The Automated Travel Agent: Hotel Recommendations Using ML

Diplav Srivastava, Harshil Jain, Kushagra Chandak

November 1, 2016

Guide: Dr. Avinash Sharma TA mentor: Tarun Gupta

Abstract

In this project, we aim to create the optimal hotel recommendations for Expedia.com customers that are searching for a hotel to book. We will model this problem as a multiclass classification problem and build variations of classic support vector machines (SVMs) and Naïve Bayes Classifier to predict the most likely hotel grouping from which a user will book a hotel. We use feature selection techniques to select optimal feature subsets, then build a unique combined SVM that achieves a higher precision and recall than either individual model alone.

1 Introduction

Our project is based on the Expedia Hotel Recommendations challenge on Kaggle. The goal of this project is to predict which of the 100 hotel clusters that a random Expedia visitor will book a hotel from. The high-level application of this project is to allow Expedia to provide the optimal personalized hotel recommendations for the user based on a user search event, which will increase the number of hotels booked through Expedia and simultaneously increase user satisfaction in the product. However, since the problem involves presenting optimal recommendations which the user is presented to choose from, this problem is not simply another multiclass classification problem
Recommending best hotel to customer on the basis of some parameters and hence giving a personalized hotel recommendation to all users. Expedia is interested in predicting which hotel group a user is going to book. In-house algorithms to form hotel clusters. These hotel clusters serve as good identifiers to which types of hotels people are going to book, while avoiding outliers such as new hotels that don't have historical data.

2 Dataset

Dataset is taken from Kaggle . It has 40 million instances and 22 features related to customer behavior are provided which includes family details, length of stay, location of person etc.

2.1 Features

There are many features given like user location,hotel location,hotel rating,distance between origin and destination etc. All these features play a significant role while booking a hotel like if the hotel lies in different continent with other features similar has less probability compared to other hotel within the same continent or state. Similarly hotel rating and family specifications matter for making a recommendation better. Some of the feature like date in time and out time was converted into length of stay which is a more significant or a useful feature for our problem.

Feature name	Description	Data type
site_name	ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp, ...)	int
posa_continent	ID of continent associated with site_name	int
user_location_country	The ID of the country the customer is located	int
user_location_region	The ID of the region the customer is located	int
user_location_city	The ID of the city the customer is located	int
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 int	double
is_mobile	1 when a user connected from a mobile device, 0 otherwise	int
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	int
channel	ID of a marketing channel	int
srch_adults_cnt	The number of adults specified in the hotel room	int
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room	int
srch_rm_cnt	The number of hotel rooms specified in the search	int
srch_destination_id	ID of the destination where the hotel search was performed	int
srch_destination_type_id	Type of destination	int
hotel_continent	Hotel continent	int
hotel_country	Hotel country	int
hotel_market	Hotel market	int

Figure 1: Features

3 Downsampling

As the dataset is very big, we down sample the data in the same ratio as in the original dataset (cluster wise) and take samples from a dataset of 400000. Since the training dataset was so huge(30 million data point) and required a lot computational time , we downsampled the training dataset such that prior probability of all the hotel cluster remained same.

3.1 Replaced Missing attribute with their Mean:

We filter out all training examples where the user did not actually book the hotel cluster that they clicked on and replaced all the example that contained missing information with their mean.

4 PCA

Principal component analysis (PCA) is a technique used to emphasize variation and bring out strong patterns in a dataset. It's often used to make data easy to explore and visualize. It is useful for reduing the dimensions of the data.

It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in

the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

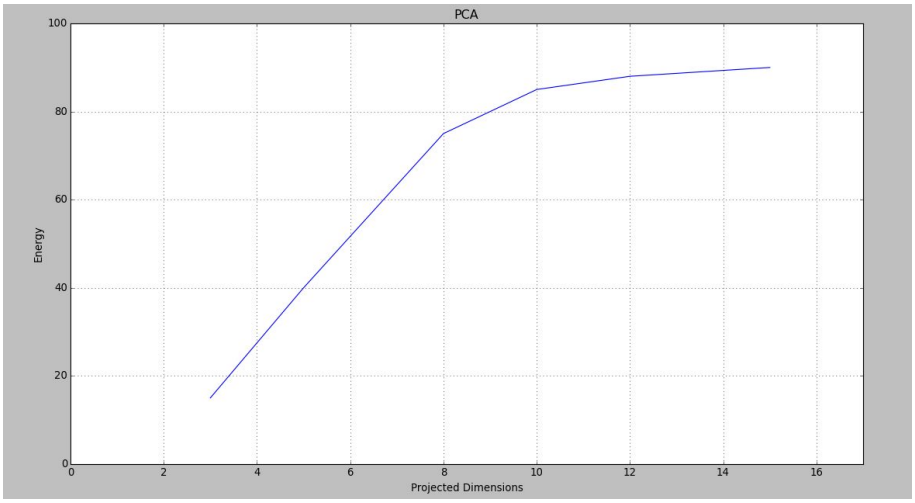

Figure 2: PCA

5 Naive Bayes Classifier

Naive Bayes classifier is based upon the principle of Maximum A Posteriori Probability (MAP). This approach is naturally extensible to the case of having more than two classes, and is shown to perform well inspite of the underlying simplifying assumption of conditional independence.

5.1 Analysis

For Naive Bayes, we hand-selected a few features, to get a basic machine learning algorithm running. We calculated target class probabilities for the categorical features, and we normalized continuous variables based on target label values. For each attribute in the test set, we converted its value to booking probability based on the the observed probabilities in the training set.

5.2 Assumptions

Attributes are linearly independent
The main assumption made while applying Naive Bayes classifier was that attribute of training dataset obtained after PCA are linearly independent so that we can directly treat each attribute as independent random variable and multiply each of them to get probability of each cluster.

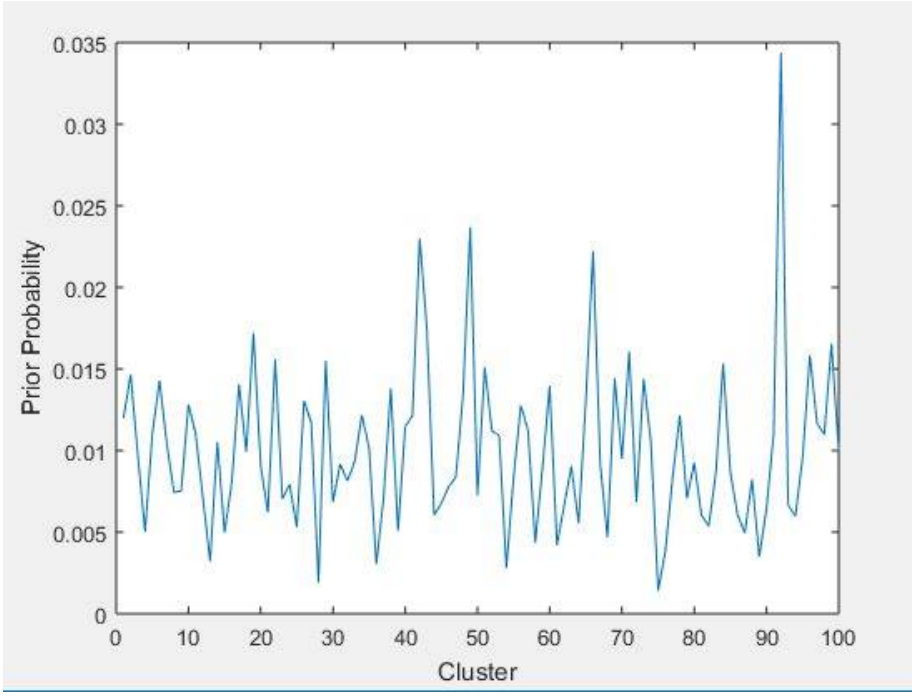

Figure 3: Prior Probability vs Cluster

Algorithm	Precision	Recall	F1
Naive Bayes	0.0596321	0.0567834	0.057489

Figure 4: Naive Bayes Results

6 Support Vector Machines (SVMs)

SVMs are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Then applied SVM RBF Kernel over the downsampled data and observed the results by varying different values of gamma and ratio of training and testing data(K).

Support Vector Machines are based upon the idea of maximizing the margin i.e. maximizing the minimum distance from the separating hyperplane to the nearest example. The basic SVM supports only binary classification, but extensions have been proposed to handle the multiclass classification case as well. In these extensions, additional parameters and constraints are added to the optimization problem to handle the separation of the different classes.
We have employed one vs all classification.

RBF kernel SVM gave the best results and linear SVM gave very poor results since the dataset was non-linear; polynomial kernel gave intermediate results.

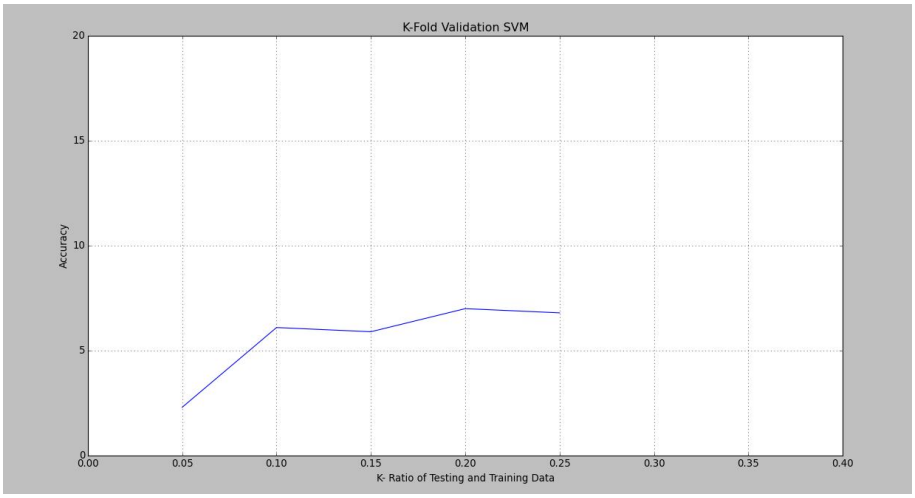

Figure 5: K-fold Validation

Algorithm	Precision	Recall	F1
SVM RBF	0.073425	0.069846	0.071976

Figure 6: SVM (RBF) results

Algorithm	Precision	Recall	F1
SVM polynomial	0.055238	0.053141	0.055980

Figure 7: SVM (Polynomial) results

Algorithm	Precision	Recall	F1
SVM linear	0.034829	0.032112	0.034091

Figure 8: SVM (Linear) results

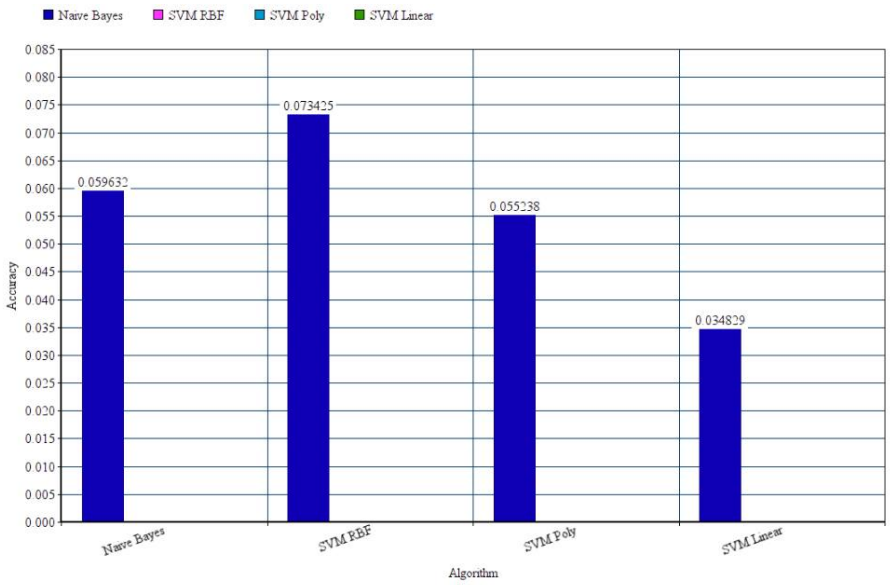

Figure 9: Comparison between different algorithms.