, J.T., Khoshgoftaar, T.M. Survey on categorical data for neural networks. J Big Data 7, 28 (2020). https://doi.org/10.1186/s40537-020-00305-w

9. Juluru K, Shih HH, Keshava Murthy KN, Elnajjar P. Bag-of-Words Technique in Natural Language Processing: A Primer for Radiologists. Radiographics. 2021 Sep-Oct;41(5):1420-1426. doi: 10.1148/rg.2021210025. Epub 2021 Aug 13. PMID: 34388050; PMCID: PMC8415041.

10. Mukaka MM. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. Malawi Med J. 2012 Sep;24(3):69-71. PMID: 23638278; PMCID: PMC3576830.

11. df-Analyze: https://github.com/stfxecutables/df-analyze

12. Ravid Shwartz-Ziv, Amitai Armon, Tabular data: Deep learning is not all you need, Information Fusion, Volume 81, 2022, Pages 84-90, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2021.11.011. (https://www.sciencedirect.com/science/article/pii/S1566253521002360)

13. Uçar, Muhammed & Nour, Majid & Sindi, Hatem & Polat, Kemal. (2020). The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets. Mathematical Problems in Engineering. 2020. 1-17. 10.1155/2020/2836236.

14. Brunstein, Daniel & Casamatta, Georges & Giannoni, Sauveur. (2023). Using Machine Learning to Estimate the Heterogeneous Impact of Airbnb on Housing Prices: Evidence from Corsica *. SSRN Electronic Journal. 10.2139/ssrn.4407202.

15. Kayrennisa. Airbnb Price Prediction with Machine Learning [GitHub repository]. GitHub. from https://github.com/kayrennisa/Airbnb-Price-Prediction-with-Machine-Learning