Machine Learning Engineer Nanodegree

Capstone Proposal

Benjamin DK Luong

Proposal

Domain Background

Expedia wants to take the proverbial rabbit hole out of hotel search by providing personalized hotel recommendations to their users. This is no small task for a site with hundreds of millions of visitors every month!

Currently, Expedia uses search parameters to adjust their hotel recommendations, but there aren't enough customer specific data to personalize them for each user. In this competition, Expedia is challenging Kagglers to contextualize customer data and predict the likelihood a user will stay at 100 different hotel groups.

Problem Statement

Planning your dream vacation, or even a weekend escape, can be an overwhelming affair. With hundreds, even thousands, of hotels to choose from at every destination, it's difficult to know which will suit your personal preferences. Should you go with an old standby with those pillow mints you like, or risk a new hotel with a trendy pool bar?

Datasets and Inputs

This is the link for data: https://www.kaggle.com/c/expedia-hotel-recommendations/data
The data in this competition is a random selection from Expedia and is not representative of the overall statistics.

The train and test datasets are split based on time: training data from 2013 and 2014, while test data are from 2015. The public/private leaderboard data are split base on time as well. Training data includes all the users in the logs, including both click events and booking events. Test data only includes booking events.

destinations.csv data consists of features extracted from hotel reviews text.

Note that some srch_destination_id's in the train/test files don't exist in the destinations.csv file. This is because some hotels are new and don't have enough features in the latent space.

File descriptions

- train.csv - the training set

- test.csv - the test set
- destinations.csv - hotel search latent attributes
- sample_submission.csv - a sample submission file in the correct format

Data fields

train/test.csv

Column name	Description
date_time	Timestamp
site_name	ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp, ...)
posa_continent	ID of continent associated with site_name
user_location_country	The ID of the country the customer is located
user_location_region	The ID of the region the customer is located
user_location_city	The ID of the city the customer is located
orig_destination_distance	Physical distance between a hotel and a customer at the time of search. A null means the distance could not be calculated
user_id	ID of user
is_mobile	1 when a user connected from a mobile device, 0 otherwise
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise
channel	ID of a marketing channel
srch_ci	Checkin date
srch_co	Checkout date
srch_adults_cnt	The number of adults specified in the hotel room
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room
srch_rm_cnt	The number of hotel rooms specified in the search
srch_destination_id	ID of the destination where the hotel search was performed
srch_destination_type_id	Type of destination

Column name	Description
hotel_continent	Hotel continent
hotel_country	Hotel country
hotel_market	Hotel market
is_booking	1 if a booking, 0 if a click
cnt	Numer of similar events in the context of the same user session
hotel_cluster	ID of a hotel cluster

destinations.csv

Column name	Description	Da
srch_destination_id	ID of the destination where the hotel search was performed	int
d1-d149	latent description of search regions	do

Solution Statement

Develop machine-learning models that can predict the likelihood a user will stay at 100 different hotel groups.

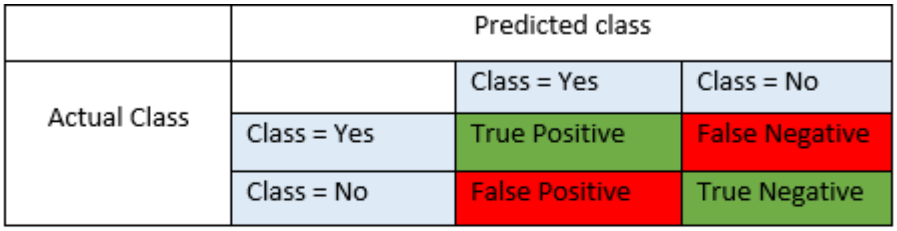
Benchmark Model

I don't have Benchmark Model. The goal is finding the best model that can predict the outcomes. Evaluation metrics are used to compare between models. The higher score is the better.

Evaluation Metrics

In order to compare between models, I use accuracy score and F-score to find out the best model.

According to Exsilio Solutions:



	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

True Positive = TP
True Negative = TN
False Positive = FP
False Negative = FN

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.

$$Accuracy = TP+TN/TP+FP+FN+TN$$

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

$$F1\ Score = 2*(Recall * Precision) / (Recall + Precision)$$

Project Design

(approx. 1 page)

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

First, I would clean up the data.

Second, I split the data into training set and testing set.

Third, I might use several algorithms in this project. Those algorithms can be:

- Support Vector Machine
- Random Forest Classifier
- SGD Classifier
- Naïve Bayes
- K-Nearest Neighbors Classifier

Finally, I compare the metrics between models to find the best model that has highest score.