Capstone Project Praposal

Finding the Interest Level of Rental Listings

Domain BackGround

- This project is taken from Kaggle data science competition of a startup company Renthop and it falls into the domain of real estate. Machine learning for real estate market is explored thoroughly in both academic and Tech industry community. For instance, in Udacity Nano Degree also we explored Boston housing prices using machine learning. In Tech industry, Zillow Inc. applies machine learning thoroughly to predict house prices using historical data and also applies machine learning in its recommender systems. The project is little different from what traditional tech real estate companies do with Machine Learning. In our project we try to categorize rental listings in such a way that end user don't need to spend time in finding the highly interested advertisements.

Problem Statement-

- In the project we need to classify the rental listings based on interest level, So we need to find out the Interest level. In the rental listing advertisement we have some important data such as No.of Bedrooms, No.of Bathrooms, Features, Description, Longitude, Latitude etc. and using these we need to classify the rental advertisement. We can model this real world real estate problem into a machine learning problem by taking the advertisement data as input applying classification algorithms on this data to develop a classification model that classifies the incoming advertisements according to the interest level. Interest level is classified into 3 classes 'High' , 'Medium' and 'Low'

Datasets and Inputs

Attached datasets along with this document. Datasets are in JSON format, and for our data analysis and machine learning modeling we convert JSON format data into DataFrames using Pandas Library

- Train data contains 49352 instances and it takes 100 MB of memory
- Test data contains 74659 instances and it takes 47.3 MB of memory

Data Source URL → https://www.kaggle.com/c/two-sigma-connect-rental-listing-inquiries/data

Data Fields lists and their description

- **bathrooms: number of bathrooms**
- **bedrooms: number of bathrooms**
- **building_id**
- **created : ad reation date**
- **description**
- **display_address**
- **features: a list of features about this apartment**
- **latitude**
- **listing_id**
- **longitude**
- **manager_id : id of the real estate manager in the renthop portal**
- **Photos: a list of photo links. You are welcome to download the pictures yourselves from renthop's site, but they are the same as imgs.zip.**
- **price: in USD**
- **street_address : house's street address in New York City**
- **interest_level: this is the target variable. It has 3 categories: 'high', 'medium', 'low'**

The possible output class labels are High, Medium and Low for the Interest Level .The training data is huge so I am not going to apply K-Fold for train test split, I am going to use regular Train –Test Split.Apart from this I am not going to use any other special methods considering the volume of data

***Note*:** test set doesn't contain target label, we need to actually split training set into train-test and get a model and apply this on our actual test data set. The results of actual test set are

Solution Statement

Incoming Data will be pre-processed and clustering algorithms will be applied on longitude and latitude columns of data and this will yield geographical clusters as output. Real estate advertisements interest level depends upon the Geo graphical locations. for ex: a listing in park avenue in Manhattan will have high interest level than flushing in Queens (Price and Geographical Locations might not be correlated for interest level, a 1 Bedroom from $5000 in wall street usually have less interest level than a 3 Bed Room loft for $2000 in Brooklyn Heights) . Later I would like to NLP on description to extract some valuable features and these features later will be converted into columns. Labeled data will be converted into numerical data using one-hot encoding. On the numerical data, I will apply Train test split and later feed the train data to classification algorithms such as Random Forest Trees, AdaBoost etc . The algorithms with the best evaluation score will be chosen using and the actual test data will be feed to the developed model to get the interest level. The solution is different from the Kaggle's anticipated solution. Kaggle want people to find out the probabilities of the 3 interest levels for the given input advertisement .whereas I am interested in predicting only the single interest value i.e the best out of 3 levels for the given interest level.so that I can tag the listing with this predicted interest level and show those to end users with sorting order (interest level High-Low) thus making apartment search easy and helping the company to better handling the fraud.

Benchmark Model

The ideal benchmark model will be a classification algorithm and it should classify the output as multi classes. So, the benchmark model is a Random Tree forests as random tree forests is a multi-class classification algorithm. The output label is interest level and it has 3 classes High, Low and Medium so the benchmark model will yield the F1 score of 0.33. So my developed model should score 0.33 on F1 evaluation metric

Evaluation Metrics-

F1 score is used as evaluation metric in the project. F1 score is the evaluation metric to measure accuracy; it is the ratio of true positive rate (recall) to precision. i.e we recall is when actually yes , how often it is predicted yes and precision is when it is predicted yes how often it is correct. So F1 score measures accuracy using the statistics precision p and recall r. Precision is the ratio of true positives (tp) to all predicted positives (tp + fp). Recall is the ratio of true positives to all actual positives (tp + fn). F1 is a very good evaluation metric for the classification as F1 weights recall and precision equally .A good retrieval algorithm will maximize both precision and recall simultaneously. Thus moderately good performance on both will be favored over extremely good performance on one and poor performance on the other. I want to clearly classify the given listing as High or Medium or Low i.e. tagging a listing with any of these 3 labels. So, in our project, I am interested it to find the Recall and precision ratio. As F1 summarizes exactness and completeness I am going to use F1 as evaluation metric

Project Design

The input JSON data will be converted into DataFrames using pandas library and later this dataframes are pre-processed. K-Means clustering algorithm will be applied on longitude and latitude to get geographical clusters.
I will apply NLP on description and extract some valuable features such as brand new, vibrant, renovated, subway etc. I am not going to use any geo spatial libraries. I might use NLTK library, stemming, stop words to extract valuable information. The extracted data will be converted into columns, data of these columns is numerical, 0 if that label is absent in the listing and 1 if the feature is present in the listing. From the features column , I will take the most common features that appears in the input such as GYM, Laundry etc. and convert them into columns, data of this columns is numerical, 0 if that feature is absent in the listing and 1 if the feature is present in the listing. As a final step of pre-processing, labeled data will be converted into numerical data using one-hot encoding. Features selection is depend upon the domain knowledge, features such as building ID will not be considered for ML modeling. I might apply seaborne visualizations to check how the output varies by different inputs values of a column(Feature) (but this might not be required as Real Estate domain and Features are straight forward to understand) .After all these preprocessing steps we apply Train test split and later feed the train data to classification algorithms. I would like to use supervised learning, classification model to make predictions. This is a multi-class classification project, so we need to use a multi class classification model such as Random Forest Trees