Machine Learning Engineer Nanodegree

Capstone Proposal

Jim DiLorenzo
3/22/2017

Domain Background

Proposed project is to classify interest level in online apartment listings. The problem comes from a kaggle competition found here. The data and problem are provided by an apartment rental company located in the United States. The company has provided data for its apartment listings in the hope that machine learning can be applied to improve the quality and fidelity of listings.

Problem Statement

Given the various information apartment managers input into the online profile of an apartment, the problem is to classify online apartment interest level into one of three categories: low interest, medium interest, and high interest. By successfully identifying what features are important in listing apartments, future listings can be improved to ensure that apartment managers and apartment renters are able to best display and acquire information from the online listing.

Datasets and Inputs

Data available--and descriptions--are listed here. All data is available from the rental company on the kaggle page. The data includes the apartment listing information:

- 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
- price: in USD
- street_address
- interest_level: this is the target variable. It has 3 categories: 'high', 'medium', 'low'

The target output variable interest has three classifications: low, medium, and high. Apartment features are available as a list, which may require transformation and added input of multiple new input features. Of further interest, all photos related to the apartment listings are also available.

Solution Statement

Solution is to come up with a model that--using the available data inputs--best classifies apartment interest as a probability that interest in that apartment was low, medium, and high. Available test set and withheld test set can be used to show the accuracy of the model.

Benchmark Model

The proposed benchmark model would be to assign a guess in interest level that is in proportional to the probability of each interest level in the training set. Ie if the overall probabilities of the training set are 10% of interest level is high, 20% medium, and 70% low the benchmark model would predict 0.1, 0.2, and 0.7 for each guess.

Evaluation Metrics

The proposed evaluation method is the multi-class logarithmic loss function. For each test sample, the model will assign an output that is probability of high interest, probability of medium interest, and probability of low interest. The estimated probabilities are then judged against the actual label. Guesses are penalized based on how far the estimated probability for the actual label is from the actual label. This is the evaluation method being used for the competition. The method penalizes models for being overly confident while allowing the model to show uncertainty. It is also applicable to cases like this one where the target classes are unbalanced.

Project Design

My theoretical workflow is as follows:

1. Data Exploration
2. Data Preprocessing
3. Model Iteration
4. Reflection (Possible Feature and Model Expansion)
5. Model Selection

For step 1. Data Exploration, I would like to get an overview and feel for the data. I will perform such preliminary steps as a five number summary for continuous variables, table counts for categorical variables, correlation plots, and preliminary visual exploration of the data. Potential topics of interest include what parts of the United States the data is from (possible application of clustering analysis), possible exploration of how geographic markets relate to price, analyzing manager_id, price versus number of bedrooms, examining outliers and searching for duplicated data.

One point of interest will be exploring the list of apartment features in-depth to see the quantity and frequency of various features. This may results in greatly expanding the input feature set to account for the presence or absence of certain apartment features. While the image data appears to be interesting, I would prefer to hold off on analyzing it in-depth and potentially adding it to the model until step 4.

The next step, data preprocessing will be selecting and preparing the desired features. Depending on which features are selected and expanded upon, this will likely feature feature normalization and some one hot encoding. Possibly using PCA and building models upon those features will be considered.

For the model iteration step, I would like to start with simpler models and gradually increase complexity with accuracy prediction hopefully justifying the increase in complexity. This will not only help me review and reinforce the techniques I have learned in this course, but also is justified in the scope of the problem. The data provided by the problem at first glance looks simple: 14 input columns and 1 output column of three categories. However the feature space can be greatly complicated by turning one input--list of apartment features--into dozens of input variables; creating engineered features (combining the location variables into their own sorts of clusters); and possibly even using deep learning analysis on the 90 Gb of image data. However those steps can be costly. I would like to establish a baseline of the simpler method before increasing the complexity; both to avoid costs and to be able to measure the marginal improvement.

I would like to start with a simple logistic regression model with minimum features and see how accuracy is improved as more features are added. Then I would like to try SVM, Naive Bayes, random forests, and some other techniques as model complexity is gradually increased and parameters are tuned.

I included a step 4. Reflection because at this point is when I would like to reconsider how to use the image data. I am hopeful that at this point that I will be able to see some possible ways to use the image data. Perhaps using the image meta data like number of photos for each listing or brightness of the images rather than and in-depth look at each image. I would also like to see if a deep learning neural net of some sort could be applied here.

Further feature engineering may also be done at this point and fed back into the most promising models of the earlier steps. Error analysis will be done to gain insight into where models are failing and to allow for the possible combination of models. Based on results found so far, I anticipate re-engineering various features, selecting the most promising models, iterating to find the best hyper-parameters and selecting a finalized model for submission.