CS5228 Kaggle Project Final Report

He Yingxu	Huang Xuhui	Liang Bowen	Xu Xuanyue
SoC	*SoC*	*SoC*	*SoC*
NUS	*NUS*	*NUS*	*NUS*
A0162656Y	A0225539Y	A0224920M	A0229566R

I. Introduction

A. Motivation

The resale market of HDB flats is a big business in Singapore. And the resale price is always what a buyer or a seller most cares about. There are many factors affecting the resale price, like the size, type of flat, etc., and how to identify and quantify them to evaluate the price is rather important. Therefore, we explore train datasets and some auxiliary data to find out key attributes and quantify them, then try to refine a regression model through tuning. Our goal is to predict the resale price exactly based on known information, which can enable buyers or sellers to evaluate the resale market of HDB better.

B. methodology

To achieve our goals, we do careful exploratory data analysis in the pre-processing step, including identify potential outliers, identify and clean dirty data if exists. This allows us to understand more about the available data we have. Next, we do data collection to improve the data quality of auxiliary data. In the data processing phase, we explore the relationship between all the features and the dependent target to obtain the selected training features. Try different encoding methods to convert categorical features into numerical features. Lastly, we explore different models and do experiments to predict resale price, and try to refine the selected model by tuning and cross-validation.

II. Exploratory Data Analysis and Pre-processing

In this part, we first clean the training dataset and try to identify possible outliers. Then we focus on categories attributes, converting them into different numeric types, which can be handled in a regression model. After that, we are trying to create some useful new features to improve the accuracy of our model.

A. For data

1) Data quality: In the main dataset[1], no missing data were found in both train and test datasets. However, there is some missing and wrong information in auxiliary data. For example, in population data, we find there is a 'central region' in the training dataset but not in population data. According to 2015's population data in *data.gov.sg*, we fill this missing data, then merge subzones' population into train and test datasets. At the same time, the train stations dataset contains missing data as well. Another important missing information is that we should consider the opening year of MRT and LRT stations. For example, the Botanic Gardens statin has no 'opening year' and the type is 'other', we manually update this row data. Next, we manually collect the 'opening year' for the LRT stations.

2) Identify outliers:

- **Normalize the resale price**: Since the resale price has skew distribution, which does not meet the assumption of the regression. To fix it, we apply boxcox transformation and normalization for the price to transform it into norm distribution.

$$f(x) = \begin{cases} \frac{(y+\lambda_2)^2-1}{\lambda_1}, & \text{if } \lambda_1 \neq 0 \\ \log_{10} y + \lambda_2, & \text{if } \lambda_1 = 0 \end{cases} \quad (1)$$

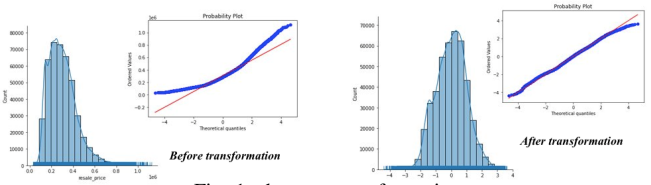

Fig. 1. boxcox transformation

- **Manual detection by visualization**: According to the strong positive correlation and distribution between area and price (showing in Fig.4.), in the same town, we can draw a scatter map for area and price to find outliers. The Fig.2. showing the outliers in town Bukit Merah and Clementi (annotated as the red point). However, it is found whether deleting outliers or not is not essential for our prediction (see table 5). We believe that it is because we have a large enough dataset, and outliers are too few compared with the training dataset. (will discuss more details in the evaluation part)

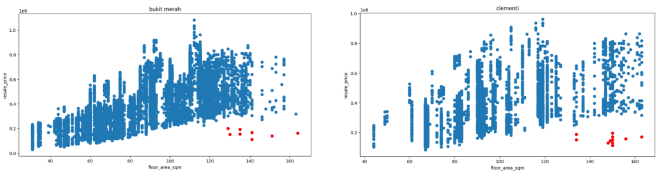

Fig. 2. outliers

[1]https://www.kaggle.com/c/cs5228-2020-semester-2-final-project/data

B. For attributes

1) Filter features: To choose a useful feature, first, we choose some intuitively import features, and discuss the price distribution with these features and evaluate their importance. For example, Fig.3. shows the price distribution with different locations and infrastructures, which indicates that price distribution is highly correlated with the location and infrastructures of HDB, as we expected before.

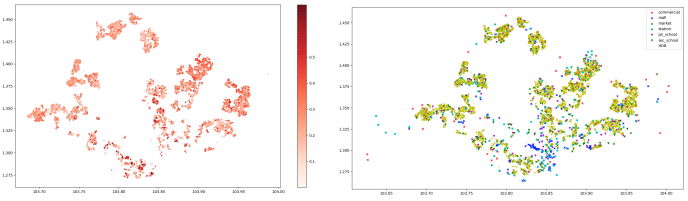

Fig. 3. price distribution with location and infrastructures

Besides, 'eco category' and 'elevation' have only one unique value, intuitively there is no use for regression. Therefore, we remove them during training and prediction. We also filter features with duplicate information. For example, 'Street name' and 'block' is overlapping with location information such as longitude and latitude or subzone, planning area, town, and region, we choose to use the latter one and drop the former one. Finally, we also draw a correlation map and importance map for all features and choose important ones for our final model, which is discussed in the last section of this part.

2) Explore correlation between features: According to the correlation figure above, the original numeric features excluding price have a low linear correlation (Pearson Correlation Coefficient) with each other. Area, story, sale year, lease commence date of HDB have a strong linear correlation with price. Especially, the area and price have a strong positive correlation. We also apply a T-test and observe the P-value of each feature find out whether this feature is significant.

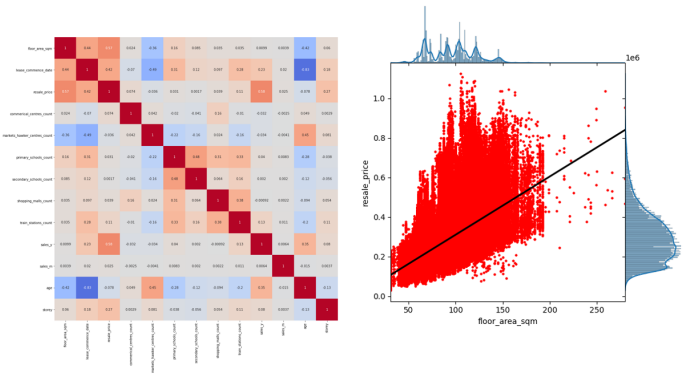

Fig. 4. correlation heatmap(left) and distribution between price and area(right)

3) Convert category attributes into numeric ones:

- **Convert it by average value**: In the 'storey' attribute, there are ranges to cover the HDB floors, and the classification ranges are not reasonable. Therefore, we replace it with the average floor value of the ranges.
- **One-hot dummies**: We apply one-hot encoding for category attributes and drop original columns. However, due to so many categories, the dimension of this dataset is extended largely (around 300). Therefore, we also do experiments using label encoding. And we find out under our regression model, the one-hot encoding is better than label encoding. Besides, to tackle the large dimension and the sparse problem with one-hot encoding, we create others features as support features for encoding (discussed in the next section). And it seems the 300 dimensions feature has little harm for our model because our model tends to use the most significant feature for regression (discussed in the evaluation part).
- **Computing WOE values**: Since WOE generally is used for classification, we adjust the calculation methods so that it can apply in this regression problem [6]. Firstly, group the dataset by a certain category attribute, then aggregate each count and sum of values. Then get each sum and count's proportion of the whole category feature to get WOE for each category by the formula below. In this way, all categories can be quantified by numerical values without increasing the data dimension.

$$\begin{aligned} sum_i &= \sum price_i / \sum price \\ count_i &= count(*_i)/count(*) \\ WOE_i &= \log(sum_i/count_i) \end{aligned} \tag{2}$$

4) Generate new features:

- **Combine some features as one variable** : Through research, we notice that some categorical features such as flat type and flat model, or town, region, planning area and subzone are related. And to denote the type information and location information of HDB. It is better to combine these features as a new feature and do one-hot encoding for them.
- **Create support features for one-hot encoding** : In the correlation analysis we find the sealing date and location of HDB are highly related to the resale price. To help one-hot decoding better reflect the intrinsic and internal information, we create some new support features. For sale date, we look up to the HDB price index [4] which is released by the government to denote the HDB price index in different years. The price index curve is showing below. Using this price index as a support feature, our model can learn the price level in different years more directly.
  Out of the same purpose, we calculate the mean price of HDB with the same flat type and flat model to help the model understand the price difference of different models and types. We also calculate the mean price of HDB in the same town, region, planning area, and subzone to help model differentiate the price in various locations. In addition, we do some experiments of creating new features using the price of its K nearest neighbor. However, the results are not ideal and it harms the final prediction, so it is discarded in our final model. The reason why KNN has

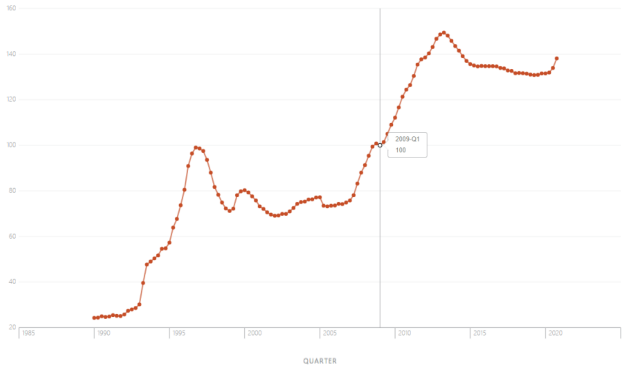

Fig. 5. price index of different year

bad performance is that we have an unbalanced dataset. For some rare flat types and models, many HDB cannot find enough neighbors in the same area. Therefore, the KNN value is not reliable for those rare flat models and flat types. But it still has good performance on the model and type with enough data. (more details are discussed in the evaluation part, see Fig.12. and table 4)

- **Create new features using auxiliary data** : Using auxiliary data, we calculate the counts of infrastructures within a certain range and distance between nearest school, commercial center, hospital, MRT station, and HDB with the BallTree structure (discussed in next section). And use them as new features to denote the convenient level of this HDB. There are two methods to calculate these new features. The first option is to find the shortest distance between the HDB and the different types of infrastructures. However, this might not be very accurate because we do not have the opening year but only the location information for all the infrastructures. For example, a shopping mall is very near to an HDB, which gives these new features a high score. But it hasn't been used until this HDB was sold in a particular year, which means this shopping mall has no effect on this HDB price. To avoid this problem, instead of finding the shortest distance from an HDB to the shopping mall, we try to find all the shopping malls within a given distance, and their corresponding distance to the HDB. Then calculate a weighted score using the following formula:

$$Score = \sum_{1}^{n} R - x_i * C \tag{3}$$

where R is the hyperparameters to denote the distance range, and C is the constant, takes 6371 here. The second method is not relying on only one piece of data that might not be accurate but try to use more information, which is more stable and reliable. After adding these new features, our model gets an excellent improvement based on the validation set. Besides infrastructures, We also calculated the proportion of people aged above 65 and below 15 years old in each subzone using population data.

- **Extra Data collection and extra features**: During this project, we tried to manually collect some extra data and create more features with them. We use part of them to train our final model. The extra data and features are listed below. All of the geolocation data were collected from Wikipedia and the GeoHack website [1]. For school data, we refereed the number one property website [2].
  - **Hospital dataset**: We collected the hospital dataset that contains their opening year, latitude, and longitude. In the experiment, we found that this dataset has a slight improvement for the model. In the final model, we group the hospital dataset with the commercial-centers dataset.
  - **Train station dataset**: We used both the MRT and LRT data to train our final model. And we also tried to collect the data for those statins are the interchange. The opening year for each interchange is collected as well, then we add each interchange as a new data row with the corresponding opening year to the train station dataset. However, those interchange data did not seem very effective with the model in our experiment. We did not use the interchange data in our final model.
  - **School dataset**: We did experiments to see if some famous schools have an impact on the HDB resale price. We manually collected the primary school ranking based on the school popularity, vacancies left after each admission phase. In the experiments, we found that these ranking data have not positive effect on the prediction result. This might be due to the popularity and vacancies of each school is different every year. It is difficult to collect the more robust data in this area, so we did not use the school ranking in our final model.
- **Final features used**:Finally, we use features below for our final model:
  - **Numerical features**: floor area, lease commence data, age of HDB, year and month of sales, storey, prince index of time, average price for different region, average price for different HDB model and type, count and distance of infrastructures.
  - **Categorical features (encoded as one-hot vectors):** flat type and flat model, information about location (such as planning area, town, region and so on).

These features are selected due to their contribution to our predictions which can be reflected from the correlation heatmap (showing in appendix Fig.10.) and importance bar figure (Fig.6.). In correlation heatmap, features with higher correlation with resale price usually mean more importance for our model, which can also be proved in features importance level for our final model (Fig.6.)
It is easy to find that support features we create and new features extracted from auxiliary data are extremely important and useful for our final model, which demonstrates the validity of our feature engineering. And among

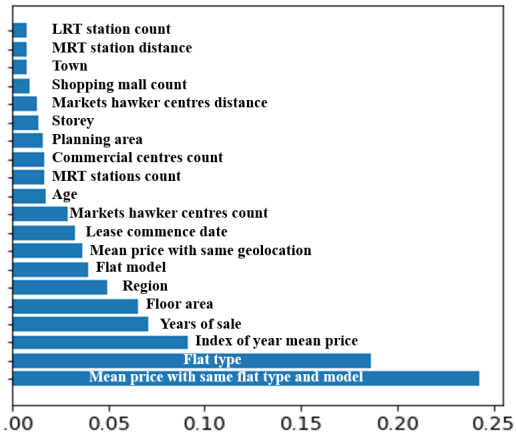

Fig. 6. Feature importance level for our final model

the original features, the Flat Type and Model, location and date information, and floor area of HDB are important for our prediction, which is the same as we expected in the overall feature filtering.

III. Data Mining and Modeling

In this study, we mainly explored ensemble learning methods to model the relation between features and resale prices. Popular ensemble techniques include bagging and boosting. By subsampling the data set and different features, the bagging technique can mitigate the influence of noise while preserving useful information that can be generalized to other unseen data. By stacking multiple weak learners, the gradient boosting method is able to pick up complex information that is difficult for any single player to learn and achieve competitive accuracy. Specifically, algorithms such as XGBoost, LightGBM, CatBoost, and RandomForest are tested in our experiments. Traditional data mining model such as K-NeareastNeighours is also used as a baseline.

1) BallTree (KNN method): The distance between the HDB and the different infrastructures is a very informant feature for predicting the resale price. To make use of these features, firstly, we used the 'geopy' package to calculate the distance between the two different locations using the Vincenty formula. However, the time complexity will be O(mn) for m HDB and n infrastructures, and we have many types of infrastructure. Therefore, we make use of the BallTree data structure to construct the BallTree for each type of infrastructure such as commercial centers, shopping mall hawker centers, etc. This reduces the time complexity to O(nlogn). Then, with the BallTree data structure, it is easy to find the number of infrastructures within the given range, and HDB location. The BallTree structure is from the sklearn package. One limitation of the BallTree is that the distance calculation is using the haversine formula, which is a simpler computation but less accurate compared with the Vincenty formula.

$$haversine(\theta) = haversine(\varphi 2 - \varphi 1) + \cos(\varphi 1)\cos(\varphi 2) haversine(\lambda 2 - \lambda 1) \quad (4)$$

2) Extreme Gradient Boosting (XGBoost): Similar to a standard gradient boosting machine, XGBoost trains one decision tree at a time and freezes the previous tree before training the next one on the residuals. On the other hand, XGBoost adds a regularization term in the objective function to prevent the model from growing too complex. In addition, it searches for best splits at each node by considering the distributions of different features, which largely reduces the search space. It also adopts post-pruning techniques which could potentially push the potential of every single tree to an extreme.

3) Light Gradient Boosting Machine (LightGBM): LGBM works using similar techniques as XGBoost. However, it grows the tree leaf-wise rather than depth-wise. On the other hand, instead of performing random sampling or no sampling on the training data, LGBM adopts the technique known as Gradient-based One-Side Sampling (GOSS), which keeps all the training instances with large gradients from the last step random samples the small gradient instances. It preserves the information needed to make up the residuals while largely increases the efficiency. In general, LightGBM is able to achieve the same performance as XGBoost in a much shorter time.

4) CatBoost: CatBoost uses a modified version of the target-mean-encoding method to deal with categorical features with cardinalities more than maximal one hot size. While the mean encoding assigns each category with the expected target values, it could result in the target leakage problem. To overcome that, CatBoost adopts the idea of online learning where multiple random permutations are generated, and the statistics for each training instance are calculated based only on the available history.

5) Random Forest: RandomForest trains multiple trees at the same level simultaneously and aggregates the predictions afterward. To avoid overfitting, each tree is trained only on a subset of the training data, with a subset of the original features, such that the common patterns could be agreed by the majority while the irrelevant information carried by noises can be deprioritized.

6) Voting Regressor: The voting regressor works as an ensemble wrapper which aggregates the predictions made by each of the regressors, as each learner might have weakness in different situations, and combining their predictions further improve their strengths and offset the weaknesses. Nonetheless, the possible improvement is based on the premise that each player should have a similar capacity.

IV. Evaluation and Interpretation

1) Hyper-parameters XGBRegressor: With the processed data, we start the model tuning. The model performance is not good with the default parameters. Firstly, we increase the tree depth to 10 and increase the number of trees, the model performance is much better comparing with the default setting. Next, we try to use a small learning rate to train the model. Then we need to increase the number of trees when the learning rate is small. During the model tuning, we also use different techniques to avoid the overfitting issue. For example,

increase the minimum sum of instance weight needed in a child of the tree, reduce the subsample ratio of the training set, and subsample ratio of columns when constructing each tree. With different feature selections for the model training, all the mentioned hyper-parameters may need to update.

TABLE I
HYPERPARAMETERS

Parameter	Score (RMSE) on validation set
Default parameter	29991.88
n-estimators=550 max-depth=10	16263.49
n-estimators=550 max-depth=10 learning-rate=0.01	18089.45
n-estimators=8000 max-depth=10 learning-rate=0.01	16041.25
n-estimators=8000 max-depth=10 learning-rate=0.01 min-child-weight=20	15984.73
n-estimators=8000 max-depth=10 learning-rate=0.01 min-child-weight=20 colsample-bytree=0.4 subsample=0.9 reg-lambda=5	15622

2) Analysis on geolocation of facilities: From the auxiliary dataset, there are many different types of a facility like a shopping mall, primary school, etc. In this project, we did different experiments to explore the relationships between the geolocation of facilities and the HDB resale price. For the range to calculate distance weighted score and count, 0.8 km is used for MRT and LRT stations. For the other facilities, 5 km is used.

TABLE II
GEOLOCATION PROCESSING

Geolocation Features Processing	Score (RMSE) on validation set
Use shortest distance from HDB to different types of facility	15715.5
Use the count of the different types of facility within the given range	15701.6
Use the distance weighted score of the different types of facility within the given range	15651
Use both the distance weighted score and the count of the different types of facility within the given range	15622

3) Analysis on categorical feature encoding: In the data processing phase, we tried different ways to handle the categorical features. First, we tried to encode some of the categorical features manually. For example, we tried to encode the 'flat-type into numerical features such as number of bedrooms, number of bathrooms, and living room based on the Government Agency Website [3]. However, we found that using the manual encoding caused poorer model performance compared with using other encoding methods. This might be due to some information lost after manual encoding. We also tried some traditional encoding techniques. From our experiments, we found that for each categorical feature, the different encoding techniques have a slightly different impact on the model performance, but the difference is very small. We used One Hot Encoding in our final model based on the validation result.

TABLE III
ENCODING METHODS

Encoding method	Score (RMSE) on validation set
Label Encoding	15648.3
Binary Encoding	15662.4
One Hot Encoding	15622
Use both the distance and the count of the different types of facility within the given range	15622

4) Analysis on support features: Like we mentioned in the EDA section, we create extra features to help our model learn information from one-hot vectors. For the Flat model and Flat type, we use the mean price of HDB with the same flat model and type as a support feature (denoted as $mean_2$ here). For location information, we use the mean price of HDB with the same district (same region, town, subzone, and planning area, etc. denoted as $mean_3$ here). To get a more accurate mean price, we also do experiments on the combination of the former two mean prices, which means the price of HDB with the same district and same flat type and model. (denoted as $mean_1$ here) Furthermore, based on the method of $mean_1$, we used the KNN method to calculate the mean price of the best K nearest neighbor as our support feature. And the experiment results are listed below:

TABLE IV
SUPPORT FEATURES

Features used in prediction	Score (RMSE) on validation set
Original features	16166
With KNN	16309
With $mean_1$	16144
With $mean_2$	16122
With $mean_3$	16114
With all mean price	16084
With $mean_1$ and $mean_3$	16117
With $mean_1$ and $mean_2$	16120
With $mean_2$ and $mean_3$	16094

From the results, we can easily find that a combination of different methods for mean price gives the lowest RMSE, 16084, which demonstrates our validity of using support features. However, using the KNN method means the price as support harms our predictions. We believe it is because this feature is highly correlated with resale price, which means our model has a high tendency to use it for prediction.

However, this KNN feature has a adversary impact on the prediction because of its lack of generalization ability. KNN tends to use K nearest HDB with the same model and same type. But it is noted that some models and types only have limited samples. In this case, the nearest neighbor may have a long distance from each other, which makes the price not accurate. For models with enough samples, KNN would have

good performance. But for those models with fewer samples, KNN tends to have large deviations (details are shown in Fig.12. The model and type with fewer samples are likely to have a strange and rough curve). With these large deviations, a model may perform badly on predict the price of HDB with these models. That's the reason why the KNN method won't work. Hence, in the final model, we discard KNN and use other mean prices as our support features.

5) Analysis on outliers: To test whether deleting outliers helps our model predict more accurately more not, we conduct the ablation experiments on the validation set and test set. However, the performances on the validation set are nearly the same. Hence, we only list results on the test set in the table below.

TABLE V
OUTLIERS

Features used in prediction	Score (RMSE) on test set
With outliers	15540.5
Without outliers	15435.2

According to the results, we can find out that deleting outliers is not essential for our predictions. It is because using a robust regression model like XGBoost, a few outliers have little harm for our predictions. Since the XGBoost tends to use more significant features and more data to predict and reduce loss by boosting training on the samples having high loss, which makes the model resist single extreme value influence. Therefore, With a large enough dataset, these outliers are too few to influence our predictions by using a robust regression model.

6) Model Performance: As is mentioned in section III, multiple regression models are trained and tuned to predict the resale prices. XGBoost shows the highest precision, followed by LightGBM which has also a competitive RMSE score. On the other hand, although CatBoosting has been claimed to fit the categorical data well, its performance on the validation set is slightly lower than the above two. However, in some aspects, it might still have advantages over XGBoost and LightGBM as its ME is lower. Therefore, We decided to combine the predictions from three regressors together with a weight of $(0.5, 0.4, 0.1)$ to emphasize the advantages while offsetting the mispredictions. The voting mechanism shows a slightly stronger performance compared to all the singletons. Our final submission achieved 5th place out of the total 64 teams in the private leaderboard[2].

TABLE VI
PERFORMANCE OF DIFFERENT MODELS

Method	RMSE	ME (Max Error)
XGBoost	15622	243537.75
LightGBM	15677	251363.54
CatBoost	16482	**234381.94**
Voting Regressor	**15603**	244693.68

[2]https://www.kaggle.com/c/cs5228-2020-semester-2-final-project/leaderboard

7) Error Analysis: From Figure 7, predictions made on 96.7% of the properties have an error of less than 40000, while the rest 3.3% of the data increases the RMSE by around 3000 (20%). Thus, further analysis is conducted to analyze the aspects that the model is incompetent at.

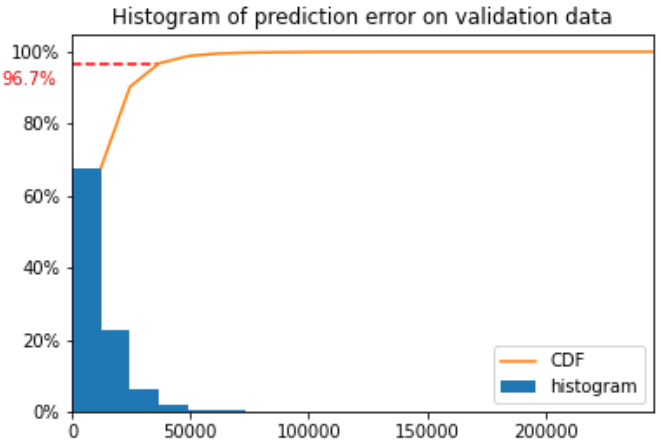

Fig. 7. Histogram & CDF of prediction error

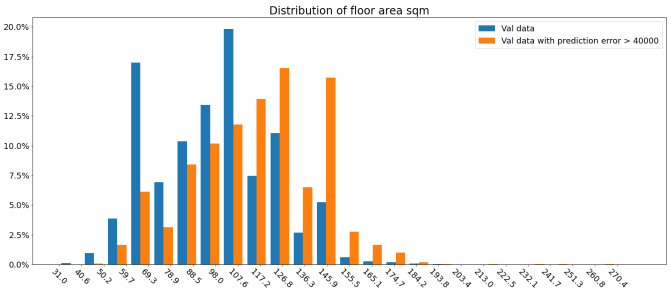

Fig. 8. Histogram of floor area (sqm)

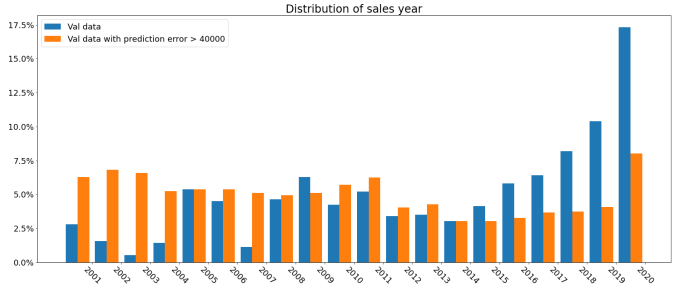

Fig. 9. Histogram of sales year

From the perspective of floor area (Figure 8), the prediction models tend to generate larger errors for properties with around 120 sqm, while most of the properties invalidation and training data have floor areas around 100 and 60 sqm. This could be probably due to limited training data with large sqm. For sales year (Figure 9), the model tends to perform worse for properties sold from 2017 to 2020, while the training data

and validation data are almost uniformly distributed among all sales years. Therefore, the poor performance on these sales years could be because more independent variables are needed for explaining the resale prices. In addition, there is also an interesting phenomenon that there are neglectable new leases from 2017 to 2020, which coincides with the time period of the abnormal properties sold and there could some association between them.

REFERENCES

[1] GeoHack, urlhttp://www-cs-faculty.stanford.edu/~uno/abcde.html

[2] Numberoneproperty, urlhttps://numberoneproperty.com/top-10-tips-for-getting-into-good-school-in-singapore

[3] Government Agency Website, urlhttps://www.hdb.gov.sg/residential/buying-a-flat/resale/getting-started/types-of-flats

[4] Government Agency Website, urlhttps://data.gov.sg/dataset/hdb-resale-price-index

[5] Government Agency Website, urlhttps://www.citypopulation.de/en/singapore/admin/

[6] listendata, urlhttps://www.listendata.com/2019/08/WOE-IV-Continuous-Dependent.html

APPENDIX

A. Breakdown of workload

Here listed the workload distribution of each group member:

TABLE VII
BREAKDOWN OF WORKLOAD

name	work load overview
Liang Bowen	Data collection for price index of HDB, EDA, visualization and feature engineering, support features experiments, feature selection and ablation experiments
He Yingxu	Proposed one way of scoring neighbourhood facilities based on distance. Tuned hyperparameters of XGBoost together with Xuhui. Experimented XGBoost, LGBM, CatBoosting as well as Voting Regressor.
Huang Xuhui	Data collection for hospital, MRT/LRT station, and school ranking. Geolocation data experiments and processing using geopy and BallTree.
Xu Xuanyue	Identifying outliers by differemt ways, converting category variables, testing different regression models

B. figure of experiments

Here listed the some important supplementary figures:

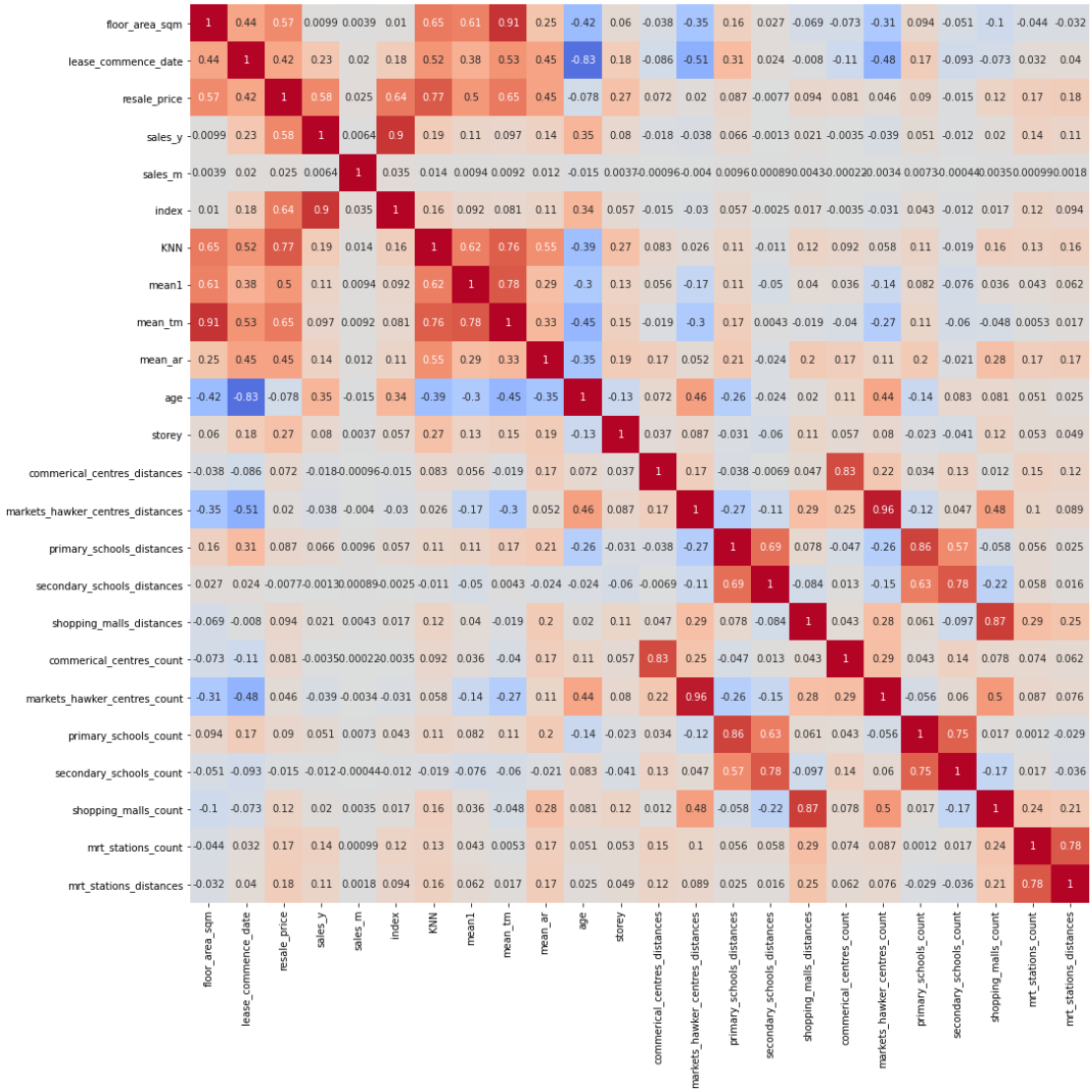

Fig. 10. correlation heatmap

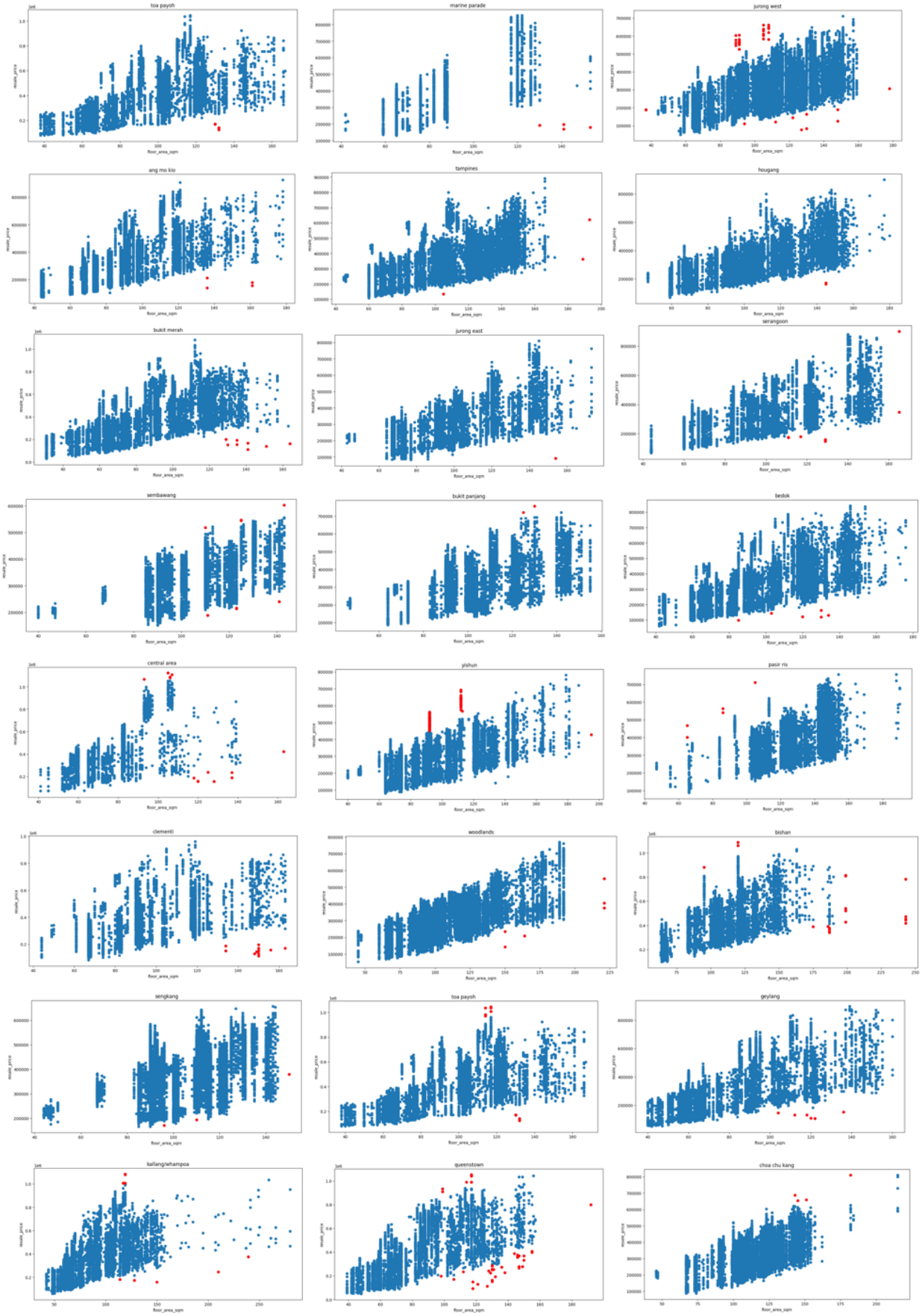

Fig. 11. all outliers in different towns

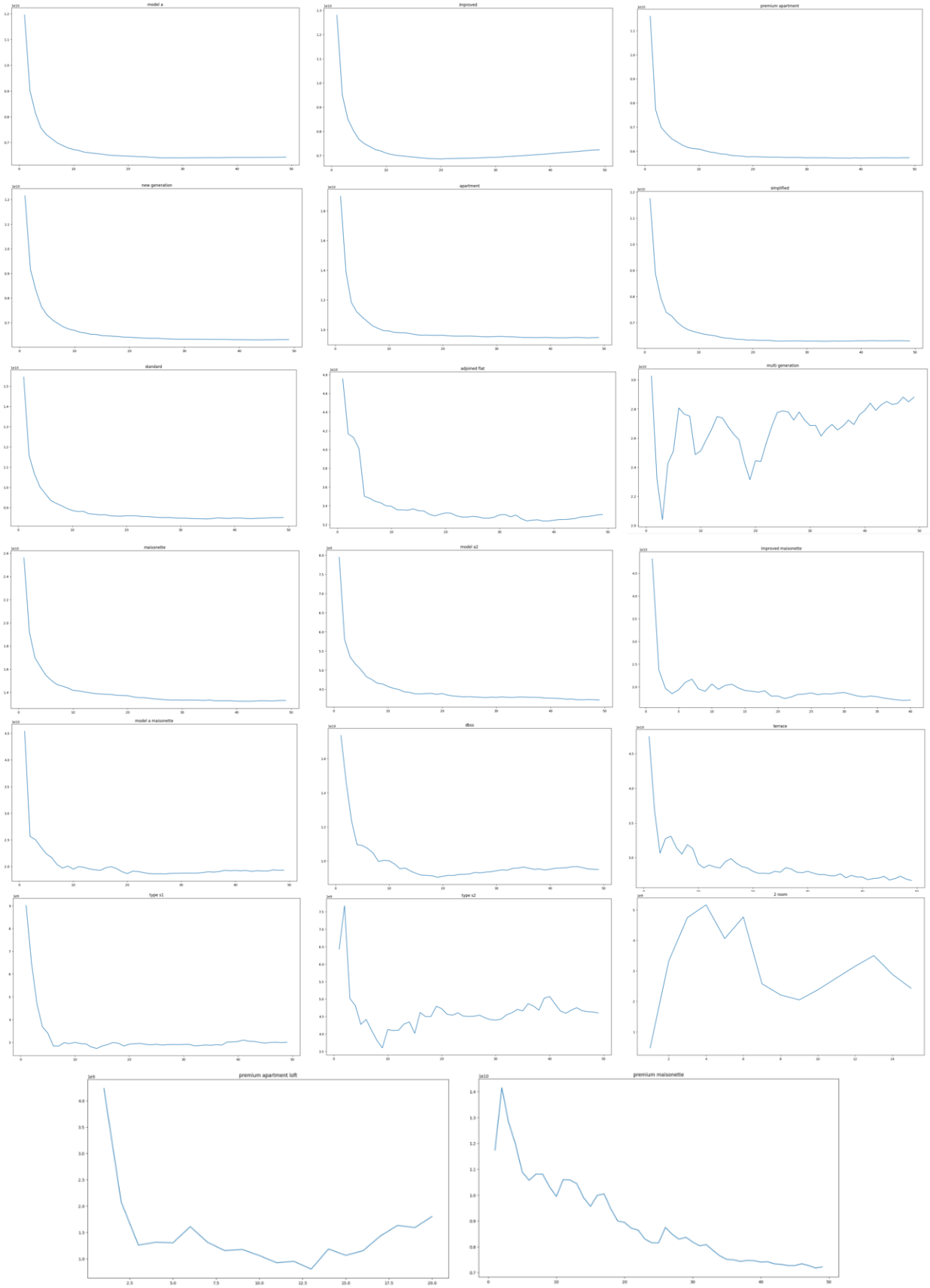

Fig. 12. KNN curve: MSE changes with different K