Milestone 2

Jin Young Kim

Insoo Rhee

Weibao Sun

1. **Which features did you select for your classifiers? Please comment on the reason for your feature selection. If you choose to work on the bonus question, you can add your features extracted from external datasets at this step. (5 points)**

The features selected are price, latitude, longitude, number of bedrooms, number of bathrooms, and features. These attributes were selected as they are likely important determinants of a listing's popularity. To improve classifier performance, feature selection may be performed using methods such as Fisher score or PCA before performing the classification to run the classifier on most relevant features only. For bonus, we previously learned that all listings were from Manhattan area by mapping them based on longitude and latitude. We derived an additional attribute named distance from Soho, which is the busiest area in Manhattan. The rationale behind adding the attribute is that listings tend to be more popular in the downtown area and can thus provide our classifiers with more information about a listing's popularity.

2. **What Python or R libraries did you use for your classifiers? (5 points)**

The sklearn was used to import the classifiers (decision tree, logistic regression, and SVM) and methods for calculating the performance metrics of the classifier. The Pandas library was also used for data manipulation.

3. **How did you perform cross-validation? Please describe the procedure. (10 points)**

Cross-validation was performed using the *train_test_split* and *cross_val_score* methods from the sklearn library. Data was first split into a training dataset and a test dataset in a 70:30 ratio (for SVM and logistic regression) or 75:25 ratio (decision tree) using *train_test_split*. Cross validation was performed on the training dataset using *cross_val_score*. The method splits the training dataset into k-1 training subsets and 1 validation set and returns the average accuracy across the k folds. For our regression analysis, k was set to 5.

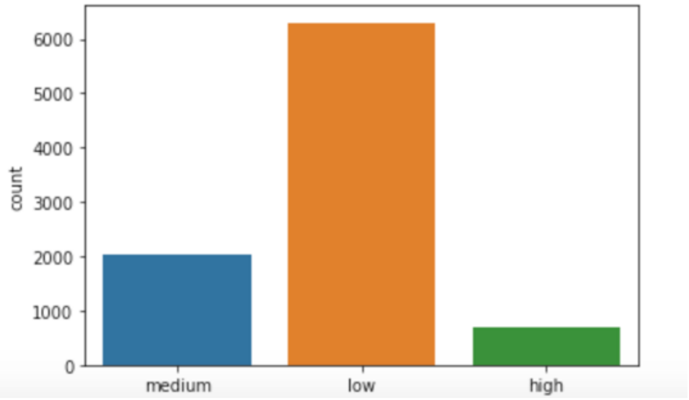
4. What performance did the first version of your classifiers achieve on the validation dataset (in cross-validation) and on the test dataset? Please comment on the performance of the classifier. (15 points: 5 points for performance, and 10 points for comments).

The first decision tree achieved an accuracy of 0.6256 on the validation dataset and 0.6396 on the test dataset. Notably, the classifier achieved a 100% accuracy on the training dataset, indicating overfitting.

The first regression model achieved an accuracy of 0.6856 on the validation dataset and 0.6890 on the test dataset. There is no overfitting as the validation test achieves higher accuracy than the test dataset.

The first SVM achieved an accuracy of 0.6911 on the validation set and 0.6923 on the test set. Based on these values, there is no overfitting.

While 63% - 69% accuracy looks like a decent performance (since the likelihood of a random guess of a class label being correct is ~33.3%), a closer look at the distribution of the interest levels shows that around 75% of the listings have low interest levels (Table 1). This means that simply guessing all listings as "low" can produce 75% accuracy. Further modifications are needed to improve model performance.



.

Table 1. Distribution of interest levels.

5. What actions did you take in order to improve your classifiers? You can modify your dataset or the parameters of your classifier. Please record your modifications in your report. (30 points: 10 points for each improvement)

Following modifications were performed on the decision tree: outlier removal, tuning of the max-depth parameter for the decision tree. Any data points deemed outliers based on the interquartile range rule were removed before classification. Decision tree classifiers are highly biased by nature. The maximum depth of the tree was tuned to reduce bias.

Following modifications were performed on the regression model: additional removal of irrelevant features, data normalization, and regularization. First, each of the selected features in our dataset was sequentially removed until an improvement in accuracy was observed. The dataset was updated to contain the original attributes minus the attribute whose removal was found to improve accuracy. For instance, removing listing_id increased the accuracy of our regression model whereas removal of the remaining attributes did not. Consequently, listing_id was removed from the dataset. Next, data was normalized using the normalizer function provided by sklearn. Lastly, data regularization was performed by specifying the inverse-regularization parameter C. After tweaking the C value between -10 and 10, we found C=7 to maximize accuracy and F1 score while minimizing log loss.

Following modifications were performed on the SVM: Outlier removal, kernel setting, and tuning the gamma value. Sklearn provides different kernel options for the SVM. Training the SVM with the radial basis function kernel increased classifier performance. Accuracy was further improved by tuning the gamma value. Gamma defines how much the influence a single training example has. Increasing the gamma from the default value of 10 to 15 improved accuracy.

6. How did you check whether any overfitting occurred during your training? Did you observe overfitting? What did you do to avoid overfitting? (10 points)

Overfitting was checked by comparing between the training accuracy and test accuracy. If the model achieved a higher accuracy on the training dataset than the test dataset, the model was deemed overfitting.

Decision tree – Overfitting was initially observed with the training accuracy=100% and the test accuracy =0.6393. Outlier removal and tuning the max-depth parameter and the ratio between training and testing accuracy helped reduce overfitting. Creating a smaller training set and a bigger testing set reduced training accuracy but produced better performance on the testing set. Overfitting was reduced after the modifications with only a 0.007 difference between the training and test accuracies.

Logistic regression – Overfitting was not observed in the initial model with a test accuracy being higher than the training accuracy by 0.0087. The modifications help to improve the classifier performance all help to avoid overfitting. By reducing the dataset to contain only the relevant features and data points, we can prevent our model from becoming excessively complex. Regularization helps reduce overfitting by penalizing large weight coefficients so that our model effectively filters out noise during training. Overfitting was not observed in our final model with training accuracy of 0.6898 and test accuracy of 0.6937.

SVM – The initial model showed no overfitting with both the training and test accuracies approximately equal to 0.68. Outlier removal and kernel tuning improved the testing accuracy to 0.710 and the test accuracy to 0.714. No overfitting was observed in the final model.

7. What performance did you achieve on the validation dataset (in cross-validation) and on the test dataset after your modifications? Please, try to explain the gains. (15 points: 5 points for performance, and 10 points for explanation)

The decision tree improved in accuracy from 0.6226 and 0.6457 for the validation and test sets, respectively, before the modifications (outlier removal and tuning of the max-depth parameter and training/test dataset size ratio) to 0.6888 and 0.6811 after the modifications. These accuracies were achieved when the tree depth was set to 15. Outlier removal also helped increase classifier performance as more relevant data allow for better splitting decisions.

Logistic regression without data normalization or feature removal produced accuracy of 0.6856 on the validation dataset and 0.6879 on the test dataset. Following feature removal, the accuracy increased to 0.6904 and 0.6946 for the respective datasets. Data normalization further increased accuracy to 0.6998 and 0.7037, and regularization to 0.7031 and 0.7033 for the respective datasets.

The preprocessing the dataset and tuning the parameters for SVM improved accuracy score by 0.30 from 0.6845 to 0.714 on test dataset. The cross-validation dataset also improved from 0.6845 to 0.710.

8. Evaluate one additional evaluation metrics mentioned in class on the validation dataset. Which metric did you use? What were the results? How do these results compare to the results for multi-class logarithmic results for multi-class logarithmic loss? (10 points)

F1 score was used as an additional evaluation metrics. Large F1 scores indicate better classifier performance while large log loss indicates the opposite.

The initial decision tree had a F1 measure of 0.6423. The series of modifications (outlier removal and depth tuning) improved the F1 score to 0.61. Log loss decreased from 12.457 for the initial tree to 4.8 after the modifications. The F1 score increase and log loss reduction indicate improved classifier performance.

The initial regression model had a F1 measure of 0.5837. The series of modifications (removal of irrelevant feature, normalization, and regularization) increased the F1 measure to 0.5973, 0.6152, and 0.6251, respectively. The initial model had log loss of 0.8771. The modifications reduced log loss to 0.7219, 0.6977, and 0.6945, respectively.

The initial SVM model had a F1 measure of 0.613. The modification (removal of outlier, converting ‘feature’ texts to figures, and tuning the parameters) increased the F1 measure to 0.654. The initial model had log loss of 0.760. After the modification log loss was improved by indicating lower scores of 0.744.

9. Bonus (10 points): You can combine your data with other, related datasets to create additional relevant features, for example, based on the nearby subway stations and malls. Which additional features did you create? By how much did these features improve the performance? If you do not train two different versions of your classifier (with and without the additional features), what evidence do you have that the additional features helped?

As previously stated, we derived an additional attribute, distance from Soho, which is the busiest area in Manhattan. This attribute was derived by subtracting the longitude and latitude of each listing from that of Soho. The accuracy of all classifiers improved with the new attribute added to the dataset. The accuracy of the decision tree model was increased from 0.6457 for the test set to 0.6811 simply by adding the new attribute in the dataset without any modifications. Likewise, the test accuracy of logistic regression increased from 0.6777 to 0.6877 for the logistic regression model. The test accuracy of SVM stayed the same at 0.714.