Two Sigma Connect: Rental Listing Inquiries

Group 10
Yiqing Liu, *yq.liu@rutgers.edu*
Wenzhe Zhang, *wenzhe.zhang@rutgers.edu*
Guanyu Huang, *gh292@scarletmial.rutgers.edu*

***Abstract*—This project is a Kaggle competition which intended to help Renthop better understand needs and preferences of renters. We did feature engineering and built classification models by SVM, neural network, random forest and XGBoost. We used ensemble learning to update four model results with better predictive performance and lower variance. There is a high correlation between price and interest level.**

***Keywords*—SVM, MLP, Random Forest, XGBoost.**

I. BACKGROUND

We choose the project and dataset from Kaggle competition [1]. The purpose of this project is to predict the interest level in an apartment rental listing.

How much interest will a new rental listing on RentHop be received? This is a key issue to be solved, which has great bussiness value in that it not only helps control fraud better but also help the company to understand customers' preferences and needs.

We hope that our research can help RentHop better recognize potential listing quality problems, and allow owners and agents to make better listing choices to attract customers.

II. DATA AND FEATURE ENGINEERING

A. Data

In our dataset, we have about 49,000 entries with label as training data and 74,000 test data. But in our studies, we mainly used labeled data.

1) Independent Variables:

- bathrooms: number of bathrooms
- bedrooms: number of bathrooms
- building id: reference of building
- displayed address
- features: a list of features about this apartment
- created description: the description of the rentals, a texture feature.
- latitude: geolocation used to classification
- longitude: geolocation used to classification
- manager id: Managers reference
- price: in USD
- street address: Locations

2) Response Variables: interest level: this is the target variable. It has 3 categories: 'high', 'medium', 'low'

B. Feature Engineering

Feature engineering is the process of using original data to create features. It is fundamental to the application of machine learning and is both difficult and time-consuming. Sometimes, feature engineering even decides whether our model is good or not.

bathrooms	bedrooms	building_id	created	description	display_address	features
1.5	3	53a5b119ba8f7b61d4e010512e0dfc85	6/24/16 7:54	Oven & LG Fridge? washer /dryer in the apt? Cable	Metropolitan Avenue	[]
1	2	c5c8a357cba207596b04d1afd1e4f130	6/12/16 12:19		Columbus Avenue	tor', 'Fitness Center', 'Cats Allowe
1	1	c3ba40552e2120b0acfc3cb5730bb2aa	4/17/16 3:26	d and conveniently located near A,C,E,L,1,2,3 train	W 13 Street	shwasher', 'Hardwood Floors', 'Pe
1	1	28d9ad350afeaab8027513a3e52ac8d5	4/18/16 2:22	t E,M,6,7 This Apartment also feature a	East 49th Street	['Hardwood Floors', 'No Fee']
1	4	0	4/28/16 1:32	ess steel appliances including a dishwasher and a r	West 143rd Street	['Pre-War']
2	4	38a913e46c94a7f46ddf19b756a9640c	4/19/16 4:24		West 18th Street	[]
1	2	3ba49a93260ca5df92fde024cb4ca61f	4/27/16 3:19	r@renthop.com 766-103-4663Hablo espanol!Build	West 107th Street	ed', 'Cats Allowed', 'LOWRISE', 'SI
2	1	0372927bcb6a0949613ef5bf893bbac7	4/13/16 6:01	ss app,dish washer,closets,marble bath,laundry,ele	West 21st Street	ice', 'Laundry in Unit', 'Dishwashe
1	1	a7efbeb58190aa267b4a9121cd0c88c0	4/20/16 2:36	<p><a website_redacted	Hamilton Terrace	'Dogs Allowed', 'Elevator', 'Laun
2	4	0	4/2/16 2:58	ut notice all the neighborhood bars, restaurants, c	522 E 11th	['Dishwasher', 'Hardwood Floors'

interest_level	latitude	listing_id	longitude	manager_id	photos	price	street_address
medium	40.7145	7211212	-73.9425	5ba989232d0489da1b5f2c45f6688adc	os.renthop.com/2/7211212_c17853c4b869af6f53af08b(	3000	792 Metropolitan Avenue
low	40.7947	7150865	-73.9667	7533621a882f71e25173b27e3139d83d	s.renthop.com/2/7150865_af28a5075bd321e69479164	5465	808 Columbus Avenue
high	40.7388	6887163	-74.0018	d9039c43983f6e564b1482b273bd7b01	f6ea4796b9d177786bb.jpg', 'https://photos.renthop.co	2850	241 W 13 Street
low	40.7539	6888711	-73.9677	1067e078446a7897d2da493d2f741316	os.renthop.com/2/6888711_44fddcd3ff7fb74ea1240882	3275	333 East 49th Street
low	40.8241	6934781	-73.9493	98e13ad4b495b9613cef886d79a6291f	s.renthop.com/2/6934781_742045d91f2a0c9acbd939ce	3350	500 West 143rd Street
medium	40.7429	6894514	-74.0028	b209e2c4384a64cc307c26759ee0c651	.renthop.com/2/6894514_ab18ba34a854348a3d39bfac	7995	350 West 18th Street
low	40.8012	6930771	-73.966	01287194f20de51872e81f660def4784	22f4f6df8c257ee95ca8.jpg', 'https://photos.renthop.cor	3600	210 West 107th Street
low	40.7427	6867392	-73.9957	e6472c7237327dd3903b3d6f6a94515a	s.renthop.com/2/6867392_db8f0216c0c98788e08692b6	5645	155 West 21st Street
medium	40.8234	6898799	-73.9457	c1a6598437b7db560cde66e5a297a53f	.renthop.com/2/6898799_31105b3786c2aa5cbed8706l	1725	63 Hamilton Terrace
low	40.7278	6814332	-73.9808	23a01ea7717b38875f5b070282d1b9d2	os.renthop.com/2/6814332_3e2accbf6dfc5c8b78ea73a	5800	522 E 11th

Fig. 1. Features

Here we have 15 variables at first including our response variable. Then we are going to discuss how to deal with them, respectively.

1) Numerical features: There are three numerical variables: 'bathrooms', 'bedrooms' and 'price'. Based on them, we created several features which might be useful:

Financial Statistics and Risk Management, Rutgers University 'bedrooms+bathrooms', 'price to bedrooms' and 'price to bathrooms'. For the last two features, we apply the function of

$$y = \begin{cases} \log(1 + price/x) & x \neq 0 \\ 0 \quad x = 0 \end{cases}$$

where x is the number of bedrooms or bathrooms.

2) Categorical features: For 'building id', 'display address', 'manager id' and 'street address', we turn them into categorical features because some of the addresses or ids don't appear only once. Therefore, we categorize them into 1 to 999 and the others all 1000 according to their frequency.

3) Time: We first turn the time variable into 'month', 'weekday', 'hour'. Then we transform the 'weekday' variable into seven dummies from Monday to Sunday, and the other two are numerical.

4) Text: Text is the most important part of the process of feature engineering. There're mainly two texts need to be dealt.

Description:

- Percent of uppercase words
- Number of special characters like $!, <, \$, *$ etc.
- Length of description
- Phone number dummy. If it contains the phone number, 1 and 0, otherwise.
- Email dummy. This is just the same as Phone dummy creation.

Features:

- Calculate the frequencies of all the words in all the features
- Select the most frequent words appeared(here we choose the most frequent 60 words)
- Create dummies for these 60 words, which we regard as important in attracting customers.

5) Location: We consider only longitude and latitude can't represent location feature. Therefore, we apply the following methods:

1) Explore the data by plotting them on a figure. We find some outliers, however, we decided not to drop them.
2) Apply K-means cluster,

$$argmin\Sigma_{i=1}^{k}\Sigma_{x\in S_i}||x-\mu_i||^2$$

where k is the number of centers and μ_i is the mean value of points within set S_i.

This is a very classic method for data mining, and in our project we set k=6. After applying the clustering method, we can get two features. The first is the label of location from 0 to 5 and the second is the straight distance of every point to the center of their label.

6) Others: The only variable left to be dealt is 'photos'. Here we drop it because we have to deal with over 60G picture data and we don't have enough time and good facilities to conduct this when applying machine learning methods. Therefore, we have to drop it with regret.

In conclusion, we build our features according to experience and read some articles as well as some discussions of some other kagglers. All what we do is to try building the most useful features for predicting people's interest level. Here we don't normalize or center the data because our teammates have different models to apply and they may transform it if needed.

III. Methodologies

A. SVM

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. As for the dataset from Renthop, we could apply the support vector machine to do the classification. The response has three classes. The classification for multiple classes SVM has two methods, which is one-against-one or one-against-rest. We decided to implement the two above to solve the multi-classification problem.

B. Neural Networks

Artificial neural networks (ANN) is one of the most helpful neural network[2]. The strength of neural networks comes from their ability to learn the representation in training data and how to best relate it to the output variable. Multi-layer Perceptron classifier is a method of ANN to solve classification problem.

Neural networks include neurons. Neurons are simple computational units which have weighted input signals

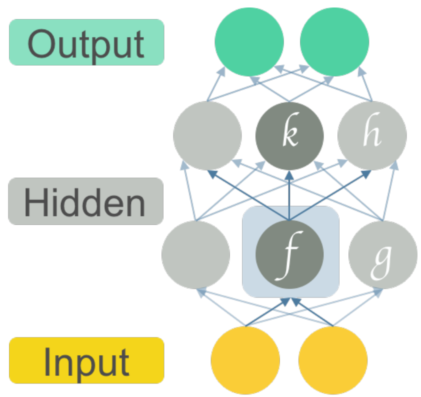

Fig. 2. Artificial neural networks

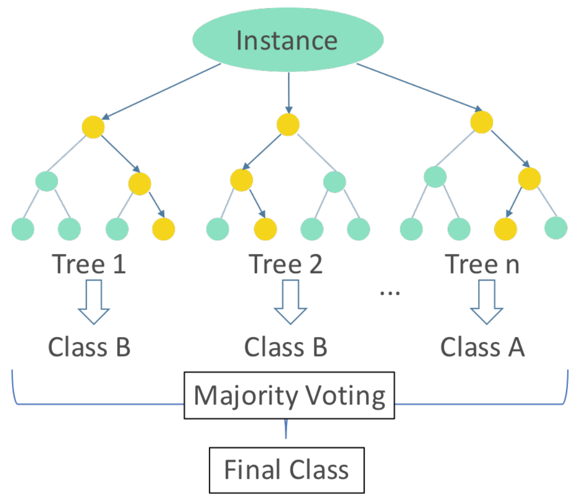

Fig. 3. Random Forest

and produce an output signal using an activation function. A row of neurons is a layer and one network can have multiple layers. The bottom layer is input layer where original data come from. Layers after the input layer are called hidden layers. More hidden layers mean more complicated model. The final hidden layer is called the output layer, and it is responsible for outputting a vector of values. A multi-class classification problem may have multiple neurons in the output layer, one for each class.

C. Random Forest

Decision trees are a popular method for various machine learning tasks. Random Forest builds a large amount of decorrelated decision trees on bootstrapped training samples, which is an improvement based on bagging. It's invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. Therefore, random forest has strong generalization ability and is a good choice for our project

D. XGBoost

Gradieant boosting is a machine learning technique for classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion as other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

XGBoost (Extreme Gradient Boosting) is a scalable machine learning system for tree boosting that is widely used by data scientists and provides state-of-the-art results on many problems[3]. XGBoost provides a parallel tree boosting that solve many data science problems in a fast and accurate way.

IV. Experiments and Results

A. Experiment Steps

To build classification models with different algorithms, we designed following experiments steps:

1) Split data into train set and testing set, and normalizing X_i.
2) Use cross-validated grid-search [4] find optimized parameters by minimizing log loss.
3) Build the best classifier with the optimized parameters.
4) Predict on the testing set by the best model.

B. Evaluation Metrics

We used log loss as main metric to select models and compare model results by precision, recall, and f1-score.

Models are evaluated using the multi-class logarithmic loss. For each listing, models generate a set of predicted probabilities. The log loss calculate by:

$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\ln(p_{ij})$$

where N is the number of listings in the test set, M is the number of class labels. y_{ij} is 1 if observation

i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j.

Precision is defined as

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

Recall is defined as

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

F1-score is defined as

$$\text{F1} = 2\frac{\text{precision} \times \text{Recall}}{\text{precision} + \text{Recall}}$$

C. Results

The results of four methods are shown below Table. XGBoost has smallest log-loss score. MLP and Random Forest have similar performance.

TABLE I. LOG-LOSS OF FOUR MODELS

	SVM	MLP(NN)	Random Forest	XGBoost
Log-loss	0.6539	0.6259	0.5921	0.5509

For SVM model, we chose parameter by cross-validation, which C = 1, Kernel =radial, gamma = 0.01, decision function shape = one vs one. The LogLoss is 0.6539, however, the F1-score of the low interest level performs better than high and medium interest level. We find the imbalanced data size is the main reason. Only 8 percent of entries are high interest level. For example, when limited numbers of high interest level entries were surrounded by other interest entries, SVM may not be very effective as we expected.

TABLE II. SVM

	Precision	Recall	F1-score	Support
Low	0.72	0.98	0.83	10312
Medium	0.43	0.08	0.13	3316
High	0.62	0.02	0.05	1178
Ave/Total	0.64	0.70	0.61	14806

For MLP model, we used three hidden layers, which numbers of neural is 10, 40, 5 respectively. The LogLoss is 0.6259.

Using grid search and cross-validation on a set of parameters, we found best random forest model with 1000 trees, Gini index criterion and 10 maximum features. The LogLoss equals 0.5921. However, RF is likely to be overfitting and this may be caused by maximum depth. We tried several values for the parameter and the result shows the best value for it is unlimited, which means keep searching until the best. Therefore, how to balance the model performance and overfitting problem may be another issue to be solved in the future.

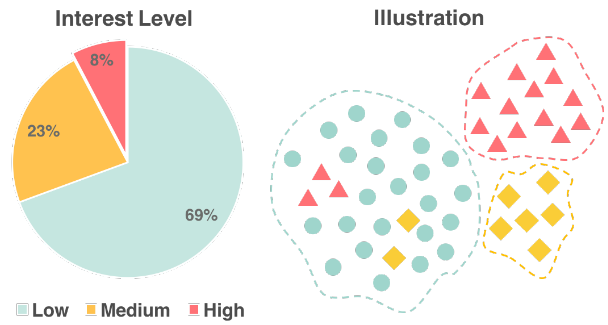

Fig. 4. Imbalanced Classification Problems

TABLE III. MULTI-LAYER PERCEPTRON

	Precision	Recall	F1-score	Support
Low	0.80	0.90	0.84	10312
Medium	0.44	0.35	0.39	3316
High	0.54	0.27	0.36	1178
Ave/Total	0.7	0.72	0.70	14806

TABLE IV. RANDOM FOREST

	Precision	Recall	F1-score	Support
Low	0.79	0.94	0.86	10312
Medium	0.48	0.29	0.36	3316
High	0.55	0.23	0.32	1178
Ave/Total	0.70	0.74	0.70	14806

TABLE V. XGBOOST

	Precision	Recall	F1-score	Support
Low	0.82	0.92	0.87	10312
Medium	0.50	0.41	0.45	3316
High	0.59	0.29	0.39	1178
Ave/Total	0.73	0.76	0.74	14806

For XGBoost, numbers of learning rounds is 793 and learning rate is 0.03. We got 0.5509 LogLoss on test data. Combined with F1-score, XGBoost performance best among the four models.

As four models we used, all of them perform well on low interest level, but not well on other two groups, especially SVM model. Generally, the performance of

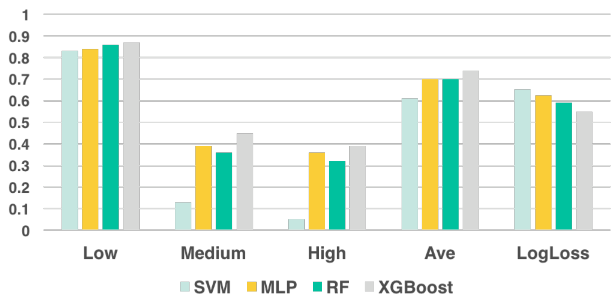

Fig. 5. F1-Score and Log Loss of Different Methods on Test Data

XGBoost is best. Random forest and neural network are similar, which better than SVM but not good as XGBoost.

Compared the results of training and test data set, SVM and neural network have consistent performance. But, random forest and XGBoost perhaps have the over-fitting problem, because both F1-score and LogLoss of the two methods have significant differences between training and test data set.

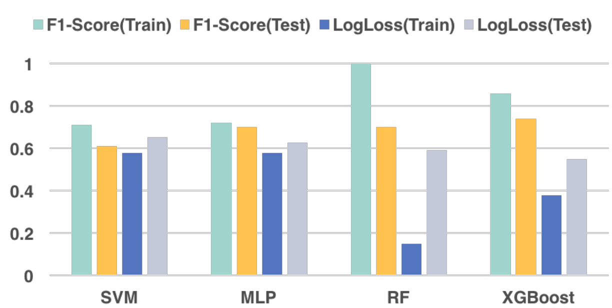

Fig. 6. Training vs. Test

Therefore, we faced imbalanced classification problems. How to solve it? First, we tried different methods. The tree models are better than SVM model. Then, we can try to generate synthetic samples by SMOTE method.

To obtain better predictive performance with low variance, we tried ensemble learning. We used the weighted average ensemble method to summarize the predict results of four models. The performance of ensemble model that LogLoss is 0.5543 is similar to XGBoost model and better than other three. Reasons for no significant improvement of ensemble model are that four models used same input features and the numbers of models are limited.

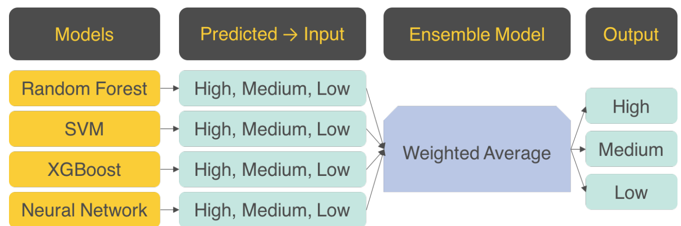

Fig. 7. Ensemble Learning

V. Conclusion

Tree models perform pretty well in this problem, especially XGBoost. Based on the model we built, the most important feature is price. Not only the total price of an apartment, but also the unit prices of bedroom and bathroom are significant explainable variables. Moreover, we find that manager of rental listing has a high correlation with interest level, which Renthop can do more research on it.

Acknowledgment

We would like to thank Zijian Guo, Chencheng Cai and Ting Yang. They guidance along with their unconditional support towards making this paper original proved to be an asset.

References

[1] Kaggle, "Two sigma connect: Rental listing inquiries — kaggle,," 2017, [Online; accessed 27-Nov-2017]. [Online]. Available: https://www.kaggle.com/c/two-sigma-connect-rental-listing-inquiries

[2] J. Brownlee, "Crash course on multi-layer perceptron neural networks," 2017, [Online; accessed 27-Nov-2017]. [Online]. Available: https://machinelearningmastery.com/neural-networks-crash-course/

[3] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.