RENTAL LISTING INQUIRIES

Team:

Akash Dobaria
Anamika Paul
Khushbu Durge
Akhauri Prateek Shekhar
Swapnil Sagar
Urvi Patel

Problem Statement

- **PROBLEM**: Apartment Rental Listing Inquiry
- **DATA SOURCE**: Kaggle
- **EXPECTED OUTPUT**: Interest Level (High, Medium, Low)

DATA FIELDS

- bathrooms: number of bathrooms
- bedrooms: number of bathrooms
- building_id
- created
- description
- display_address
- features: a list of features about this apartment
- latitude
- listing_id
- longitude
- manager_id
- photos: a list of photo links
- price: in USD
- street_address
- interest_level

❖ APPROACH 1:

Selecting only features having quantitative values and initially ignoring categorical values.

	bathrooms	bedrooms	latitude	longitude	price
10	1.5	3	40.7145	-73.9425	3000
10000	1.0	2	40.7947	-73.9667	5465

K-NN Classifier:
Precision: 0.57
Recall: 0.49
F-score: 0.52

Multinomial Naive Bayes:
Precision: 0.58
Recall: 0.39
F-score: 0.44

SVM_RBF_Kernel Classifier:
Precision: 0.63
Recall: 0.70
F-score: 0.64

SVM_Poly_Kernel Classifier:
Precision: 0.58
Recall: 0.47
F-score: 0.55

Multinoulli_Naive_Based Classifier :
Precision: 0.58
Recall: 0.39
F-score: 0.44

SVM_Linear_Kernel Classifier:
Precision: 0.57
Recall: 0.69
F-score: 0.57

❖ APPROACH 2:

Expanding on the features column. Here, “**features**” contains a list of several attributes corresponding to every listing ID.

Adding each individual attribute as a column to the main data frame and assigning binary values:

- **0**- if the listing ID does **not** contain that feature.
- **1**- if the Listing ID contains that feature.

```
10                                                          []
10000      [Doorman, Elevator, Fitness Center, Cats Allow...
100004     [Laundry In Building, Dishwasher, Hardwood Flo...
100007                          [Hardwood Floors, No Fee]
100013                                          [Pre-War]
100014                                                      []
100016     [prewar, elevator, Dogs Allowed, Cats Allowed,...
100020     [Doorman, Elevator, Pre-War, Terrace, Laundry ...
100026     [Cats Allowed, Dogs Allowed, Elevator, Laundry...
100027                        [Dishwasher, Hardwood Floors]
Name: features, dtype: object
```

❖ ADDING NEW FEATURES TO THE TRAINING DATA FRAME

❖ Extracting Individual fields from these features and adding them to the main data frame, the new data frame was a **49000 x 300** matrix which looked something like this:

diplomats ok	amazing deal!	month & no-fee	common backyard	small dogs	part-time doorman	business center	700 sf	e fireplace	small dog ok on	entrances	park block	private balcony
actual photos	exclusive	private garden	washer/dryer	outdoor space	!!!!low fee!!!!	1/2 bath	specialist! harry	outdoor space	washer in unit	jacuzzi	sofa bed in living	hi rise
hi rise	playroom /nursery	central ac	gym/fitness	pool	parking-space	renovated 1 bed	virtual doorman	gas included	outdoor areas	service garage	case by case.	roofdeck

Random Forests
Precision: 0.92
Recall: 0.91
F-Score: 0.90

Ada Boost
Precision: 0.92
Recall: 0.90
F-score: 0.90

Logistic Regression
Precision: 0.96
Recall: 0.95
F-score: 0.95

❖ PROBLEMS ENCOUNTERED:

1. **High Variance:** Several columns with ‘0’ values
2. **Overfitting**: The model highly overfit the training data, giving the above scores
3. Classifier did not perform well when the data was split into 70% training, 15% validation and 15% testing data.

❖ ENSEMBLE APPROACH

- ❖ Two ensemble approaches used:
  - **Random Forest**
  - **Boosting (Adaboost)**

- ❖ Two different classifiers for text based and numerical features.

- ❖ Take the weighted mean or the best of both classifiers.

- ❖ Classifier for text based features:
  - "**Features**" and "**Description**" columns merged
  - **Preprocessing**: removed HTML tags, delimiters and stop words.
  - **TF-IDF Vectorization** of resulting column and using it in classifier

Random Forest				
	precision	recall	f1-score	support
0	0.63	0.03	0.05	960
1	0.71	0.99	0.83	8571
2	0.54	0.08	0.14	2807
avg/total	0.67	0.71	0.61	12338

AdaBoost				
	precision	recall	f1-score	support
0	0.32	0.17	0.22	960
1	0.77	0.81	0.79	8571
2	0.35	0.35	0.35	2807
avg/total	0.64	0.66	0.65	12338

❖ ENSEMBLE APPROACH

- ❖ Classifier for numerical features:
  - Features used: bathrooms, bedrooms, display_address, latitude, longitude, manager_id, No_of_photos, price
  - Every listing has certain number of photos. Added a new column **number_of_photos** for each listing.
  - Treat all features as categorical

Random Forest				
	precision	recall	f1-score	support
0	0.64	0.15	0.24	960
1	0.74	0.96	0.84	8571
2	0.43	0.16	0.23	2807
avg/total	0.67	0.72	0.65	12338

AdaBoost				
	precision	recall	f1-score	support
0	0.44	0.3	0.36	960
1	0.8	0.83	0.81	8571
2	0.38	0.38	0.38	2807
avg/total	0.68	0.69	0.68	12338

❖ HURDLES

1. Data is very skewed with the following class distribution: Low Interest Level: 70%; Medium Interest Level: 23%; High Interest Level: 7%
2. Training data was split into 75% of the total data and Test data constituted 25% of the entire data.
3. Stratified Sampling was used while splitting Training and Test data
4. Feature expansion:
    a. Lots of noise in features column.
    b. Difficult to identify similar columns like pets allowed, cats/dogs allowed, cats allowed, dogs allowed both in training and testing data-set.
    c. Number of expanded features not same in training and testing data-set.
    d. Hence, we moved forward with the ensemble approach.

❖ CONCLUSIONS

1. Due to the skewed dataset, the best results could be achieved by the below two classifiers: AdaBoost and Random Forests

2. A large number of features in a dataset causes the model to be biased and thus overfits the data.

3. High number of features results in overfitting of classifier.

4. Dimensionality Reduction (using **Variance Threshold** where the features having high variance are dropped) does not solve overfitting problem.

5. It is efficient to convert all categorical data to quantitative values to better analyse and classify. **For example**: The columns “Latitude” and “Longitude” were converted to numerical codes and the class labels “interest_level” containing categorical values “high”,”medium”,”Low” were converted to quantitative values “0”,”1” and “2”.

Thank You

Questions?