In this milestone, you will train on the data that has been preprocessed in milestone

1. You can choose among the following three classifiers:

1. Decision Tree

2. Logistic Regression

3. SVM

Each group member has to choose a different classifier and train it on the dataset. If a group only has two team members, two out of the three classifiers shall be trained. Cross-validation (5-fold or 10-fold) should be performed on the training dataset (train.json). The evaluation metric should be multi-class logarithmic loss as mentioned in the evaluation section in Kaggle. Once the predictions for the test dataset (test.json) are generated, please upload them to Kaggle to obtain the accuracy for the test dataset. The detailed steps are in the evaluation section in Kaggle.

In your submission, please answer the following questions:

------------------------------------------------------------------------------------------------------------------------------------------

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 final features I ended up using were:

* Bathrooms
* Bedrooms
* Latitude
* Listing\_id
* Longitude
* Price
* # of photos
* # of features
* Hour
* Day
* Year
* # of description words
* Distance to nearest subway station

I used a recursive feature elimination to determine which features were best.

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

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Glob
* Sklearn
* Scipy
* Datetime
* Pillow
* Skimage

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

Cross-validation is performed using the sklearn library, specifically sklearn.model\_selection.cross\_val\_score. The parameters I input were the logistic regression model, X (the dataframe containing features), y (the dataframe containing classes), and cv = 10 (which specifies the cross-validation function to perform 10-fold validation).

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).

On the validation dataset, my first version of the Logistic Regression classifier achieved a performance of:

[0.76220999 0.76303064 0.76705655 0.76369677 0.76246301 0.76420182 0.76493154 0.76864835 0.76738619 0.76932477], which averages out to 0.765294963.

On the test dataset, it achieved a score of 0.76616.

This score represents the multi-class logarithmic loss (log loss). We want to minimize the score on log loss, so this is a poor performance. The top performers on Kaggle achieved a score of just under 0.5.

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)

First, I added a function that de-duplicates the ‘features’ column. There are some values in the features column that are repeated, which lowers the accuracy of the model.

Second, I used a recursive feature elimination function to find the least important feature. With this, I removed ‘month’.

Lastly, I replaced the appropriate missing values with the median value in the features.

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)

To check overfitting, I compared my cross-validation scores with the test score that I received in Kaggle.

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 final performance I achieved through cross-validation was 0.76517221. The data was [0.76207378 0.76291682 0.76696038 0.76359589 0.76232369 0.76407492 0.7648164 0.76854531 0.76729301 0.7692511]. The test score was 0.76601.

I think that the cross-validation achieves a higher score because the model is built from training data and tested on training data. Therefore, it might be over-fitted slightly, and this will result in a higher validation score.

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 loss? (10 points)

The other metric that I used is Classification Accuracy. The results were: [0.69334148 0.69334148 0.69331025 0.69331025 0.69345147]. It is difficult to compare these results with the multi-class logarithmic loss because for accuracy, the closer the value to 1, the better. For log loss, the closer the value is to 0, the better.

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?

I added a dataset with information containing the latitudes and longitudes of subway stations. Then, I calculated the distance of each property to the nearest station and added that as a feature. This feature, surprisingly, did not improve performance by much.