CRISP-DM analysis of Saudi Arabia Real Estate dataset scraped from aqar.fm

Polina Zelenskaya
Innopolis University
Innopolis, Russia
p.zelenskaya@innopolis.university

***Abstract*—Aqar is a popular platform to rent and sell real estate in Saudi Arabia. In this paper, we will apply CRISP-DM analysis to their real estate dataset to identify possible beneficial use cases of data.**

***Index Terms*—CRISP-DM, Real Estate, Analytic, Machine Learning.**

I. Introduction

CCRoss-Industry Standard Process for Data Mining (CRISP-DM) is a standardized method to analyze, describe, and research data for industrial purposes. This technique aims to make discovered results reliable and repeatable, as well as comprehensible by people with little-to-no Data Mining (DM) experience. This approach consists of 6 major steps and, if done correctly, leads to an in-depth understanding of data and use cases it could either benefit or harm.

Saudi Arabia Real Estate dataset [1] contains information about real estate in Saudi Arabia scraped from Aqar [2] online platform. Data consists of text descriptions, photos, locations, and numeric values representing prices, phone numbers etc. In total, it contains 45 different features with 663946 samples.

II. Business Understanding

A. Business Objectives

To start the analysis, it is important to understand the business situation itself (Due to language barrier and translation difficulties, this part should be considered carefully, as all information is taken from one official website [2] and post-proceeded by Google's Arabic-English translator). Aqar is the largest real estate platform in Saudi Arabia. It attracts 50 million viewers and contains information on about 1.5 million properties. The company mostly operates in the digital space and makes money from selling leads to real estate agents, selling add spaces and recommendation priority, as well as selling subscription models for customers. According to [3] Company's net worth is estimated around 7.9 million dollars as of 2019.

Considering everything have said, we can now form a business objective. From a business perspective, clients primarily want an easy-to-use platform to find commodities with reasonable prices, truthful descriptions, and safety guarantees. However, Aqar also provides an interface for real estate agents, so from their point of view, their advertisement must be shown to a target audience and result in a high engagement rate. Additional objectives are platform stability, a high priority in search engines, client market expansion, and client return rate.

It is also important to state business success criteria. As we are not related to the company directly and have no insights about their company culture and goals, we can only speculate. Success could be measured as the number of new users signing in a month, the number of active users on the platform, the total value of transactions between real estate agents and clients, retention rate, or just revenue generated.

B. Situation Assessment

To conduct this analysis we need to consider available resources. First of all, we have Saudi Arabia Real Estate dataset [1] with over 300 thousand objects each having 45 features, the official Aqar website [2], and an Arabic-English Google translator to break the language barrier. Speaking about experts, we are free to discuss this work with our Teacher Assistant Alaa Aldin Hajjar, and Professor Armen Beklaryan. In a discussion of computational resources, we have Nvidia 2080 SUPER mobile GPU and Intel core i9 processor, this should be enough to train light models and scrape information in a reasonable amount of time. Considering our team's knowledge, we will use Python 3 programming language and the frameworks it offers (Scipy, Pandas, Numpy, Matplotlib, Pytorch, Requests, BeautifulSoup4).

It is also important to discuss requirements, assumptions, and constraints. There are several report requirements: the work deadline is the 13th of May 2024 (work started on the 27th of January 2024), and the analysis as a result should provide a beneficial solution to the company, with highlighted data patterns and outcomes. Speaking about data assumptions, we suppose that there are patterns in the dataset and those patterns correlate with business success. In addition, we can suppose that data is fair and unbiased, as well as has no crucial mistakes (such as invalid locations, unreasonable prices e.t.c.). And discussing constraints, we should note that the language barrier is the main one, as no one in our team knows Arabic. In addition, we have no connection with the Aqar company, so all company details are from public resources found on the internet. Dataset quality is data constrain, as we use already scraped data and we do not have resources to collect additional information, this is especially important as the dataset covers only dates from 01.01.2023 to 04.08.2023 (so it is already outdated for half a year).

Risks are not business related, as this is an educational project, and only our success on the course is influenced by it. However, if it is deployed, ethical considerations should be made, and the model should be tested on different kinds of bias it could face. In addition, if the model won't be provided new data it may get outdated and remove all benefits it was intended to add. For safety reasons, an explainable and simple approach should be developed, that will be enabled instead of the model in case of emergency.

Now we can discuss the cost and benefits. From this dataset we see potential from the price prediction model, it could provide new functionality to the platform, be used in a subscription for more in-depth analysis of a price customers see, and show real estate agents what price is reasonable for customers to buy or rent property. Also if the model will support an opportunity for an importance feature matrix, we can show clients and agents how features influence the price. Now we can dive into details. According to [3], the company has an average of 175 thousand unique visitors each day and generates about $4300\$$ daily from ads. If we want to embed a price prediction model, we will need to add a dedicated server with a modern GPU to support such amount of clients. We can suppose that most of the server requests come in the daytime for Saudi Arabia (their target clients are people from Saudi Arabia) and more computational power is required during the day. For that reason, to suppose an average number of requests we will divide the total number of unique users by a number of working hours (8) and result in approximately $6 \approx \frac{175673}{60 \cdot 60 \cdot 8}$ new unique users each second. As we need to estimate the upper bound, we can suppose that each user will spend on average 2 hours on the website, resulting in $44000 \approx \frac{175673}{60 \cdot 60 \cdot 8} \cdot (2 \cdot 60 \cdot 60)$ users at the same time online, each sending requests to our new dedicated server. For such a load we will require a high-performance computing server, which we can rent with the Tesla T4 cluster from Yandex Cloud [4] for around $950\$$ a month. With this in mind, we now need to determine how the model could benefit the company. If it will be introduced as a subscription to provide enhanced information about price and feature correlations, to fulfill the expenses we can determine the number of subscribed clients required if the monthly subscription price will be p - $N = \frac{950}{p}$. We can see that even with $1\$$ payment we only need 950 subscribed clients, which is a reasonable amount considering the daily count of visitors is around $175k$. With a caching system, we can cut down costs a lot more, however, we will only consider the upper bound in this case. Speaking about benefits from this model, we can't say for sure a number of recurring users, or a number of new users visiting the platform daily, however, we have information that daily platform faces $175k$ visitors, and according to [5] industry average of all users willing to buy subscriptions is 0.62%. Knowing all that we can make a rough estimate that approximately $1085 \approx 175673 \cdot 0.0062$ users would get a subscription. And with price as low as 1 per month, this model would support itself in making small revenue in the long run, and with $p = 5\$$ it could boost daily income from estimated [3] $4321\$$ to $4470\$ \approx 4321\$ - \frac{950\$}{30} + \frac{1085 \cdot 5\$}{30}$.

In addition to production costs, we also need to consider the cost to develop this project. From a data analysis point of view, this project could be done by one Data Science Junior developer. The time constraint is difficult to estimate as it could vary from individual experience, however considering our project plan, we suppose that this work could be done and fine-tuned in 3 months. With this in mind, we can estimate developing expenses. According to [6], the average Junior developer salary in Saudi Arabia is about $50000\$$ per year, meaning 3 months of development will cost approximately $12500\$ \approx 3 \cdot \frac{50000\$}{12}$. With $p = 5\$$, company will profit from project in 84 days $84 \geq 3 \cdot \frac{50000\$}{12} \cdot \frac{1}{4470\$ - 4321\$}$, and with $p = 1\$$ in around 2778 days $2778 \geq 3 \cdot \frac{50000\$}{12} \cdot \frac{1}{4325.5\$ - 4321\$}$.

C. Data Mining Goals

Now when we have estimated cost benefits, we can describe our data mining goals. As we already discussed in *Situation assessment*, the main goal is to develop a model that will predict price and show how each real estate features correlate to it. The most important factor is the model's interpretability, so deep learning techniques should be considered carefully.

Considering the model's evaluation, in this problem MSE metric should be preferred to MAE as one mistake in $1000\$$ is more crucial than 10 mistakes by $100\$$. The model always predicting in the correct hundreds would be considered good, but even more precise predictions would benefit even more. Speaking about subjective evaluation, the model should be unbiased and provide results that would correlate with human intuition.

D. Project Plan

To develop this project we need to consider each CRISP-DM step. In total, there are 6 steps (Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluating, and Deploying), where on each step we have 2 weeks to develop and report. As 01.02.2024 is the deadline for the first step, we can calculate each new deadline accordingly. For the last two weeks deadline will be report finalization and outcomes of the work.

Tools we will be using are Python3 with a machine learning stack (Pytorch, Matplotlib, Gradio, Sklearn, Pandas, Scipy, Numpy). If required, we will consider some additional libraries and tools.

III. Data Understanding

A. Collect Initial Data

First of all, let us discuss the collection step. As we decided to use [1], which is Saudi Arabia Real Estate dataset that contains information about buy and rent price of properties throughout the whole county. This data is already scrapped and all information in the dataset is taken from [2]. For now, we consider this dataset sufficient to make prediction model, as all features are taken directly from Aqar. This means that new data appearing after deployment would be treated identically with no additional scraping required, which simplifies both deployment and maintainment of the model.

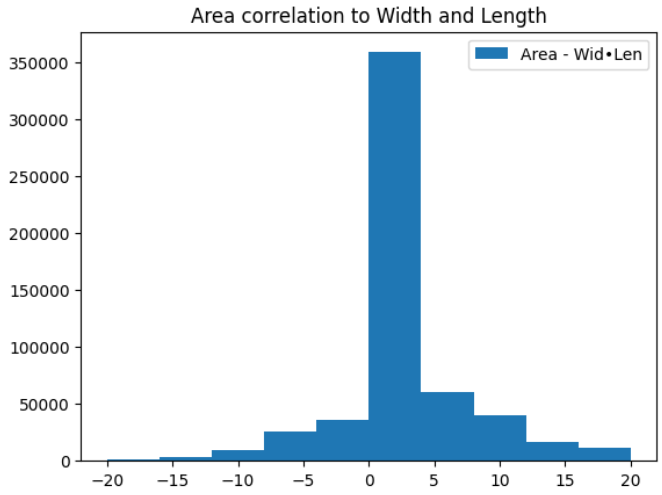

Fig. 1: Deviation of ‘Area‘ and it’s estimation via ‘Length‘ and ‘Width‘ on -20 to 20 range with 10 bins.

All statistics from this section is available publicly on our GitHub repository [1] in ’Data exploration’ notebook.

B. Data Description

Considering this data superficially, we can say that [1] contains of 45 different features with 663946 samples. Features contains information about house location, price, inside structure (number of rooms and their types), create time, user reviews, owner information, estate category (rental or sell, apartment or villa or room) e.t.c. In addition, it contains images, which could also be used. All this information is taken from [2], which is convenient as it would be simple to increase dataset if required. Worth mentioning data modernity. This dataset consist information for time-period from 01.01.2023 to 04.08.2023, and as for year 2024 it is already outdated by half a year.

C. Data Exploration

First of all, let us discuss target - price. It has mean of 3986665 and std of 62561260. Most interesting part is that prices mostly located at low values, and then gradually decrease as cost rises. In addition to that, there is spike for less than 5000 values, probably indicating rent costs.

Speaking about some feature correlations, we decided to investigate how ‘length‘ and ‘width‘ correlates with ‘area‘. For that reason, we applied basic idea that ‘area‘ is approximately ‘length‘ times ‘width. To proof this, we subtracted from ‘length‘ product of ‘length‘ and ‘width‘, and surprisingly enough, this estimate shows strong correlation between these three columns. With error window ranging from -100 to 100 we cover 97% of data, with range from -15 to 15 we cover 90%, and with range as small as 0 over 50% of samples. Whole data could be seen in Fig. 1. For this strong correlation, we will consider deleting these columns to remove multicollinearity.

[1]https://github.com/cutefluffyfox/saudi-arabia-real-estate

Discussing real estate owners, mean rating is 4.41. Interestingly, 25 percentile is equals to 4.19, meaning most of owners have rating of at least 4. Around 99% of users verified, and about half have ‘rega_id‘ filled. Around 34.9% are classified as ‘normal_marketer‘, 30.9% ‘exclusive_marketer‘, 19.6% ‘agents‘ and 10.8% ‘owners‘.

Most of the real estate is located in the capital of Saudi Arabia Riyadh (53.2%), followed by Jeddah (15.3%), Dammam (5.5%) and Al Khobar (4.6%). Popular distincts are An Narjis (5.1%) and Al Aarid (4.1%). Around half (48.4%) of the real estate was posted in the January of 2023. Average house contains 3 wc, 1 living room and 4 beds. About 58.5% of real estate is not furnished, and around 53.9% having kitchen. Most of the estate promoted being new with 38.5% was built less than a year ago and mean age value of 3.78 years.

D. Data Quality Verification

First of all, before going in-depth, we can look into how pure our data are. By looking at number of missed values, we can see that all data named ‘native.*‘ (5 columns) are almost fully absent (with only 2 samples containing any information about them). For that reason, we consider that these columns have no important information, so we can remove them entirely.

In addition, column ‘has_extended_details‘ has only 10422 (1.5%) non-Nan values. However, this feature is binary and all filled values equals to 1, meaning we can fill this column by 0 to restore all data.

Worth mentioning, column ‘rent_period‘ also seems to contain a lot Nan’s, to be precise 571232 (86%) samples. Firstly this information seems to be logical as sales advertisements won’t have information about rent period, however if we will consider ‘rent_period‘ column for any property for sale (e.x. ‘class‘ 2), we see that some of them will have this information filled. Meaning not only that rent period do not indicate rental only adverts, but also that some further filtering of the data is a must. Furthermore, there exists ‘daily_rentable‘ column with 315493 (47.5%) missing values, which is also present in properties for sale and have all kinds of ‘rent_period‘ values.

Speaking about interior features, almost third of all values missed in ‘furnished‘, ‘livings‘, ‘ketchen‘, ‘wc‘ and ‘beds‘, and 555898 (83%) values missed in ‘ac‘ column. It is hard to say for sure whether this information is not filled on purpose or by mistake, so we will not impute these columns.

Discussing exterior features situation is better. The most unfilled column is ‘age‘ with about third of values being Nan, and for the ‘street_width‘, ‘street_direction‘, ‘width‘ and ‘length‘ on average 8% values missed.

Considering home holder themselves, 118066 (17.7%) owners do not have reviews, 222565 (33.5%) do not have profile image, and 338325 (50.9%) did not provide importation about their Real Estate General Authority id.

And lastly, features ‘type‘ and ‘advertiser_type‘ have 502773 (75%) and 23932 (3%) respectively. The main problem with these features is that they are hard to interpret, as no description in dataset itself [1] or Aqar platform [2] could be

found. For that reason, we consider to remove such features in the future.

IV. Data Preparation

A. Data Selection

To understand what data we is relevant to us, first we need to address data mining goals, as well as some data constrains we face. We want to implement interpretable model that will show clients and owners how price is formed. For clients this will be beneficial as they will know what parameters are the most influential, when for owners they will know what parameters to tweak in order to get higher price (e.x. furnish, change sell to rent e.t.c.). Looking from this perspective, adding another dataset seems to be not viable as we won't have immediate access to additional features when new real estate will appear.

And in discussion about limitations, *Data Quality Verification* showed that even thought data is problematic, basic cleaning techniques could make data more robust. In addition, there are already several columns which require no data cleaning. Nevertheless, not all data could be imputed, 'native.*' attributes contains only 2 samples which is too few to even normalize.

In addition, some data type is hard to use in interpretable way, for example we do not have access to website images, and even if we had, it would be difficult to design such model which could highlight importance. There are some ways to make is possible, such as GradCam [7] or LIME [8] , however they are still difficult to interpret in current task. Why should some image pixels correlate with real estate price? Same question could be asked for textual data and will be considered out of scope for current iteration.

B. Data Cleaning

As a first step we decided to remove columns that: had too few values ('native.logo', 'native.title', 'native.image', 'native.description', 'native.external_url'), was almost fully unique and could lead to bias ('user.phone', 'user.rega_id', 'user_id', 'id'), had image or text as data type ('user.img', 'uri', 'user.name', 'city', 'district', 'imgs', 'path', 'content', 'title'), represented website time data which do not relate to real estate ('last_update', 'create_time', 'createdAt', 'updatedAt', 'refresh'), strongly correlated with any other column ('width' and 'length'), and could not be explained ('type'). This step still left 22 features with different number of null-values. Most of the columns were already discussed in *Data Selection* and *Data Quality Verification* sections, however 'content' and 'title' should be noted. These columns could provide more insights into property and advertiser intend, however as our aim is to make model more interpretable and considers correlation with price, it raises several questions. First of all it is price manipulation with prompts that do not resemble reality, and second reason is interpretability itself. For such reasons we disregard both columns.

As a second step we highlighted non-imputable column that have nan's. From one side all columns seems important, however the main criteria for this group were: whether it makes sense to impute values and will they impact data distributions. For example, 'age' is is hard to impute as 38% are unknown and any modifications will effect how model analyze this column. For the same reasons we removed all rows where at least one value from 'beds', 'livings', 'wc', 'ac', 'ketchen', 'furnished', 'advertiser_type' was equal to nan. This process left us with 91501 rows which is 13% of all original data. This is not a problem as we are developing model to get feature importance, not make precise prediction, where quality is more important over quantity.

Lastly, in discussion of outliers, only two columns have extreme values - 'area' and 'price'. To detect them, we used 'IsolationForest' with supposed contamination 0.25% and random state 42. This highlighted 431 rows of extreme values such as houses with area 3000000 or apartments costing 3.2 million USD.

C. Data Construction

In construction step we will discuss only data imputing, as we are interested in reachable data during inference. This step was simple as not so many columns contained unknown values.

For column 'has_extended_details' we filled all values with 0. We already analysed that this data is binary and contain only 1 and nan values, meaning nan is substitution for 0. For more information check *Data Quality Verification* section.

For more complex columns such as 'area', 'street_width', 'age', 'street_direction', 'user.review' we used median imputation. There are several reasons behind median, and not mean or mode. First of all it is to minimize influence of outliers, even though most of extreme one was removed, still there may be some present. In addition, some columns such as 'age' are integer and mean is float, so median is more natural in this case. And finally, all of them are numeric (not categorical) values, so mode seems out of place.

D. Data Integration

To extend our data we went for a two approaches. First is human-made features, and second is polynomial features. As for human made, those are the features that make sense for people - number of rooms, different real estate types, type of advertisement (rent, sell), all inclusive (at least 1 bed, living room, kitchen, wc, ac). These features was constructed by looking at different columns and just checking some conditions.

For the second type, features makes a bit less sence as they are 2-nd degree polynomials. What do we mean by that is that we take each feature pair and multiply their values, this makes some bizarre 'beds-street_width' and other combinations, however they may result useful in future steps.

E. Data Formatting

The last step is to make data viable for models to process. In order for that we applied one-hot-encoding on categorical features 'rent_period', 'advertiser_type', 'daily_rentable', 'user.iam_verified'. Worth mentioning is that when nan value

occur, we just consider it as 'all-false' case to save meaning of null-value, yet not adding another column for it.

And for scaling we decided not to apply any type. This decision was made based on different meanings it could bring. For example, min-max scaler and z-transformation scalers will provide completely different results and may even not be applicable by model such as CatBoost regressor which needs to know un-scaled values.

V. Modeling

A. Select modeling technique

Before starting modeling it is crucial to understand the types of models we can use and the limitations behind them. Our main focus is model interpretability, and due to that deep learning pipeline seems less applicable. The main problem with DL is the lack of transparency, as a multi-layer neural network would have complex dependencies that are hard for people to understand and explain. Even though there exist some ideas on how to describe black box architectures (such as LIME), they are not exhaustive enough to be applicable. For that reason, we will stick to a classic machine-learning approach.

In machine learning, there exist different ideas and architectures. As we are working with table-format data, it is a good idea to use boosting or stacking architectures as they usually perform better than just single-model architectures. For now, we will stop on the Catboost model family as they are easy to use and have high-level API with all the required functionality. In addition, catboost is based on decision trees, which could explicitly show how the "decision" process works.

B. Generate test design

To test model quality we should first look to a target. As the main parameter is real estate price, we can easily understand that we are working with a regression task. However, our data has a wide range of values from low rent prices to luxurious houses on sale. For this reason, not all metrics are applicable. For example, basic MSE (Mean squared error), MAE (Mean absolute error) are based on the idea of raw error, however errors in higher prices could yield much higher errors than in low prices, for that reason, some metrics that measure percentage error or variability should be applied. Such metrics could be MAPE (Mean absolute percentage error) and R2 (Coefficient of determination). Nevertheless, we will use all the described metrics above, but mainly we will focus on MAPE and R2 (MSE, RMSE, MAE would just provide more insight into how the model performs).

It is also a good practice to split the dataset into train and test sets, in our case we used the 'train_test_split' function from 'sklearn.model_selection' package with 20% being dedicated to test, 80% to train, and seed used is 42.

C. Build model

The first idea was to just use out-of-the-box CatBoostRegressor (with RMSE loss function) on the whole training dataset to obtain baseline metrics. The surprising part is that even such a model performed pretty well with MAPE ≈ 2.479 and R2 ≈ 0.87. All metrics are shown in Table 1 column *base*.

The second thought was to select some columns based on feature importance, and then train the model on them. We used 0.1, 0.2 and 0.5 splits. The best split was 0.2, you can check all metrics in Table 1 columns *Im 0.2*.

The third option is to change the loss function. Instead of RMSE, we decided to use MAE and MAPE. MAE performed better than MAPE overall. You can check the results in Table 1 in column *MAE*

The fourth way was to apply a grid search. Param field we used are iterations [200, 300, 500], learning rate [3e-4, 0.01, 0.03, 0.1], depth [2, 4, 6, 8] and l2 leaf reg [0.2, 0.5, 1, 3]. In addition, we combined them with different loss functions. The best one was RMSE with depth 6, iterations 500, learning rate 0.1, l2 leaf reg 0.5, and resulted in around the same performance as base. For more details check table 1 column *gRMSE*

TABLE I: Methods 1-4

	Methods			
Metric	Base	Im 0.2	MAE	gRMSE
MAE*	1.268	1.259	**1.218**	1.278
RMSE*	4.340	4.262	4.834	4.426
R2	0.875	0.880	0.846	0.871
MAPE	2.479	2.467	**2.355**	2.561

* scaled down by 10000.

The fifth hypothesis was to split the dataset by properties for sale and rent, as we thought it could help decrease variability. For both dataset types, we applied grid search with RMSE. Best parameters found are depth 6, iterations 500, learning rate 0.1, l2 leaf reg 0.2 for rent only, and depth 4, l2 leaf reg 1, iterations 500, learning rate 0.1 for sale only. All metrics can be found in Table 2 in columns *Rent* and *Sale* accordingly.

The last idea was to use catboost automatic feature selection algorithm with a different number of features to save. The optimal number of features to save is 26, as 25 and less result in a sudden performance decrease, but until 26 RMSE loss function slowly decreases. This idea is superior almost by all metrics considered, only second only to raw *sale* (or MAE if we do not consider *sale* due to partition). More information can be found in Table 2 under column *Auto26*

TABLE II: Methods 1, 5-6

	Methods			
Metric	Base	Sale	Rent	Auto26
MAE*	1.268	26.269	1.071	1.282
RMSE*	4.340	36.338	2.720	**4.146**
R2	0.875	0.649	0.662	**0.886**
MAPE	2.479	0.293	2.623	2.394

* scaled down by 10000.

D. Assess model

According to Tables 1 and 2, we can see that *Auto26* approach has proven to be superior to others. A high R2 coefficient shows that predicted data preserves the same distributions and statistics as real values (at least on the test dataset). In addition, MAPE of 2.394 expresses that we still face not a small percentage error in our predictions (due to value not being scaled from 0 to 100, 2.394 means 239.4% error deviation from expected results, this is probably happening due to extremely low values in rent prices as raw sales do not have such problem, more time should be spent identifying this problem). Even though MAE of 1.282 (scaled) and RMSE of 4.146 (scaled) are measurable to other models, they are not so useful due to the extreme price differences in the dataset. We still consider 12800 MAE an acceptable score.

In our opinion, this model shows potential in achieving our data mining goal and should be deployed. The main step is to understand how we can use it to show feature-to-price correlation, and for that, we can use the idea of LIME to tweak parameters individually and infer the model. This approach could be optimized if only columns from the list of 26 will be modified. In addition to stated before, we could show generated graph splits for clients to provide them with more insights into the decision-making process.

VI. Evaluation

A. Results evaluation

Let us quickly recap what was the business objective and what have been done. As we discussed in Business Understanding, clients want easy-to-use platform where they can actually find commodities with truthful descriptions, safety guarantees, and reasonable or explainable prices. Agents want to show ads to target audience and receive high engagement rate. To help achieving both needs, we decided to develop a model that will predict prices and show how each real estate features correlate to it, the most important factor is the model's interpretability. Basically we want good explainable model that can follow same distributions as in training data.

In the Modeling part we trained CatBoost regression model, which is based on the decision trees. The nice part about decision trees is their interpretability as we can visualize them with their thresholds. But as we are working with boosting, this may result in a bit more complex system, but in general it is just a list of trees (where each one could be explained) stacked on one another. So for some given sample we can see how each individual tree modifies the result. Considering the interoperability, we would say that it is met.

Speaking about metrics, in Table 2 we can notice that MAE and RMSE are pretty high. In our data this error means that average error is around 128000 Saudi royal (which is high). However, judging by R2 score we can see that most of the distributions are preserved. Basically, model is making big error in the exact price prediction, but is follows similar patters to price change as real data. So partly our goal was achieved, but surely not to a full extend.

Considering model limitations, the main one is exact price prediction. This model should not be applied in scenarios where price values influence a lot, however it could be used to show how price differs in percentage, rations. The other limitation is that all selected 26 features should be filled to even apply this model in the first place.

B. Process review

Let us look into more details how we made this model. First part after Business and Data Understanding was Data preparation. Simply describing, we noticed that data contains a lot of null values, so we made a list of columns that are considered 'non-imputable' and filtered rows by that criteria. In addition, we entirely removed columns that are fully unique, contain too few values, have non-numerical data type (text, images), and most of the time data. Then we imputed data by median, and lastly removed outlier with use of IsolationForest technique in area and price columns.

Looking just at data prepossessing, we can raise several question. First if all, are the selected 'non-imputable' columns actually result in the "best performance"? Furthermore, text and image data may had an impact on model performance, is it positive or negative? And is IsolationForest is the best choice to remove outliers?

In modeling part, we heavily relied on CatBoost regression model. We selected features with it, applied grid search algorithm to it, and then suggested that generated graphs are representative enough for be considered explainable. In addition, we rely on several constant such as number of features to select, loss function and train test split.

Here we can also ask ourselves whether CatBoost is superior approach for this dataset? Can described constants be tuned to achieve better performance?

Simply, there may be room for improvement and we can tweak a lot of parameters both in data preparation and modeling parts, but to justify new or already made decisions we require additional resources including time. Overall, the techniques used are industry standards, but this doesn't not mean they are the best, just a good practice.

C. Next steps

Looking at evaluation and review, we can start to look forward to the next scenarios. Currently there are two possible ways what could happen to model. We will look into both of them.

The first scenario, we can start deploy model now. We need to understand that current model could only be used to show how features relate to the price, without showing exact price changes. This was a result of high MAE and RMSE scored seen in Tables 1 and 2. The price difference in too high (MAPE over 100%) even to try making prediction about exact price changes. However, as it was discussed, R2 score of 88% shows that we can use model to predict percentage difference, as distributions are mostly preserved. From business side, this approach could start making a small profit even if we will make a 1\$ subscription to show this data. We already discussed

in situation assessment how much exactly this model could make, and with $5\$$ company will start making profit in 84 days (daily income boost from $4321\$$ to 4470), and for $1\$$ model would sustain itself and profit in 2778 days (check situation assessment for the math behind).

The second scenario is to continue developing model. As model is not capable to predict price even in a hundreds, more research and analysis should be made to understand whether it could be done in the first place, and if so, how. Basically, more time should be dedicated to analyze how to make model more precise with price prediction, but this task may be impossible in the first place. In addition, more experiments should be made about data filtering and data modeling choices (other models trained, different preparation techniques, constants tweaked, etc.). In this case model would profit in a longer span (considering we will use same subscription prices) as more resources would be spend in the first place (money to pay for developers work, computational servers used, and time in general).

From our point of view, we see the first option to deploy model much more appropriate in this case. Model could be deployed with low subscription price to sustain itself (2778 is unreasonable amount of time to make a profit), while tests in the background could be made to try and optimize the performance and make sure model do not have biases towards some properties, cities, locations etc. .

VII. Conclusion

The research conducted shows that developed CatBoost model almost fully achieve our data mining goals. With some limitations, this model could be deployed and tested with real samples and requests from users. However, more research should be done to analyze whether better performance is achievable, either by changing architecture itself, or tweaking some parameters that are already defined. With $1\$$ subscription cost this model could sustain itself, and with $5\$$ price it could make a profit in around 84 days.

References

[1] Mohd PH. 2023. Saudi Arabia Real Estate dataset. https://www.kaggle.com/datasets/mohdph/saudi-arabia-real-estate-dataset/data .

[2] Aqar - Platform for real-estate marketing. 2014. https://sa.aqar.fm/ .

[3] SiteIndices - Ranking platform that collects information about business websites and estimates businesses value and income. 2019. https://aqar.fm.siteindices.com/ .

[4] Yandex Cloud - Cloud platform providing scalable infrastructure and storage, for digital services and applications. 2024. https://cloud.yandex.com/en-ru/prices .

[5] Service Form - advertisement company that helps growing business through internet. 2023. https://www.serviceform.com/blog/ultimate-guide-to-real-estate-website-conversion-marketing .

[6] SalaryExpert - platform for publication of Economic Research Institute studies. 1989. https://www.salaryexpert.com/salary/job/data-scientist/saudi-arabia .

[7] Ramprasaath R. et. al. 2016. CAM: Visual Explanations from Deep Networks via Gradient-based Localization. https://doi.org/10.48550/arXiv.1610.02391 .

[8] Marco Tulio Ribeiro et. al. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. https://doi.org/10.48550/arXiv.1602.04938 .